Aspect-based Document Similarity for Literature Recommender Systems

---

Dissertation

zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades
"Doctor rerum naturalium"
der Georg-August-Universität Göttingen

im Promotionsprogramm Computer Science (PCS)
der Georg-August University School of Science (GAUSS)

vorgelegt von
Malte Ostendorff
aus Kiel

Göttingen, 2023

Betreuungsausschuss:

Prof. Dr. Bela Gipp
Institut für Informatik, Georg-August-Universität Göttingen

Prof. Dr. Georg Rehm
Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Dr. Moritz Schubotz
FIZ Karlsruhe, Leibniz Institut für Informationsinfrastruktur

Mitglieder der Prüfungskommission:

Referent:

Prof. Dr. Bela Gipp
Institut für Informatik, Georg-August-Universität Göttingen

Koreferent:

Martin Klein, Ph.D.
Research Library, Los Alamos National Laboratory

2. Koreferent:

Prof. Dr. Harald Sack
Institut für Angewandte Informatik und Formale Beschreibungsverfahren, Karlsruher Institut für Technologie

Weitere Mitglieder der Prüfungskommission:

Prof. Dr. Georg Rehm
Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Dr. Moritz Schubotz
FIZ Karlsruhe, Leibniz Institut für Informationsinfrastruktur

Prof. Dr. Constantin Pape
Institut für Informatik, Georg-August-Universität Göttingen

Tag der mündlichen Prüfung: 25.04.2023

Acknowledgements

This thesis would not have been possible without the tremendous help and support from my family, friends, colleagues, and supervisors.

First and foremost, I am deeply thankful to my doctoral advisers, Bela Gipp, Georg Rehm, and Moritz Schubotz, for providing me with invaluable guidance, encouragement, and support. Their support and counsel enabled me to realize this thesis. I want to express my gratitude to Bela and Moritz, who made me pursue a research career in the first place and supported me throughout my research. Likewise, I thank Georg for providing me with the necessary resources and support to carry out my research.

I especially wish to thank Till Blume and Terry Ruas for our fruitful and always engaging discussions and for their valuable feedback and proofreading of the manuscript.

I further wish to gratefully acknowledge my friends, collaborators, colleagues, and co-authors with whom I had the distinct opportunity to work together, including Corinna Breitinger, Norman Meuschke, Julian Moreno-Schneider, Nils Rethmeier, Isabelle Augenstein, Arne Binder, Christoph Alt, Jan Nehring, Karolina Zaczynska, Jagrut Kosti, Volker Makl, and Elliott Ash. I also wish to thank Christian Schulze and his team for providing the computing infrastructure for my experiments.

My last and most deep gratitude goes to my family, who always cheered me in good and bad times and constantly backed and supported me.

Contents

Acknowledgements iii
Contents iv
Abstract ix
Zusammenfassung xi

I Introduction and Related Work

CHAPTER 1
Introduction **3**
1.1 Problem Setting 3
1.2 Research Gap 5
1.3 Research Objective 7
1.4 Thesis Outline 8
1.5 Prior Publications 10

CHAPTER 2
Related Work **13**
2.1 Recommender Systems 13
2.1.1 Content-based Recommender Systems 13
2.1.2 User-based Recommender Systems 18
2.1.3 Recommender System Evaluations 19
2.2 Similarity 23
2.2.1 Similarity vs. Relatedness 23
2.2.2 Similarity in Philosophy and Psychology 23
2.2.3 Similarity in Information Theory 25
2.2.4 Similarity Measures 27
2.3 Text-based Representations 28
2.3.1 Vector Space Model 28
2.3.2 TF-IDF 28
2.3.3 Neural Networks 29
2.3.4 Word Vectors 30
2.3.5 Paragraph Vectors 33
2.3.6 Recurrent Neural Networks 33
2.3.7 Transformer Language Models 34
2.4 Graph-based Representations 38
2.4.1 Direct Citations 39
2.4.2 Bibliographic Coupling 39
2.4.3 Co-Citations 40
2.4.4 Co-Citation Proximity Analysis 40
2.4.5 DeepWalk 42

2.4.6 Walklets 43
2.4.7 Other Graph Embeddings 43
2.5 Aspects in Information Processing 44
2.5.1 Aspect-based Expert Matching 45
2.5.2 Aspect-based Sentiment Analysis 46
2.5.3 Aspect-based Summarization 47
2.5.4 Aspect-based Representations 47
2.5.5 Aspect-based Text Similarity 49
2.6 Summary of the Chapter 50

II Aspect-free Document Similarity

CHAPTER 3
Wikipedia Article Recommendations **53**
3.1 Methodology 55
3.1.1 Dataset 55
3.1.2 Evaluated Methods 57
3.1.3 Evaluation Methdology 58
3.1.4 User Study Design 59
3.2 Offline Evaluation 61
3.2.1 Optimizing the CPI Model 61
3.2.2 Overall Results 62
3.2.3 Impact of Article Properties 65
3.2.4 Manual Sample Analysis 67
3.2.5 Discussion of Offline Evaluation 70
3.3 User Study Evaluation 71
3.3.1 Primary Results 71
3.3.2 Secondary Results 75
3.3.3 Discussion of User Study 76
3.4 Summary of the Chapter 78

CHAPTER 4
Legal Literature Recommendations **79**
4.1 Methodology 80
4.1.1 Datasets 80
4.1.2 Evaluation Methdology 81
4.1.3 Evaluated Methods 82
4.2 Evaluation 85
4.2.1 Overall Results 86
4.2.2 Impact of Document Properties 87
4.2.3 Coverage and Similarity of Recommendations 89
4.2.4 Manual Sample Analysis 91
4.3 Discussion 92
4.4 Summary of the Chapter 94

CHAPTER 5
Hybrid Research Paper Representations **97**
5.1 Methodology 99
5.1.1 Contrastive Learning 99
5.1.2 Citation Neighborhood Sampling 101
5.1.3 Datasets 103
5.1.4 Evaluated Methods 104
5.1.5 Implementation Details 105
5.2 Evaluation 106
5.2.1 Overall Results 106
5.2.2 Impact of Sample Difficulty 107
5.2.3 Ablation Analysis 110
5.3 Discussion 111
5.4 Summary of the Chapter 112

III Aspect-based Document Similarity

CHAPTER 6
Pairwise Classification for Wikipedia Articles **117**
6.1 Methodology 119
6.1.1 Dataset 119
6.1.2 Evaluated Methods 122
6.1.3 Implementation Details 124
6.2 Evaluation 125
6.2.1 Overall Results 125
6.2.2 Impact of Sequence Length 126
6.2.3 Impact of Concatenation 127
6.2.4 Impact of Aspect Classes 127
6.2.5 Manual Sample Analysis 129
6.3 Discussion 130
6.4 Summary of the Chapter 132

CHAPTER 7
Pairwise Classification for Research Papers **133**
7.1 Methodology 134
7.1.1 Datasets 134
7.1.2 Evaluated Methods 136
7.1.3 Implementation Details 138
7.2 Evaluation 138
7.2.1 Overall Results 138
7.2.2 Impact of Aspect Classes 139
7.2.3 Manual Sample Analysis 141
7.3 Discussion 142
7.4 Summary of the Chapter 143

CHAPTER 8
Specialized Research Paper Representations **145**
8.1 Methodology 148
8.1.1 Dataset 148
8.1.2 Evaluated Methods 150
8.1.3 Evaluation Methodology 151
8.2 Evaluation 151
8.2.1 Pairwise Results 152
8.2.2 Overall Results 153
8.2.3 Overlap of Recommendation Sets 155
8.2.4 Manual Sample Analysis 156
8.3 Discussion 159
8.4 Summary of the Chapter 161

IV Final Considerations

CHAPTER 9
Conclusion and Future Work **165**
9.1 Summary 165
9.2 Contributions 168
9.3 Lessons Learned 172
9.4 Future Work 173

Appendix
List of Figures 179
List of Tables 181
Bibliography of Publications 183
Bibliography 185
Glossary 213

Abstract

Literature recommendation systems assist readers in the discovery of relevant documents. Content-based systems recommend documents similar to the currently viewed document. However, the simple distinction between similar and dissimilar documents neglects the many aspects that make documents similar. For instance, two scientific papers may use a similar methodology while covering different research problems. Current document similarity measures are aspect-free, i.e., they cannot differentiate between specific aspects of the document content.

To address this limitation, this thesis proposes aspect-based document similarity for literature recommendations. By incorporating aspect information, recommendations can account for specific aspects of the document content. This thesis makes three contributions: First, it evaluates document representations and similarity measures and demonstrates that the lack of aspect information notably impacts recommendations. Second, it designs a new scientific document representation method that improves upon the state-of-the-art. Third, it designs two approaches for aspect-based document similarity that address the limitations of aspect-free similarity.

The thesis evaluates existing document similarity methods, focussing on methods that use graph and text information. The qualitative and quantitative evaluations reveal that although the overall user satisfaction is comparable between the two information sources, users perceive the recommendations from these sources as different. Therefore, the choice of similarity measures affects the generated recommendations, i.e., they implicitly address different aspects.

Furthermore, the thesis designs a novel scientific document representation method. The method is called SciNCL and relies on citation graph embeddings to select the most informative samples for the contrastive fine-tuning of a text-based document encoder. SciNCL achieves state-of-the-art results and is applicable for both aspect-free and aspect-based similarity.

Subsequently, the thesis first designs an aspect-based document similarity measure based on a pairwise multi-class classification approach. Unlike aspect-free similarity, which is a pairwise binary document classification – similar or not, the extension to a multi-class classification allows measuring similarity for a given aspect. The pairwise classification approach is implemented and evaluated for Wikipedia articles and scientific literature. The thesis also implements a second approach using specialized document representations to further improve the efficiency of aspect-based similarity. By formulating aspect-based similarity as a vector similarity problem in aspect-specific embedding spaces, aspect information is encoded only once per document and aspect. This makes the approach scale linearly with the corpus size. Further evaluations reveal that aspect-free representations have an implicit bias towards one aspect, confirming the problem of missing aspect information. The specialized document representations mitigate potential risks from implicit biases by making them explicit and controllable.

Finally, the practicality of aspect-based document similarity is demonstrated with a prototypical research paper recommender system. The prototype provides diverse recommendations from different aspects and recommendations tailored to specific aspects.

Zusammenfassung

Literaturempfehlungssysteme unterstützen den Leser relevanten Dokumente zu finden. Dabei nutzen inhaltsbasierte Systeme sog. Dokumentenähnlichkeitsmaße. Die alleinige Unterscheidung zwischen ähnlichen und unähnlichen Dokumenten vernachlässigt jedoch die vielen Aspekte, die Dokumente ähnlich machen. So können beispielsweise wissenschaftliche Artikel ähnlich in ihrer Methodik aber unterschiedlich in dem behandelten Problem sein. Heutige Dokumentenähnlichkeitsmaße sind aspektfrei, d.h. sie unterscheiden nicht zwischen Aspekten des Dokumentinhalts.

Um dieses Problem zu adressieren, schlägt diese Arbeit eine aspektbasierte Dokumentenähnlichkeit vor. Die Einbeziehung von Aspekten ermöglicht Empfehlungen für bestimmte Aspekte des Dokumentinhalts. Diese Arbeit leistet drei Forschungsbeiträge: Erstens werden Dokumentrepräsentationen und Ähnlichkeitsmaße evaluiert und es wird gezeigt, dass das Fehlen von Aspektinformationen Empfehlungen beeinträchtigt. Zweitens wird eine Methode zur Dokumentenrepräsentation entwickelt, die den Stand der Technik verbessert. Drittens werden zwei Ansätze zur aspektbasierten Dokumentenähnlichkeit entwickelt, die die genannten Probleme adressieren.

Die Arbeit evaluiert Dokumentenähnlichkeitsmaße, die Graph- bzw. Textinformationen verwenden. Die Evaluationen zeigen, dass die Nutzerzufriedenheit zwischen den beiden Informationsquellen zwar vergleichbar ist, die Nutzer aber die Empfehlungen aus diesen Quellen als unterschiedlich wahrnehmen. Daher beeinflusst die Wahl der Ähnlichkeitsmaße die generierten Empfehlungen, d.h. sie adressieren implizit unterschiedliche Aspekte.

Außerdem entwickelt die Arbeit eine Repräsentationsmethode für wissenschaftliche Artikel. Die Methode mit dem Namen SciNCL nutzt Contrastive Learning und Embeddings des Zitationsgraphen, um einen Dokumentenkodierer zu trainieren. SciNCL verbessert den Stand der Technik und ist sowohl für aspektfreie als auch aspektbasierte Dokumentenähnlichkeit anwendbar.

Anschließend wird zunächst ein aspektbasiertes Dokumentenähnlichkeitsmaß entwickelt, das auf einem paarweisen Mehrklassen-Klassifikationsansatz beruht. Im Gegensatz zur aspektfreien Ähnlichkeit, bei der es sich um eine paarweise binäre Dokumentenklassifikation handelt, ermöglicht die Mehrklassenklassifikation die Messung der Ähnlichkeit für einen bestimmten Aspekt. Der Klassifikationsansatz wird für Wikipedia und wissenschaftliche Artikel implementiert und evaluiert. Die Arbeit implementiert auch einen zweiten Ansatz mit verbesserter Effizienz, der auf speziellen Dokumentrepräsentationen basiert. Die aspektbasierte Ähnlichkeit wird als Vektorähnlichkeitsproblem in aspektspezifischen Embedding Spaces formuliert, dadurch werden die Aspektinformationen nur einmal pro Dokument und Aspekt kodiert. Weitere Evaluationen zeigen, dass aspektfreie Repräsentationen eine implizite Tendenz zu einem der Aspekte aufweisen, was das Problem der fehlenden Aspektinformationen bestätigt. Die spezialisierten Dokumentrepräsentationen machen diese Tendenzen explizit und somit kontrollierbar.

Schließlich wird die Anwendbarkeit der aspektbasierten Dokumentenähnlichkeit anhand eines prototypischen Empfehlungssystems für wissenschaftliche Artikel demonstriert. Der Prototyp bietet nicht nur vielfältige Empfehlungen zu unterschiedlichen Aspekten, sondern auch auf bestimmte Aspekte zugeschnittene Empfehlungen.

Part I

Introduction and Related Work

Chapter 1

Introduction

This thesis investigates aspect-based document similarity measures to improve content-based literature recommender systems, which is an open research challenge in information retrieval (IR) and natural language processing (NLP). Section 1.1 describes and motivates the problems arising from using document similarity measures for content-based recommender systems. Section 1.2 summarizes the research gap regarding the lack of aspect information in existing document similarity measures. Section 1.3 presents the research objective and research tasks, which guided the research, and defines relevant terminology. Section 1.4 outlines the presentation of my research in this thesis. Section 1.5 gives an overview of peer-reviewed publications, which are fully contained in this thesis.

1.1 Problem Setting

The continuously increasing amount of digitally available content has led to an information overload making it increasingly difficult to find relevant content (Roetzel, 2018). For example, the number of scientific papers published each year has grown steadily for over two centuries by about 3% per year (Johnson et al., 2018; Ware and Mabe, 2015). Such growth makes it more difficult and time-consuming for scientists to browse all papers published in their field (Ding et al., 2014). Consequently, recommender systems have become a crucial filtering and discovery tool for coping with the information overload, which many users of digital libraries rely on. For example, Lin and Wilbur (2007) report that PubMed's recommender system[1] for biomedical literature receives about 19% of the clicks.

Many recommender approaches like the popular collaborative filtering (Resnick et al., 1994, CF) rely on information about their users to provide individual recommendations based on the collected data. CF and related user-based approaches are crucial to many commercial services, e.g., recommendations for Amazon products (Smith and Linden, 2017) or Youtube videos (Davidson et al., 2010). However, in numerous scenarios, user-based recommender systems are not applicable because of the unavailability of user information or the user's information need changing too frequently to provide meaningful recommendations. Literature recommender systems often deal with such a scenario. Therefore, most of the literature recommender systems (approximately 55%) employ content-based document features and corresponding similarity measures instead of user-based features (Beel et al., 2016b). Content-based systems are based on the assumption that a user perceives a recommendation as relevant when the recommended document is semantically similar to the currently viewed document (Van Rijsbergen, 1979).

The task of recommending documents is typically divided into two major phases, feature representation and retrieval. First, features of documents are represented as numerical vectors, both the seed document (also called query document) and the document collection. Translating natural language text and other information into n-dimensional vectors is a core problem in IR and NLP. The vector space model (Salton et al., 1975), TF-IDF (Jones, 1972), Paragraph Vectors (Le and

[1] `https://pubmed.ncbi.nlm.nih.gov/`, last accessed: 18/01/2023

Mikolov, 2014), and BERT (Devlin et al., 2019) are common approaches to capture semantic text features. Similarly, non-textual document elements, such as citations in the scientific literature or hyperlinks in the Web, are an essential source of semantic information (Garfield, 2001; Gipp and Beel, 2009; Kessler, 1963; Small, 1973). Second, a retrieval method selects the most similar documents from the collection to the seed document. The cosine similarity is one common measure that computes the similarity score between document vectors. As illustrated in Figure 1.1, a similarity score is assigned to each document pair of seed and recommendation candidates. Then, the top-k recommendations are chosen from the candidate documents with the highest similarity to the seed document.

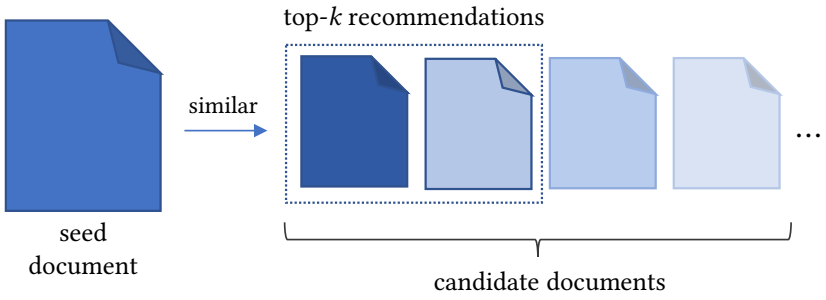


Figure 1.1: Content-based recommender systems assume that a user perceives a recommendation as relevant when the recommended document is semantically similar to the seed document, i.e, the recommendations are the most similar candidate documents.

The described process works entirely on content-based features and does not require any user information. Content-based features refer to all information directly derived from the documents themselves such as textual content like titles or the body text, graph information like citations, and metadata like author information. In contrast, user-based features are collected from user-document interactions, i.e., implicit feedback like clicks and explicit feedback in the form of ratings. Citations share some properties of user-based features. For instance, citations can be used to measure the popularity of a document. However, they are essentially document-document interactions that can be derived from the documents alone without users interacting with a system. Therefore, citations are typically considered content-based features.

Content-based recommender systems are independent of user information and, therefore, have the advantage of avoiding the cold start problem (Lika et al., 2014; Volkovs et al., 2017), which user-based approaches face when handling new items or novel users. But the disadvantage is that content-based approaches rely heavily on the notion of similarity and similarity acts only as an approximation for relevancy or another optimization goal. This originates from the goal of providing *relevant* recommendations. The question of whether a recommendation is relevant or not, however, is highly subjective and depends on the information needs of the individual users. Accordingly, relevancy can only be measured through user interactions. Since content-based systems do not have access to user information and instead rely on similarity measures, content-based recommendations struggle to address individual information needs. Moreover, similarity is also not necessarily an indicator of relevance. For instance, a document can be too similar to the seed to make it a relevant recommendation but rather be considered a duplicate or even plagiarism (Foltýnek et al., 2019). Determining the similarity of documents also requires

encoding the semantic information contained in the documents. Despite the recent progress in NLP induced by large language models (Brown et al., 2020; Devlin et al., 2019; Liu et al., 2019; Radford et al., 2019), the understanding of documents semantics remains challenging, especially for expert domains like law (Dehio et al., 2022) or for long documents (Beltagy et al., 2020).

Despite the challenges that arise with the development of content-based literature recommender systems, the demand for recommender systems will further increase. The trend of more available content is accelerating rather than slowing down, therefore, making progress on this problem is crucial. Moreover, document similarity is relevant to other related IR and NLP problems. For instance, finding semantically similar content is fundamental for many other applications, including question answering (Schwarzer et al., 2016a), plagiarism detection (Wahle et al., 2022), semantic storytelling (Rehm et al., 2022), and visualizations (Breitinger et al., 2020).

1.2 Research Gap

Content-based recommender systems provide a single untailored set of recommendations that is the same for all users and that is based on the similarity of the candidate documents with the seed document (as illustrated in Figure 1.1). The composition of the recommendations exclusively depends on the underlying feature representation method and the employed similarity measure. In contrast to these untailored recommendations, user-based approaches tailor recommendations specifically for individual users and their information needs. Content-based systems lack the ability to tailor recommendations. This inability is one major limitation of content-based recommendations compared to their user-based counterparts. The limitation originates from the use of similarity measures.

Today's similarity measures simply distinguish between similar and dissimilar documents. To put it differently, they neglect that documents can be similar not just in one but in many different ways. Such a distinction is too simple and does not reflect the heterogeneous semantics of complex documents such as research papers or court decisions, which are subjects of literature recommender systems. For instance, research papers cover multiple aspects of a topic, e.g., methodology, background, or results. Consequently, research papers can be similar in methodology but different in their results. Likewise, court decisions can have similar statements of facts but different legal consequences. These aspects, in which documents can be similar, are neglected by today's similarity measures. Instead, the similarity measures treat documents as singular entities even though the document semantics are rather heterogeneous. As a result, it remains unclear to what aspect the similarity relates, i.e., the similarity is aspect-free. Goodman (1972) already argued that the similarity of A to B is meaningless unless one can say "in what respect" A is similar to B. Hence, the similarity of A and B should only make sense if we know what aspects are considered. In other words, the similarity should be aspect-based instead of aspect-free.

Figure 1.2 illustrates how the similarity of documents can change depending on a given aspect. For aspect a_1 (green), the most similar documents and corresponding recommendations differ from the ones for aspect a_4 (orange). In practice, the set of aspects depends on the application domain and should be chosen to match potential user information needs. For example, the aspects of scientific literature could be the background, methodology, or results of research papers. Determining the document similarity with respect to one of these aspects enables recommendations tailored to specific aspects, which are relevant to the users.

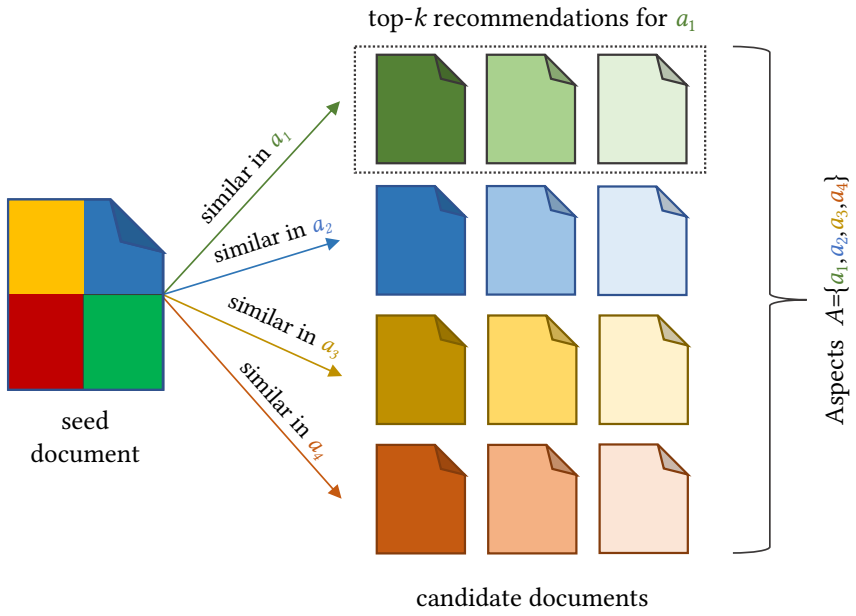


Figure 1.2: Aspect-based document similarity for recommender systems. Document semantics are heterogeneous (illustrated by different colors). The similarity measurement changes depending on the considered aspect. Recommendations can be tailored to a specific aspect.

Assuming that the set of aspects matches with information needs, the recommendations generated from aspect-based similarity are more likely to be relevant compared to the ones from aspect-free similarity. That is because recommendations from aspect-free similarity cannot be tailored to specific information needs. The aspect-based similarity would be especially beneficial for recommender systems with an expert audience. Experts have complex information needs and often search for relations between literature that may not be evident at first sight. For the domain of scientific literature, Chan et al. (2018) emphasized how important the discovery of analogies between research papers is for scientific progress. Current document similarity measures are not designed for solving analogical queries. An example of an analogical query is the recommendation of other papers with similar methodology but different results, i.e., combining the similarity or dissimilarity in multiple aspects. Solving such queries is crucial for finding distantly related yet highly relevant documents. One relevant paper that shares a specific aspect might remain undiscovered because it is from a different research field and does not share the vocabulary or citations with the seed document (Kang et al., 2022). At the same time, the ability to address aspects individually also allows mixing aspects to improve recommendation diversity.

Measuring the similarity of documents based on aspects also allows using these aspects as explanations to users. Aspects provide a better intuition on why a certain document is being recommended. With current recommendations from aspect-free similarity, the reasons for the recommendation remain opaque to the users. The lack of transparency can lead to mistrust, ultimately risking the general success of recommender systems, as pointed out by Zhang and Chen (2020) and Kunkel et al. (2019). More explainable recommendations improve the transparency, persuasiveness, effectiveness, and satisfaction of recommendation systems.

In this thesis, I investigate what I call aspect-based document similarity to address the previously described limitations. Instead of determining similarity in an aspect-free manner, namely as a tuple of a seed document d_s and a target document d_t, the aspect-based document similarity is a triple of d_s, d_t and an aspect a_i. My aspect-based interpretation of document similarity goes beyond the traditional document similarity approaches commonly used in related work. Hence, aspect-based document similarity can be considered a less established research problem. This is also reflected in diverse terminology, i.e., the literature refers to aspects as facets, semantic relations, or fine-grain document similarity.

In summary, the inability to tailor recommendations to specific information needs is one key weakness of content-based systems compared to their user-based counterparts. Aspect-based similarity measures address this weakness and also enable more explainable and diverse recommendations without the need to collect user information.

1.3 Research Objective

Taking the identified research gap as motivation, I define the following objective for my research:

Research Objective

Design, implement, and evaluate automated approaches to generate literature recommendations based on aspect-free and aspect-based document similarity measures.

To achieve my research objective, I derived the following research tasks:

Research Tasks

- **I** Evaluate state-of-the-art document similarity measures and underlying document representations that use text or graph information.
- **II** Design one document representation method that improves upon the state-of-the-art while using both text and graph information.
- **III** Design an aspect-based document similarity measure to address the limitations of existing aspect-free similarity measures.
- **IV** Implement aspect-based document similarity such that it scales to large document corpora.

To further specify the research objective and tasks, I define the relevant terminology and the scope of this thesis in the following. A definition of other terms can be found in the glossary of the thesis. Section 2.2 presents a more detailed discussion about the similarity term.

Aspects. The term *aspect* originates from the Latin word *aspectus* that means "looking at" (Lewis, 1891). Thus, an aspect of something is the direction or perspective from which it is looked at. Following this meaning, the Merriam Webster dictionary defines an *aspect* as "a particular status or phase in which something appears or may be regarded".[2] Transferred to the context of this thesis, an aspect of a document is the perspective from which a user may look at

[2] `https://www.merriam-webster.com/dictionary/aspect`, last accessed: 18/01/2023

the document's content. Thus, aspects are closely coupled to the information need a user may have and depend on the application domain. For example, a researcher might be interested in the methodology or the research problem of a paper, whereas a legal professional might be interested in the statement of facts or legal consequences of a court decision. Generally, the thesis considers only aspects concerned with the semantic level of documents. Other commonalities, such as visual or linguistic features, are excluded.

The literature uses various terms for such aspects. For instance, aspects are also referred to as multi-senses (Mancini et al., 2017; Nguyen et al., 2017), multi-perspectives (He et al., 2015), facets (Mysore et al., 2021; Risch et al., 2021), or contexts (Hofmann et al., 2010). These terms can have slightly different connotations and nuanced meanings while often referring to the same concept. However, these terms sometimes also refer to concepts different from this thesis. Facets are often referred to as boolean filters, whereas this thesis' scope is a more fuzzy similarity. Multi-senses are often used to describe word-level senses, whereas this thesis is about senses on a document level. Therefore, I settle on the term of aspects following the related and established NLP task of aspect-based sentiment analysis.

Aspect-based similarity. The key characteristic of aspect-based similarity is that the similarity assessment changes as the considered aspect changes. Given that an aspect is the perspective from which something is looked at, the aspect in the aspect-based similarity defines the perspective from that one looks at the document content when assessing the similarity. The aspect-free similarity is the counterpart to aspect-based similarity, i.e., the similarity assessment is regarded without any particular aspect.

Literature recommendations. Recommender systems are applied in diverse use cases. Prominent examples of use cases are e-commerce (Smith and Linden, 2017), entertainment (Davidson et al., 2010), or news (Karimi et al., 2018). This thesis focuses on the use case of literature recommender systems that are used by digital libraries to assist their users in finding relevant content. Accordingly, the goal is to recommend pieces of literature, which are generally any form of written work. To account for the diversity of available literature, the thesis conducts experiments with three types of literature: Wikipedia articles, court decisions, and research papers. In general, the thesis refers to instances of these literature types as documents.

Content-based features. Digital libraries often operate in a setting where no or only little user data is available (Beel et al., 2016b). To reflect this limitation, the thesis restrains its experiments to methods that rely exclusively on content-based features to generate recommendations and that do not require user data. Accordingly, the investigated methods exclusively rely on information derived from the content of documents, which is the main body text, graph data like citations or links, and metadata, e.g., the title or abstract of the document.

1.4 Thesis Outline

This thesis presents experiments that can be divided into aspect-free and aspect-based document similarity. The aspect-free experiments investigate a diverse set of methods in three different application domains, namely, Wikipedia articles, legal documents in the form of court decisions, and research papers. The aspect-based experiments propose and evaluate approaches to integrate

aspect information into document similarity measures. Wikipedia articles and research papers are revisited as application domains. The overall thesis is structured as follows:

Part I: Introduction and Preliminaries

Chapter 1 presents the problem of content-based literature recommendations and document similarity measures, identifies the research gap that motivated this thesis, and describes how the thesis addresses the research objective and the four research tasks.

Chapter 2 introduces the reader to related literature and background information relevant to the experiments presented in this thesis. The chapter reviews methods for document similarity and document representations on a conceptual level (Research Task **I**).

Part II: Aspect-free Document Similarity

Chapter 3 evaluates document similarity measures for Wikipedia articles comparing two graph-based and one text-based method (Research Task **I**). The methods are compared in an empirical offline evaluation and a qualitative user study. The comparison reveals that text-based and graph-based methods yield different notions of document similarity, each addressing different information needs and that this difference is also perceived by the users.

Chapter 4 evaluates 25 document representation methods for legal literature recommendations (Research Task **I**). This chapter extends the experiments of the previous chapter to a new literature domain and to a large number of state-of-the-art methods. The experiments reveal a little overlap in the recommendations from text-based and graph-based methods.

Chapter 5 designs a document representation method combining text and graph information (Research Task **II**). The designed method, called SciNCL, uses citation graph embeddings for the contrastive fine-tuning of a language model and achieves new state-of-the-art results.

Part III: Aspect-based Document Similarity

Chapter 6 conducts the first experiments on extending document similarity with aspect information using a pairwise multi-class document classification approach (Research Task **III**). This chapter revisits the literature domain of Wikipedia articles.

Chapter 7 extends the pairwise classification approach from a single-label to a multi-label classification problem and demonstrates its validity for research papers (Research Task **III**).

Chapter 8 continues with research papers as literature domain and improves upon the pairwise classification approach (Research Task **IV**). This is achieved by modeling the aspect-based similarity as a classical vector similarity task in aspect-specific embedding spaces.

Part IV: Final Considerations

Chapter 9 summarizes the research contributions of this thesis and proposes future work.

1.5 Prior Publications

To subject my research to peer review, I have published most of the content in this thesis in the publications listed below. The publications are in chronological order and associated with the chapters to which they contribute.

1. "Evaluating Link-based Recommendations for Wikipedia" by **Malte Schwarzer**, Moritz Schubotz, Norman Meuschke, Corinna Breitinger, Volker Markl, and Bela Gipp. In: *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries (JCDL)*, 2016. Chapter 3.
2. "Citolytics: A Link-based Recommender System for Wikipedia" by **Malte Schwarzer**, Corinna Breitinger, Moritz Schubotz, Norman Meuschke, and Bela Gipp. In: *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys)*, 2017. Chapter 3.
3. "Pairwise Multi-Class Document Classification for Semantic Relations between Wikipedia Articles" by **Malte Ostendorff**, Terry Ruas, Moritz Schubotz, Georg Rehm, and Bela Gipp. In: *Proceedings of the 2020 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2020. Chapter 6.
4. "Contextual Document Similarity for Content-based Literature Recommender Systems" by **Malte Ostendorff**. In: *Proceedings of the Doctoral Consortium at ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2020. Chapter 6-7.
5. "Aspect-based Document Similarity for Research Papers" by **Malte Ostendorff**, Terry Ruas, Till Blume, Bela Gipp, and Georg Rehm. In: *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, 2020. Chapter 7.
6. "Evaluating Document Representations for Content-Based Legal Literature Recommendations" by **Malte Ostendorff**, Elliott Ash, Terry Ruas, Bela Gipp, Julián Moreno-Schneider, and Georg Rehm. In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law (ICAIL)*, 2021. Chapter 4.
7. "A Qualitative Evaluation of User Preference for Link-Based vs. Text-Based Recommendations of Wikipedia Articles" by **Malte Ostendorff**, Corinna Breitinger[3], and Bela Gipp. In: *Towards Open and Trustworthy Digital Societies: 23rd International Conference on Asia-Pacific Digital Libraries (ICADL)*, 2021. Chapter 4.
8. "Specialized Document Embeddings for Aspect-based Similarity of Research Papers" by **Malte Ostendorff**, Till Blume, Terry Ruas, Bela Gipp, and Georg Rehm. In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2022. Chapter 8.
9. "Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings" by **Malte Ostendorff**, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. In: *The 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022. Chapter 5.

[3]Corinna Breitinger and I equally contributed to the publication.

Additionally, I contributed the following publications that are partially related to the research presented in this thesis. For example, such publications address related natural language processing or information retrieval tasks like semantic storytelling, summarization, question answering, or document classification.

11. "An Interactive e-Government Question Answering System" by **Malte Schwarzer**, Jonas Düver, Danuta Ploch, and Andreas Lommatzsch. In: *LWDA 2016 conference - Lernen, Wissen, Daten, Analysen (LWDA)*, 2016.
12. "Enriching BERT with Knowledge Graph Embeddings for Document Classification" by **Malte Ostendorff**, Peter Bouronje, Maria Berger, Julián Moreno-Schneider, Georg Rehm, and Bela Gipp. In: *Proceedings of the GermEval Workshop 2019 – Shared Task on the Hierarchical Classification of Blurbs co-located with the 15th Conference on Natural Language Processing (KONVENS)*, 2019.
13. "Towards an Open Platform for Legal Information" by **Malte Ostendorff**, Till Blume, and Saskia Ostendorff. In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2020.
14. "Towards Discourse Parsing-inspired Semantic Storytelling" by Georg Rehm, Karolina Zaczynska, Julián Moreno-Schneider, **Malte Ostendorff**, Peter Bouronje, et al. In: *Proceedings of the Conference on Digital Curation Technologies (QURATOR)*, 2020.
15. "Named Entities in Medical Case Reports: Corpus and Experiments" by Sarah Schulz, JuricaŠeva, Samuel Rodriguez, **Malte Ostendorff**, and Georg Rehm. In: *Proceedings of the 12th Language Resources and Evaluation Conference (LREC)*, 2020.
16. "Semantic Storytelling: From Experiments and Prototypes to a Technical Solution" by Georg Rehm, Karolina Zaczynska, Peter Bouronje, **Malte Ostendorff**, Julián Moreno-Schneider, et al. In: *Computational Analysis of Storylines: Making Sense of Events*, 2021.
17. "Ordering Sentences and Paragraphs with Pre-trained Encoder-Decoder Transformers and Pointer Ensembles" by Rémi Calizzano, **Malte Ostendorff**, and Georg Rehm. In: *Proceedings of the 21st ACM Symposium on Document Engineering (DocEng)*, 2021.
18. "HiStruct+: Improving Extractive Text Summarization with Hierarchical Structure Information" by Qian Ruan, **Malte Ostendorff**, and Georg Rehm. In: *Findings of the Association for Computational Linguistics (ACL)*, 2022.
19. "Generating Extended and Multilingual Summaries with Pre-trained Transformers" by Rémi Calizzano, **Malte Ostendorff**, Qian Ruan, and Georg Rehm. In: *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*, 2022.
20. "Semantic Relations between Text Segments for Semantic Storytelling: Annotation Tool – Dataset – Evaluation" by Michael Raring, **Malte Ostendorff**, and Georg Rehm. In: *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*, 2022.
21. "Claim Extraction and Law Matching for COVID-19-related Legislation" by Niklas Dehio, **Malte Ostendorff**, and Georg Rehm. In: *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*, 2022.
22. "Identification of Relations between Text Segments for Semantic Storytelling" by Georg Rehm, **Malte Ostendorff**, Rémi Calizzano and Karolina Zaczynska and Julián Moreno-Schneider. In: *Proceedings of the Conference on Digital Curation Technologies (QURATOR)*, 2022.
23. "Efficient Language Model Training through Cross-Lingual and Progressive Transfer Learning" by **Malte Ostendorff** and Georg Rehm. In: *Practical ML for Developing Countries Workshop co-located with the International Conference on Learning Representations (PML4DC@ICLR)*, 2023.

24. "Integration of a Semantic Storytelling Recommender System in Speech Assistants" by Maria Gonzalez Garcia, Julian Moreno Schneider, **Malte Ostendorff**, and Georg Rehm. In: *Proceedings of Text2Story – Sixth International Workshop on Narrative Extraction from Texts held in conjunction with the 45th European Conference on Information Retrieval (ECIR)*, 2023.

To acknowledge the fellow researchers with whom I published, collaborated, and discussed ideas, I will use "we" rather than "I" in the remainder of this thesis.

Preprints of all my publications are available at
`https://ostendorff.org/pub`

My Google Scholar profile is available at
`https://scholar.google.de/citations?user=8WfhSIcAAAAJ`

Chapter 2

Related Work

This chapter provides background knowledge and relevant literature for the subsequent chapters. It introduces the fundamentals of recommender systems while focusing on a particular type of recommender systems that will be used throughout the thesis, namely content-based systems (Section 2.1.1), and their evaluations (Section 2.1.3). Given that similarity is crucial for content-based recommendations, the concept of similarity and its use in psychology (Section 2.2.2) and information theory (Section 2.2.3) is explained in this chapter. The remaining sections introduce methods relevant throughout this thesis, while we separately discuss text-based (Section 2.3) and graph-based approaches (Section 2.4). A review of related work about aspect-based NLP tasks concludes this chapter (Section 2.5).

2.1 Recommender Systems

A recommender system can be defined as an application that recommends the most suitable item to a particular user given a collection of items (Ricci et al., 2011). The problem of recommending items can be considered a sub-problem of information filtering or information retrieval. In other words, the most suitable item must be filtered or retrieved for the item collection. Recommender systems are applied in various domains and contexts, from shopping or entertainment over social media to digital libraries (Lu et al., 2015; Schafer et al., 1999). Examples for recommend items are Amazon products (Smith and Linden, 2017), YouTube videos (Davidson et al., 2010), research papers (Beel et al., 2016b), and news articles (Karimi et al., 2018), to name a few. Depending on the recommended item and the application domain, what is considered as the most suitable item can change. Thus, recommender systems are optimized for different goal metrics, for instance, relevance (Zheng et al., 2010), clicks (Feng et al., 2019b; Pan et al., 2019), novelty and diversity (Kaminskas and Bridge, 2017; Kunaver and Požrl, 2017; Mendoza and Torres, 2020), or business profit (Azaria et al., 2013; Jannach and Adomavicius, 2017).

Aside from their application domain, recommender systems can be categorized based on the information they rely on to generate recommendations. In the literature, you typically distinguish between user-based or content-based systems and a hybrid combination of user and content information (Adomavicius and Tuzhilin, 2005; Aymen and Imène, 2022; Fayyaz et al., 2020; Ricci et al., 2011).

2.1.1 Content-based Recommender Systems

Content-based recommender systems originate from research about information retrieval (Baeza-Yates and Ribeiro-Neto, 1999; Manning et al., 2008; Salton, 1989) and information filtering (Belkin and Croft, 1992). In line with collaborative filtering, content-based recommender systems are also referred to as content-based filtering (Aggarwal, 2016).

The literature also discusses content-based approaches that match users to items that are similar to what they have liked in the past without relying on other users' data (Pazzani and Billsus, 2007). However, this thesis focuses on content-based recommendations that do not require user

data and are generally user-independent. In this pure content-based setting, the system selects the recommendations for a given seed item based on their similarity to the seed item. The underlying assumption is that a user would perceive a recommendation as relevant if the seed item and the recommended item are similar (Van Rijsbergen, 1979). Unlike collaborative filtering, content-based recommendations do not experience cold-start issues and are less prone to filter bubbles. In the following, we review specific instances of content-based recommender systems categorized by their application domains relevant to this thesis. For other domains and a general overview, we refer to the surveys from Aggarwal (2016), Deldjoo et al. (2020), and Lops et al. (2011).

2.1.1.1 Research Paper Recommender Systems

Several literature surveys provide a comprehensive overview of studies about research paper recommender systems (Ali et al., 2021; Aymen and Imène, 2022; Bai et al., 2019; Beel et al., 2016b; Li and Zou, 2019; Ma et al., 2020). In particular, the recent survey from Kreutz and Schenkel (2022) highlights the recent developments in this field of study. To complement the existing surveys, we review studies in the following that reflect the diverse approaches to this research problem while we focus on more recent works.

The first research paper recommender systems can be traced back to traditional libraries science approaches such as bibliographic coupling or co-citations (see Section 2.4). Most studies reflect the general progress in NLP and IR, whereby the degree of task-specific modifications differ from study to study. Early recommender systems apply text-based techniques like n-grams (Ferrara et al., 2011; Nascimento et al., 2011), word-based topic models (Jiang et al., 2012; Lin and Wilbur, 2007; Wang and Blei, 2011), or term-frequency weighting (Ekstrand et al., 2010; Giles et al., 1998; Nascimento et al., 2011). But also, the citation graph is commonly used to find related research papers (Giles et al., 1998; Habib and Afzal, 2019). Graph information is not limited to citation graphs, i.e., studies also rely on graphs from co-author or venue networks (Ali et al., 2020; Baez et al., 2011; Du et al., 2020; Zhou et al., 2008).

In recent works, we can observe a trend towards more machine learning-based approaches and towards combining text and graph information. For instance, Kong et al. (2018) build paper embeddings from text with Paragraph Vectors (Section 2.3.5) and from citations with struc2vec (Ribeiro et al., 2017) and retrieve related papers based on the cosine similarity of their embeddings. Bhagavatula et al. (2018) use bag-of-words representations to encode the textual content of papers and then rank the papers with a second model that is trained with a triplet loss, whereby citations are used to sample positive and negative papers. Collins and Beel (2019) compare recommendations generated from key phrases (Ferrara et al., 2011), TF-IDF, and Paragraph Vectors in an online evaluation based on the Mr. DLib system (Beel et al., 2011). Mohamed Hassan et al. (2019) conduct the first study with Transformers, in which a combination of BM25 and Universal Sentence Encoder (Cer et al., 2018) outperformed strong baselines such as ELMo, BERT, or SciBERT on the CiteULike dataset (Wang and Blei, 2011). Zhang et al. (2019) learn hybrid text- and citation-based embeddings with a Skip-gram model (Section 2.3.4), whereby they incorporate word-to-word relations similar to word2vec, document-to-word relations with Paragraph Vectors, and document-to-document relations with a random walk approach on the citation graph (similar to DeepWalk). Ali et al. (2020) employ a mixture of Sentence-BERT embeddings, LDA topic modeling, and graph information from citations, co-authorship, and venues. Kanakia et al. (2019) present a hybrid recommender system based on the Microsoft Academic Graph using co-citation and a combination of word vectors and TF-IDF.

Despite this trend, the recent literature still investigates rather classical methods. Habib and Afzal (2019) extend bibliographic coupling with information about the section in which the citations are located, similar to CPA. Renuka et al. (2021) generate recommendations based TF-IDF representations from extracted keywords and key phrases. Tao et al. (2020) rank candidate recommendations based on their PageRank extracted from the citation graph.

Even though the literature employs a wide range of methods, the underlying approach is often essentially the same. In the first step, research papers are converted into embeddings, sometimes separately converted into text and graph embeddings, and then concatenate, sometimes jointly learned. Next, the recommendations are generated based on a similarity measure between the paper embeddings, e.g., cosine similarity. Studies that account for the many aspects a research paper can represent are only exceptions. Jiang et al. (2012) propose to satisfy user-specific information needs by recommending the most *problem-related* papers or *solution-related* papers to the user separately. They achieve this through splitting abstracts into a *problem* and a *solution* part and training separate LDA topic models for each segment. A similar segmentation approach is presented by Chan et al. (2018). Chan et al. segment abstracts into *background*, *purpose*, *mechanism*, and *findings*. Huang et al. (2020) apply the same segmentation approach as Chan et al. but to biomedical research papers. Kobayashi et al. (2018) classify sections into discourse facets. Related aspect-based approaches are also discussed in Section 2.5.4 and 2.5.5.

2.1.1.2 Citation Recommender Systems

The citation recommendation task is closely related to the recommendation of research papers (Section 2.1.1.1). The distinction between both tasks lies in incorporating a local context. A local context refers to the text surrounding a citation marker. Typical paper recommendations are independent of the local context and are only about the paper-to-paper relation. Unlike research papers, citations are specifically recommended based on a local context. The recommended citation is meant to back up single statements or claims contained in the local context. Since the local context can be also considered as an aspect, the citation recommendation task is related to the aspect-based document similarity as proposed in this thesis. The literature does not strictly follow the distinction between citation and paper recommendations, i.e, the terms research papers recommendations and citations recommendations are often interchangeably used (Ali et al., 2021; Ma et al., 2020). For an overview of the citation recommendation literature, we refer to the survey from Färber and Jatowt (2020).

Citation recommender systems are typically not personalized (Färber and Jatowt, 2020), exceptions are Liu et al. (2013) and Yin and Li (2017) who utilize the citing paper's author information in addition to content-based features. He et al. (2010) presented the first study explicitly focusing on citation recommendations based on a local context. He et al. rely on LDA to construct topic models for candidate papers and citation contexts. TF-IDF vectors and cosine similarity are used by Duma and Klein (2014). Huang et al. (2015) use the Skip-gram model (Section 2.3.4) to represent words in the citation context and associate them with document embedding via a feed-forward neural network. Ebesu and Fang (2017) propose an encoder-decoder model consisting of a CNN-based citation context encoder, an encoder for the paper author, and an RNN-based decoder with attention. It is worth noting that Farber et al. (2020) tried to reproduce the work from Ebesu and Fang but were unable to achieve the same model performance.

The discussed studies focus on the citations of research papers. However, there are also other applications domains, e.g., Wikipedia (Fetahu et al., 2015; Piktus et al., 2021) or news articles (Peng et al., 2016).

2.1.1.3 Legal Recommender Systems

The legal domain is another domain of interest for this thesis. Therefore, this section presents studies about recommender systems for legal literature. Legal literature covers case law, court decisions, statutes, and other documents in the context of the law.

Winkels et al. (2014) are among the first to present a content-based approach to recommend legislation and case law. Their system uses the citation graph of Dutch Immigration Law and is evaluated with a user study conducted with three participants. Boer and Winkels (2016) propose and evaluate LDA (Blei et al., 2003) as a solution to the cold start problem in a collaborative filtering recommender system. In an experiment with 28 users, they find the user-based approach outperforms LDA. Wiggers and Verberne (2019) study citations for legal information retrieval and suggest citations should be combined with other techniques to improve performance. Kumar et al. (2011) compare four different methods to measure the similarity of the Indian Supreme Court decision: TF-IDF on all document terms, TF-IDF on only specific terms from a legal dictionary, Co-Citation, and Bibliographic Coupling. They evaluate the similarity measure on 50 document pairs with five legal domain experts. In their experiment, Bibliographic Coupling and TF-IDF on legal terms yield the best results. Mandal et al. (2017) extend this work by evaluating LDA and document embeddings (Paragraph Vectors) on the same dataset, whereby Paragraph Vectors were found to correlate the most with the expert annotations. Indian Supreme Court decisions are also used as evaluation by Wagh and Anand (2020), using document similarity based on concepts instead of the full text. They extract concepts (groups of words) from the decisions and compute the similarity between documents based on these concepts. Their vector representation (average of word embeddings and TF-IDF) shows that IDF for weighting word2vec embeddings improve results. Also, Bhattacharya et al. (2020a) compare citation similarity methods, i.e., Bibliographic Coupling, Co-citation, Dispersion (Minocha et al., 2015) and Node2Vec (Grover and Leskovec, 2016), and text similarity methods like Paragraph Vectors. They evaluate the algorithms and their combinations using a gold standard of 47 document pairs. A combination of Bibliographic Coupling and Paragraph Vectors achieves the best results. With Eunomos, Boella et al. (2016) present a legal document and knowledge management system that allows searching legal documents. The document similarity problem is handled using TF-IDF and cosine similarity. Other experiments using embeddings for similarity of legal documents include Landthaler et al. (2016), Nanda et al. (2019), and Ash and Chen (2018).

More recently, several shared tasks and public benchmarks contributed to the progress in legal document processing and retrieval, e.g., the AILA series (Bhattacharya et al., 2019; Bhattacharya et al., 2020b; Parikh et al., 2021), the COLIEE series (Rabelo et al., 2022; Rabelo et al., 2020; Rabelo et al., 2021), LeCaRD (Ma et al., 2021b), and LexGLUE (Chalkidis et al., 2022). In the AILA 2019 task (Bhattacharya et al., 2019), the two best teams (Shao and Ye, 2019; Zhao et al., 2019) both use combinations of TF-IDF and BM25. The same methods were also used by many teams in the 2020 continuation of the AILA task (Bhattacharya et al., 2019) but also a BERT-based approach yielded the best result for one of the sub-tasks. The best team (Tran et al., 2019) in the retrieval sub-task of COLIEE 2019 (Rabelo et al., 2020) addresses the problem of lengthy legal documents by first summarizing the candidate documents before matching them to

a query. Westermann et al. (2021) achieve the best results at COLIEE 2020 (Rabelo et al., 2021). Westermann et al. exploit the fact that relevance in the legal context is often only concerned with specific parts of a document. Therefore, they first split each case into paragraphs, embed the paragraphs with Universal Sentence Encoder (Cer et al., 2018), retrieve candidates based on their embedding similarity, and finally select the best candidate with an SVM model that works on TF-IDF document vectors. Also, the COLIEE 2021 retrieval task Rabelo et al. (2022) is won by a rather traditional method, Ma et al. (2021a) apply the Language Model for Information Retrieval from Banerjee and Han (2009), which is a statistical probabilistic framework based on the bag-of-words representations. Having traditional methods performing so well on these shared tasks emphasizes that they are strong baselines in diverse application domains. However, more and more legal adoption from state-of-the-art approaches, e.g., Lawformer (Xiao et al., 2021) or LegalBERT (Chalkidis et al., 2020; Holzenberger et al., 2020), are being published and, therefore, we expect that future tasks will be dominated by deep learning approaches as we can already see in other application domains.

2.1.1.4 Book Recommender Systems

As the third application domain, we consider the recommendations of books. In contrast to the specific domains of research papers or legal documents, we consider books without further specification and rather as generic examples of literature. Alharthi et al. (2018a) provides an overview of the literature on book recommender systems.

While for the recommendations of research papers semantic similarity is crucial, for book recommendations also other dimensions of similarity are relevant. For instance, Vaz et al. (2012) utilize not only semantic similarity (e.g., through LDA topic models) but also style similarity that is measured based on vocabulary richness, document length, part-of-speech bigrams, and the most frequent words in a book. Garrido et al. (2014) present the SOLE-R book recommender system, which relies on topic maps (Garrido et al., 2013) that are generated from the textual content of a book and reviews about the respective book. Garrido et al. extend the topic maps with lexical concepts, e.g., WordNet (Fellbaum, 2010), to remove redundancies and ambiguities. Tsuji et al. (2014) investigate book recommendations through a user study at a university library. Their system is based on user information (loan records) and content features like TF-IDF vectors. The children's book recommender system from Ng (2016) also incorporates user-based information through collaborative filtering and TF-IDF vectors from the book descriptions. Align with their use case of children's books, Ng further filter recommendations based on the readability level of a book such that it is appropriate for the user. Similar to Vaz et al., Alharthi et al. (2018b) as well consider style as a factor for book recommendations but their approach learns the style representation through an author identification model.

The presented studies about book recommender systems suggest that this particular application domain is a niche with respect to general recommender system research. Domain-specific adaptations are applied, e.g., focus on style similarity or readability level. Even more than other domains, the research about book recommendations suffers from data scarcity. The full text of books is typically not available and, therefore, recommendations are only generated based on book descriptions (blobs) and public book reviews.

2.1.2 User-based Recommender Systems

In its most common setting, recommender systems rely on user information for recommending items. The recommendation problem is often reduced to the problem of predicting ratings for the items that users have not seen based on their past ratings. A rating can be explicit information, e.g., given in the form of ratings or *likes* and *dislikes*. When explicit rating information is unavailable or insufficient, recommender systems utilize implicit ratings that are derived from user behavior, such as click histories (Oard and Kim, 1998). In the context of scientific literature, citations are often used as an implicit positive vote for a paper (Cohan et al., 2020; McNee et al., 2002).

User-based recommender systems are referred to as collaborative filtering approaches. Goldberg et al. (1992) coined the term "collaborative filtering" and made the first step toward user-based recommendations by incorporating user opinions into a message database and search system. The Tapestry system from Goldberg et al. (1992) still required its users to query for other users' opinions actively, e.g., explicit filtering for "marked as excellent by Chris". Resnick et al. (1994) introduced the concept of collaborative filtering as it is understood today. The GroupLens system from Resnick et al. (1994) relieved users from the burden of formulating queries about other users' opinions. Instead, GroupLens used a database of historical user opinions to match each individual to others with similar opinions automatically. The underlying approach consists of gathering ratings from users, computing the correlations between pairs of users to identify a user's "neighbors" in the opinion space, and combining the ratings of those neighbors to make recommendations. The collaborative filtering approach dominates today's recommender system research as various surveys show (Koren et al., 2022; Schafer et al., 2007; Su and Khoshgoftaar, 2009; Yang et al., 2014). On the methodological level, collaborative filtering is typically implemented with matrix factorization techniques (Bokde et al., 2015; He et al., 2017; Koren et al., 2009), but also deep learning is gaining more attention in recommender system research (Cheng et al., 2016; Khan et al., 2021; Wang et al., 2015; Zhang et al., 2020b). A simple alternative to collaborative filtering but still a strong baseline is the recommendation of the most popular items (Beel et al., 2017; Ji et al., 2020).

User-based recommender systems predict ratings based on past or current information about the users and items. If this information is not available for new users or new items, the recommender system suffers from the so-called cold start problem (Bernardi et al., 2015; Lika et al., 2014). The cold start problem is prevailing predominantly in the context of literature recommendations. For example, Yang et al. (2009) find that their explicit rating data is too sparse to produce accurate recommendations since users were "too lazy to provide ratings". According to Beel et al. (2016b), data sparsity is a general problem when using collaborative filtering for research paper recommender systems.

Another problem that user-based recommender systems face are filter bubbles (Pariser, 2011), echo chambers (Ge et al., 2020), and feedback loops (Jiang et al., 2019). The reliance on machine learning-based systems that learn from past ratings or the ratings from similar users can lead to a feedback loop that decreases the diversity of personalized recommendations. Moreover, this can lead to echo chambers in which users' interests are reinforced by repeated exposure to similar items. Due to the social and political implication, i.e., polarization and radicalization (Chitra and Musco, 2020; Ledwich and Zaitsev, 2020), an increasing number of studies address these problems (Fabbri et al., 2022; Kaminskas and Bridge, 2017; Kotkov et al., 2020).

2.1.3 Recommender System Evaluations

To judge the superiority of one recommendation approach over another one, evaluations are essential. Similar to evaluations of other information systems, a valid recommender system evaluation requires appropriate evaluation methods, a sufficient number of data points, and a comparison of the novel approach against one or more state-of-the-art baselines (Grover et al., 1996; Rossi et al., 2018; Symons, 1991). While the exact evaluation setting depends on the application context, the literature typically distinguishes between three evaluation types: user studies, online evaluation, and offline evaluations (Beel et al., 2016b; Beel and Langer, 2015; Erdt et al., 2015).

2.1.3.1 User Studies

A user study is a scientific method to evaluate how a recommender system is perceived by its users (Knijnenburg, 2012). The evaluation is typically conducted through explicit ratings, which can be collected through a questionnaire and aim to quantify the participant's satisfaction or experience with the recommendations. The recommender system that receives on average the highest ratings can be considered the best system (Ricci et al., 2011). User studies can be conducted as a lab or real-world study, whereby lab studies might be affected by the participants being aware of taking part in a study (Leroy, 2011).

In general, participants in a user study should be unbiased to the evaluated recommendation approaches and need to be a random representative sample (Shani and Gunawardana, 2011). The number of participants should be large enough to produce statistically significant results (Knijnenburg, 2012). The requirement for a large number of participants makes user studies expensive to conduct, in particular in domains where expert participants are needed, e.g., law or scientific literature. This leads to the problem that many recommender system evaluations do not arrive at meaningful conclusions since their user studies are not large enough (Beel et al., 2016a).

Examples of user studies evaluating recommender systems are: Tsuji et al. (2014) conduct a user study with 32 students to evaluate their book recommender systems. Chan et al. (2018) ask members of their research group about the usefulness of their aspect-based paper recommendations. Kanakia et al. (2019) evaluate their recommender system with 40 researchers from their company. Ekstrand et al. (2010) conduct first an offline evaluation and then validate their findings with a user study.

2.1.3.2 Online Evaluations

An online evaluation, also known as real-world testing, measures the effectiveness of recommender systems in a real-world application in which real users are exposed to and interact with the recommendations (Erdt et al., 2015; Shani and Gunawardana, 2011). The key distinction from other evaluation methods is that the recommendations are evaluated under normal conditions, i.e., the evaluation setting is identical to the deployment setting.

The effectiveness is typically measured by metrics like click-through rate or download counts to approximate user satisfaction. But this approximation of user satisfaction is not without drawbacks. For instance, Zheng et al. (2010) have shown that clicks and relevance do not always correlate. In less user-centric evaluations, business metrics like advertising revenue or product sales can also be part of online evaluations (Azaria et al., 2013; Jannach and Adomavicius, 2017).

Despite providing valuable insights, online evaluations are out of the scope of most academic researchers as they require access to real-world recommender systems.

As a result, fewer literature recommender studies rely on online evaluations, as reported in surveys (Beel et al., 2016b; Erdt et al., 2015). Specifically, Kreutz and Schenkel (2022) report in their literature survey that only 3.7% of the studies were evaluated with an online evaluation. One example of an online evaluation is the work from Collins and Beel (2019), in which the authors compare three recommender approaches in the Mr. DLib system (Beel et al., 2011). Outside of the literature recommendation domain, online evaluations are more common (Amatriain and Basilico, 2015; Falk and Karako, 2022; Freno, 2017).

2.1.3.3 Offline Evaluations

Offline evaluations typically measure the accuracy of a recommender system based on a gold standard or ground truth data. Offline evaluations are also referred to as batch or data-centric evaluations since no users are involved in the evaluation. Being only dependent on data makes offline evaluation cheap and convenient to conduct. Therefore, most recommender system research relies on this type of evaluation, as the surveys from Erdt et al. (2015) and Beel et al. (2016b) reveal. But the heavy use of offline evaluation has been criticized since results from offline evaluations do not necessarily correlate with results from user studies or online evaluations (Hersh et al., 2000; Sanderson et al., 2010; Turpin and Hersh, 2001).

Common datasets used for offline evaluations in the context of research papers are CiteULike (Jiang et al., 2012; Mohamed Hassan et al., 2019), CiteSeer (Habib and Afzal, 2019; He et al., 2010; Zhou et al., 2008), DBLP (Ali et al., 2020; Bhagavatula et al., 2018; Zhou et al., 2008), ACL Anthology (Ali et al., 2020; Tao et al., 2020), PubMed (Bhagavatula et al., 2018; Jain et al., 2018; Lin and Wilbur, 2007), or Microsoft Academic Graph (Kanakia et al., 2019; Zhang et al., 2019).

2.1.3.4 Evaluation Metrics

Recommender systems are typically evaluated with common IR metrics of which precision and recall are the most prominent metrics.

Precision. Precision is the number of relevant recommendations in relation to the total number of retrieved recommendations.

Recall. Recall is the number of retrieved recommendations that are relevant in relation to the total number of relevant recommendations.

A high recall can be achieved if a system recommends all items regardless of their relevance. However, this strategy will lower the precision since irrelevant items are also recommended. Likewise, a recommender system that recommends only one relevant item when multiple relevant items exist achieves high precision but a low recall. Commonly, precision and recall behave contradictory. Whether a high precision or high recall is preferable depends on the domain and the use case. A typical user of a video platform might be interested in browsing exclusively through the first ten results (Davidson et al., 2010), thus, prefers a high precision over recall. On the contrary, a researcher doing a literature review may be willing to look at significantly more

than ten literature recommendations to find a relevant paper. Hence a researcher may favor high recall over precision.

F1-score. F1 is the harmonic mean of the precision and recall. Hence, the F1 score is typically used when neither a high recall nor a high precision is preferred by the use case.

Micro and macro average. When computing evaluations metrics for classification tasks, the final metrics can computed either as micro or macro average. The difference between macro and micro averaging is that macro averaging gives equal weight to each class while micro averaging gives equal weight to each sample. In particular for unbalanced class distributions, this difference is important. When each class has the same number of samples, macro and micro yield the identical scores.

Both precision and recall rely on the ability to judge an item's relevance. Following Manning et al. (2008), relevance is the ability to satisfy a user's information need, which can differ from user to user and query to query. In most real-world use cases, a strict division into relevant or irrelevant items is difficult. Some items might be highly relevant and others marginally. However, for simplicity and comparability, binary classification of relevance is typically used. Relevance ratings can be gathered through a user study.

Precision and recall are set-based evaluation metrics. They are calculated using unranked sets of items. However, recommender systems typically produce ranked item sets. Accordingly, rank-based metrics provide a more meaningful evaluation.

Precision and recall at k. Precision and recall at k (P@k and R@k) refers to the precision and recall limited to the top k recommendations.

Mean average precision (MAP). The MAP metric provides a single-figure measure of quality across recall levels and is defined for a set of queries Q as follows:

$$\mathrm{MAP}(Q) = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{|R_q|} \sum_{j=1}^{|R_q|} \mathrm{Precision}(R_{q,j}) \tag{2.1}$$

where $R_{q,j}$ is the recommendation for query q at rank j.

Mean reciprocal rank (MRR). The MRR metric reflects the scenario where the user is only interested to see one relevant item. For a single query q, the reciprocal rank is $\frac{1}{r_q}$ where the rank r is the position of the first relevant recommendation. If no recommendation is relevant, the reciprocal rank is 0. For multiple queries Q, MRR is the mean over the reciprocal ranks of all queries in Q:

$$\mathrm{MRR}(Q) = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{r_q} \tag{2.2}$$

Normalized discounted cumulative gain (nDCG). The nDCG metric is made for evaluations of non-binary notions of relevance. Like precision at k, nDCG is evaluated over the top k recommendations. For a set of queries Q, let $R(j,d)$ be the relevance score associated with the item d for query j. Then,

$$\text{nDCG}(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_{kj} \sum_{m=1}^{k} \frac{2^{R(j,m)}-1}{\log_2(1+m)} \tag{2.3}$$

where Z_{kj} is a normalization factor that makes a perfect nDCG ranking at k for query j equal to 1. For queries for which $k' < k$ items are retrieved, the last summation is done up to k'.

Click-through-rate (CTR). The click-through-rate represents the ratio of clicks $C_{s,d}$ from a seed item s to a recommended item d and the number of all outgoing clicks for the seed item s:

$$\text{CTR}(s,d) = \frac{C_{s,d}}{\sum_{j=1}^{|C_s|} C_{s,j}} \tag{2.4}$$

Coverage. Coverage measures the ratio of the recommended items to all available items, i.e., coverage reflects the diversity of the recommendations and not their relevance. The coverage for the method a is defined as in Equation 2.5 where D denotes the set of all available items in the corpus and D_a denotes the recommended items by a (Ge et al., 2010).

$$\text{Cov}(a) = \frac{|D_a|}{|D|} \tag{2.5}$$

The evaluation metrics discussed above represent the ones used throughout this thesis. These metrics are generally only a fraction of all metrics used in recommender system research. Depending on the application domain and use case, other metrics might be more suitable for evaluating a recommender system.

2.2 Similarity

The Oxford dictionary defines similarity as "the state of being like someone or something but not exactly the same."[1] This definition is vague since it remains unclear how "being like something" is defined. Given that similarity is essential to this work, this section discusses the concept of similarity from different perspectives and introduces relevant similarity measures. We start this section by contrasting the terminology of similarity and relatedness.

2.2.1 Similarity vs. Relatedness

Most of the literature uses the two terms *similarity* and *relatedness* interchangeably. For instance, Collins and Beel (2019) and Lin and Wilbur (2007) investigate "related articles" whereas the research subject of Mysore et al. (2022) and Mandal et al. (2017) is "document similarity". However, there are also different connotations of both terms. As noted by Resnik (1995), "semantic similarity represents a special case of semantic relatedness". According to Budanitsky and Hirst (2006), "relatedness is a more general concept than similarity". Budanitsky and Hirst provide a further differentiation in the context of lexical semantics: "similar entities are semantically related by virtue of their similarity, but dissimilar entities may also be semantically related by lexical relationships". However, in the context of literature recommendations and on the level of document semantics, the difference between similarity and relatedness is negligible. For this reason, we regard the terms as equivalent and use only the term of similarity throughout this thesis. If not otherwise mentioned, we also refer to the similarity with respect to the semantics and not to other forms of similarity such as stylistic, lexical, or structural similarity.

2.2.2 Similarity in Philosophy and Psychology

During the twentieth century, philosophical and psychological theories about similarity have been dominated by the geometrical model of similarity (Carnap, 1967; Decock and Douven, 2011).

According to Blough (2001), the geometric model expresses the representation of the similarity relationships among the members of a set of objects. An object is represented by its coordinates in a "similarity space." The similarity is defined as the distance between objects in this space. The closer together two objects are, the more similar they are. The geometric approach makes two assumptions: First, objects can be represented by values on a few continuous dimensions. Second, similarity can be represented by a distance measure δ in the coordinate space. Figure 2.1 visualizes an example of objects placed in a space with the dimensions of size and color.

The geometric approach to similarity yields a formally exact implementation of similarity, which allows comparative similarity judgments. The objects a and b are more similar to each other than objects c and d if $\delta(a,b) < \delta(c,d)$. With the help of a threshold value t for distances in the similarity space, an absolute similarity judgment can also be modeled. The objects a and b are similar if $\delta(a,b) < t$.

However, the geometric model has been criticized for its shortcomings. In his famous philosophical critique, Goodman (1972) describes "similarity as a slippery and both philosophically and scientifically useless notion" (Decock and Douven, 2011). Goodman argues that similarity

[1] `https://www.oxfordlearnersdictionaries.com/definition/american_english/similarity`, last accessed: 18/01/2023

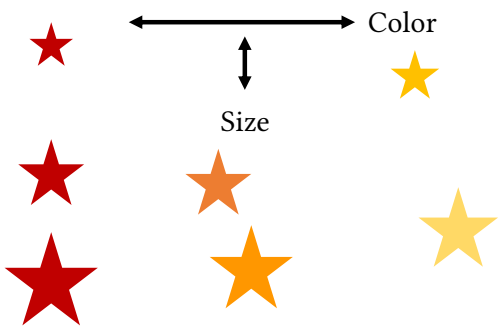


Figure 2.1: The geometric similarity model according to Blough (2001). Objects (stars) are arranged in a two-dimensional similarity space depending on their size (top to bottom) and color (left to right). In the simplest implementation, the length of a straight line between any two starts would determine their similarity.

judgments are highly context-sensitive. He states that "we must recognize that similarity is relative and variable" and that "similarity is much like motion" which requires a frame of reference. According to Goodman, similarity is an ill-defined notion unless one can say in what respects two things are similar. Medin et al. (1993) reaffirms Goodman's arguments with empirical evidence.

In addition to the context-sensitivity, Tversky (1977) criticizes the symmetry of the similarity in the geometric model, i.e., $\delta(a,b) = \delta(b,a)$. Tversky argues that similarity should not be treated as a symmetric relation. His work presents empirical evidence for the asymmetric notion of similarity, e.g., human similarity judgments find that an ellipse is more similar to a circle than a circle is to an ellipse. Given these findings, Tversky (1977) proposes his own set-theoretical approach to similarity based on feature matching: A central assumption of Tversky's linear contrast model is that objects are not characterized by points in a geometrical space but through a set of their features. For example, an strawberry can be represented as a set of features $A = \{\text{round}, \text{red}, \text{juicy}, ...\}$. The similarity of objects is then defined in terms of set-theoretical relations, whereby Tversky's model accounts for asymmetry and context-sensitivity.

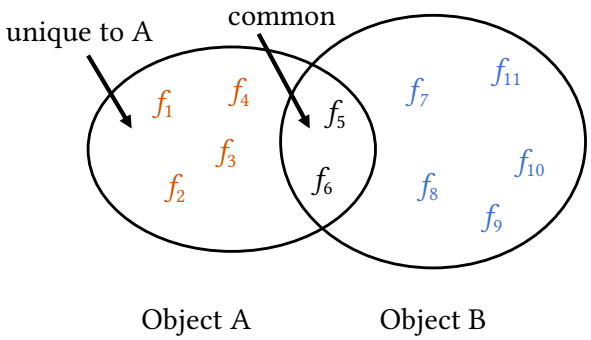


Figure 2.2: The feature similarity model from Tversky (1977). Objects A and B are represented as sets of their features, i.e., $A = \{f_1, ..., f_6\}$ and $B = \{f_5, ..., f_{11}\}$. The similarity of A and B depends on their common features and on what features are considered.

More recently, Gardenfors (2004) introduces with his *Conceptual Spaces* a contextualized version of the geometrical notion of similarity. Gardenfors argues that the relevant context for similarity can be achieved by a reference to conceptual spaces, e.g., if one compares objects with respect to color, the color sub-space should define the similarity of the objects.

We refer to Decock and Douven (2011) for a comprehensive discussion of the aforementioned similarity models. Decock and Douven conclude that "Goodman critique effectively highlights some shortcomings of the geometric model of similarity that was dominant at the time". But concerning the approaches from Tversky and Gardenfors, they find that both approaches do account for context-sensitivity and asymmetry, and, therefore, neither of these define "similarity is a slippery notion" (Decock and Douven, 2011).

2.2.3 Similarity in Information Theory

Most information retrieval systems, such as recommender systems, are based on computing the similarity between query and candidate items. Other NLP tasks, such as summarization or clustering, require computing the similarity between texts, too. Thus, computing similarity is either implicitly or explicitly a fundamental problem to many information systems. However, the similarity is mostly empirically compared without any theoretical foundation. This gap is closed by studies that use information theory to define the concept of similarity.

Resnik (1995) derives an information-theoretical definition of the semantic similarity of concepts in a taxonomy. Resnik defines the similarity between any two concepts c_1 and c_2 as the maximum information content of the set $C_{1,2}$ of all ancestors of c_1 and c_2. Relationships between concepts are given by the taxonomy. For example, "cash" and "credit" have the parent concept of "medium of exchange", while "coin" and "bill" are associated with the parent concept of "cash". More formally, Resnik defines the similarity of c_1 and c_2 as:

$$\text{Sim}_{Resnik}(c_1, c_2) = \max_{c_i \in C_{1,2}} [I(c_i)] \tag{2.6}$$

Central to Resnik's definition is the information content of a concept which follows the standard argumentation of information theory (Cover and Thomas, 2006), i.e., the information content I of a concept c is the negative log-likelihood of the probability of the concept: $I(c) = -logP(c)$. The empirical probability $P(c)$ is computed from a dataset. Taking the maximum information content is analogous to taking the shortest path in the taxonomy network with respect to edge distance.

Lin (1998) investigates the theoretical basis of similarity and derives a general form of an information-theoretic measure for object similarity. The similarity definition from Lin captures three intuitions:

1. The similarity between A and B is related to their commonality. The more commonality they share, the more similar they are.
2. The similarity between A and B is related to their differences. The more differences they have, the less similar they are.
3. The maximum similarity between A and B is reached when A and B are identical, independent of the commonality they share.

Given these intuitions, Lin derives six assumptions about the commonality, difference, similarity of identical objects, zero similarity, and similarity with independent perspectives. These assumptions lead to Lin's formal definition of similarity:

$$\text{Sim}_{Lin}(A,B) = \frac{I(\text{common}(A,B))}{I(\text{description}(A,B))} \tag{2.7}$$

The commonality of A and B (common features of A and B) is denoted as $\text{common}(A,B)$, whereas $\text{description}(A,B)$ is a proposition that describes what A and B are. Since similarity is the ratio between the information contained in the commonality and the description of the two objects, we can derive one from the other if the similarity is given.

For objects which can be represented as a set S of independent features s, the similarity definition can be reformulated to:

$$\text{Sim}_{Lin}(A,B) = \frac{2 \cdot \sum_{s \in A \cap B} \log P(s)}{\sum_{s \in A} \log P(s) + \sum_{s \in B} \log P(s)} \tag{2.8}$$

Aslam and Frost (2003) extend this general definition to document similarity by formulating the similarity of two documents as follows:

$$\text{Sim}_{Aslam}(A,B) = \frac{2 \cdot \sum_t \min(p_{A,t}, p_{B,t}) \log P(t)}{\sum_t p_{A,t} \log P(t) + \sum_t p_{B,t} \log P(t)} \tag{2.9}$$

The definition assumes that a document represents a set of independent term features. The probability $P(t)$ is the fraction of corpus documents containing the term t. For each document d and term t, let $p_{d,t}$ be the fractional occurrence of term t in a document d with $\sum_t p_{d,t} = 1$ for all documents in the corpus. The commonality of A and B is the minimum amount of term t they share in common and denoted as $\min(p_{A,t}, p_{B,t})$, while they contain $p_{A,t}$ and $p_{B,t}$ amount of term t individually.

The similarity definition from Lin (1998) accounts already for context-sensitivity through the probability of features. The similarity increases when commonalities are less likely. The residual entropy similarity from Cazzanti and Gupta (2006) aims to capture the context more strongly than Lin's similarity. Cazzanti and Gupta apply Tversky's linear contrast model with fixed parameters and measure the residual entropy to account for the context-sensitivity:

$$\text{Sim}_{Cazzanti}(A,B) = f(A \cap B) - 0.5 f(A \setminus B) - 0.5 f(B \setminus A) \tag{2.10}$$

where the salience function f is the conditional entropy of random objects R regarding their observed features with $f(X) = H(R|X \subset R)$. The salience function ensures that less frequent features are assigned with a higher weight than more frequent features, i.e., the specificity of features is captured. Amigó et al. (2017) provide a further discussion of information theoretical similarity definitions.

Even though these similarity definitions are backed by information theory and support the findings in philosophy and psychology (Section 2.2.2), they are less relevant in practice. Most of today's

information systems are based on the geometric similarity model and employ the corresponding similarity measures, e.g., vector representation and cosine similarity.

2.2.4 Similarity Measures

There are various methods to quantify the similarity between two items, a and b. An item can be given in the form of a text document d_i but also as any vector representation $\boldsymbol{x_i} \in \mathbb{R}^n$. For a comprehensive overview of text similarity and other distance measures, we refer to Deza and Deza (2013), Gomaa and Fahmy (2013), Jurafsky and Martin (2009), and Wang and Dong (2020). In the following, we discuss the similarity measures relevant to this thesis.

Euclidean distance. The Euclidean distance is the length of a straight line between two points $\boldsymbol{x_a}, \boldsymbol{x_b}$ in the Euclidean space with n dimensions:

$$\delta(a,b) = \sqrt{\sum_{i=1}^{n} (x_a^{(i)} - x_b^{(i)})^2} \tag{2.11}$$

Minimum edit distance. The minimum edit distance between two strings d_a, d_b is defined as the minimum number of editing operations (insertion, deletion, substitution) to convert d_a into d_b. In its simplest form, in which each operation is weighted equally, the minimum edit distance is also referred to as Levenshtein distance (Levenshtein, 1965). Wagner and Fischer (1974) is one example of an algorithm that determines the number of editing operations.

Cosine similarity. The cosine similarity measures the cosine of the angle between two vectors, which is the normalized dot product of the two vectors $\boldsymbol{x_a}$ and $\boldsymbol{x_b}$:

$$\text{cosine}(\boldsymbol{x_a}, \boldsymbol{x_b}) = \frac{\boldsymbol{x_a} \cdot \boldsymbol{x_b}}{|\boldsymbol{x_a}||\boldsymbol{x_b}|} = \frac{\sum_{i=1}^{n} x_a^{(i)} x_b^{(i)}}{\sqrt{\sum_{i=1}^{n} {x_a^{(i)}}^2} \sqrt{\sum_{i=1}^{n} {x_b^{(i)}}^2}} \tag{2.12}$$

Especially in a high-dimensional space, such as in word or document embeddings, cosine similarity is preferred over the Euclidean distance since cosine similarity accounts for different scales of the vectors. When vectors are pre-normalized, i.e., by dividing each vector by its length, the dot product is equal to the cosine. Cosine similarity is by far the most common similarity metric (Jurafsky and Martin, 2009).

Jaccard similarity coefficient. The Jaccard similarity coefficient (or Jaccard index) measures the similarity between two finite sets S_a and S_b and is defined as the size of the intersection $S_a \cap S_b$ divided by the size of the union $S_a \cup S_b$ of the two sets (Jaccard, 1912):

$$\text{Jaccard}(a,b) = \frac{|S_a \cap S_b|}{|S_a \cup S_b|} \tag{2.13}$$

In NLP, the Jaccard coefficient can be applied by representing documents as sets of words. Chapter 4 uses Jaccard to measure the similarity of recommendation sets.

2.3 Text-based Representations

In its original form, a text is just a string of characters with no particular structure that allows its processing by any mathematical means. Hence, the first step of any IR or NLP system is typically to derive a numerical representation of an input text. These methods of text-based representations range from the simple vector space model to complex Transformer language models and are discussed in the following section.

2.3.1 Vector Space Model

The vector space model organizes documents as vectors in a so-called term-document matrix (Salton, 1971). As illustrated in Figure 2.3, a row represents a term t_i that is part of the vocabulary V of the document collection D and a column represents a document d_j, whereby a matrix element or term weight $w_{i,j}$ is the number of occurrences of the term t_i in the document d_j.

	d_1	d_2	...	$d_{\|D\|}$
t_1	$w_{1,1}$	$w_{1,2}$	...	$w_{1,\|D\|}$
t_2	$w_{2,1}$	$w_{2,2}$	...	$w_{2,\|D\|}$
...	...	...	...	...
$t_{\|V\|}$	$w_{\|V\|,1}$	$w_{\|V\|,2}$	...	$w_{\|V\|,\|D\|}$

(a)

	d_1	d_2	d_3	d_4
car	1	5	3	0
truck	9	4	3	1
the	1	3	0	1
flower	0	0	0	4

(b)

Figure 2.3: The vector space model as a concept (a) and an example (b) for a term-document matrix with the four documents d_{1-4} and the four terms "car", "truck", "the", and "flower".

The term weight $w_{i,j}$ can be also a binary value (with 1 indicating that the term occurs in the document, and 0 indicating that it does not occur) or a term-frequency value (see Section 2.3.2). In practice, the term-document matrix becomes large, e.g., with a vocabulary size $|V| > 10^4$ and a document corpus $|D| > 10^6$. Due to most term weights being zero, the document representations are considered sparse vectors.

2.3.2 TF-IDF

Measuring the raw term frequencies as proposed by Salton (1971) can be skewed towards less informative words like *the*, *it*, or *they*, which occur very frequently but provide very little semantic meaning. The term frequency-inverse document frequency (TF-IDF) introduced by Jones (1972) addresses this shortcoming.

TF-IDF evaluates how relevant or important a term is to a document in a collection of documents. The importance of a term increases proportionally to the number of times a term appears in the document but is offset by the frequency of the term in the whole corpus. This intuition is achieved by the two factors TF-IDF consists of:

$$\text{tfidf}(t,d,D) = \text{tf}(t,d) \cdot \text{idf}(t,D) \tag{2.14}$$

The first factor of TF-IDF is the term frequency $\text{tf}(t,d)$, i.e., the frequency of the word t in the document d (Luhn, 1957), which can be simply the raw count of occurrences:

$$\text{tf}(t,d) = \text{count}(t,d) \tag{2.15}$$

With the raw count, too frequently appearing words may be over-represented. Hence, the term frequency is often normalized, for example, as in the Apache Lucene implementation[2]:

$$\text{tf}(t,d) = \sqrt{\text{count}(t,d)} \tag{2.16}$$

The second factor of TF-IDF is the inverse document frequency $\text{idf}(t,D)$ that measures the specificity of a term t with respect to its occurrences in the corpus of all documents D:

$$\text{idf}(t,D) = 1 + \log\left(\frac{|D|}{1+\sum_{d\in D}\text{count}(t,d)}\right) \tag{2.17}$$

TF-IDF is used in many studies about document similarity measures such as Boella et al. (2016), Collins and Beel (2019), Duma and Klein (2014), Kanakia et al. (2019), Kumar et al. (2011), Renuka et al. (2021), Tsuji et al. (2014), Wagh and Anand (2020), and Westermann et al. (2021), to name a few.

2.3.3 Neural Networks

Neural networks (also referred to as artificial neural networks) underpin most state-of-the-art techniques in natural language processing and other machine learning domains. Neural networks are not limited to text but can also process other modalities, e.g., graphs (Section 2.4). Essentially, neural networks can be seen as a composition of functions aggregated as layers. Different layers may perform different transformations on their inputs. In the following, we illustrate the inner workings of a neural network in its simplest form with only a single layer and with the example of a logistic classification model, which is defined as:

$$\begin{aligned} f(\boldsymbol{x}) &= \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}, \\ g(\boldsymbol{z}) &= \text{softmax}(\boldsymbol{z}), \\ \text{softmax}(\boldsymbol{z})_i &= \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \end{aligned} \tag{2.18}$$

where the input is represented as a vector $\boldsymbol{x} \in \mathbb{R}^d$ of d features, weight matrix $\boldsymbol{W} \in \mathbb{R}^{C\times d}$, bias vector $\boldsymbol{b} \in \mathbb{R}^C$, intermediate output variable $\boldsymbol{z} \in \mathbb{R}^C$, and C is the number of classes. The logistic model is a composition of $g(f(\cdot))$ of two functions f and g, where $f(\cdot)$ is an affine function and $g(\cdot)$ is a non-linear activation function. In this example, the softmax function is used as an activation function. Other examples of common activation functions are the sigmoid function or the rectified linear unit (Glorot et al., 2011).

[2] `https://lucene.apache.org/core/4_9_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html`, last accessed: 18/01/2023

Non-output layers are called hidden layers. Neural networks are typically defined according to the number of hidden layers they consist of. For example, a network with one hidden layer is commonly known as a feed-forward neural network, or multi-layer perceptron (MLP):

$$\begin{aligned} \boldsymbol{h_1} &= g_1(\boldsymbol{W_1}\boldsymbol{x} + \boldsymbol{b_1}), \\ \boldsymbol{y} &= \text{softmax}(\boldsymbol{W_2}\boldsymbol{h_1} + \boldsymbol{b_2}) \end{aligned} \tag{2.19}$$

where $\boldsymbol{x}$ is the input, $\boldsymbol{x}$ is the output, g_1 is the activation function of the first hidden layer. Each layer l is parameterized by a weight matrix $\boldsymbol{W_l}$ and bias vector $\boldsymbol{b_l}$. $\boldsymbol{h_l}$ is commonly referred to as the hidden state of the neural network at layer l. The weights of the neural network are typically trained with stochastic gradient descent and back-propagation (Rumelhart et al., 1986).

2.3.4 Word Vectors

A word vector aims to numerically represent the meaning of a word. Each word w_i in the vocabulary V is mapped to its vector representation $\boldsymbol{x}_i$, which is known as the word embedding of w_i. The word embeddings are stored in an embedding matrix $\boldsymbol{E} \in \mathbb{R}^{|V| \times d}$. A given text such as a single sentence or a full document, which consists of a sequence of words $w_i, ..., w_n$ can be represented by a sequence of word embeddings $\boldsymbol{x}_i, ...\boldsymbol{x}_n$.

Traditional count-based and sparse approaches to text representations, such as TF-IDF do not capture information about the context in which words are used. In essence, the count-based approaches treat words as atomic units and represent them as indices in a vocabulary (also referred to as one-hot-encodings). This neglects the (semantic) relationships between words and, therefore, represents language in a naive way. Instead, word embeddings map semantically similar words to proximate points in the embedding space. This results in multidimensional continuous real-valued number representations, i.e., dense vectors as opposed to sparse vectors (illustrated by Figure 2.4 and 2.5). In order to learn such a mapping, word embedding techniques rely on the *distributional hypothesis* (Harris, 1954), which states that words that share similar contexts tend to have similar meanings.

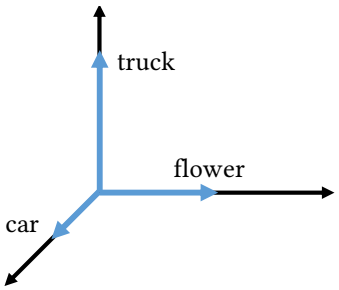


Figure 2.4: One-hot representation.

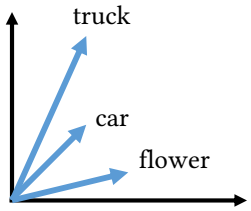


Figure 2.5: Dense representation.

The surveys from Camacho-Collados and Pilehvar (2018) and Wang et al. (2020b) provide an extensive overview of word embedding techniques.

2.3.4.1 Word2Vec

With Word2Vec, Mikolov et al. (2013a) and Mikolov et al. (2013b) popularized neural network-based learning of word embeddings. Word2Vec implements two models: Continuous Bag-of-Words (CBOW) and Skip-gram which are both simple single-layer neural networks based on the inner product between a pair of word vectors.

The training objective of the Skip-gram model is to learn word embeddings that predict the surrounding words in a sentence or a document given a target word w_t. More formally, given a sequence of training words $w_1, w_2, ..., w_T$, the objective of the Skip-gram model is to maximize the average log probability, which is defined as:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} \log p(w_{t+j}|w_t) \tag{2.20}$$

where w_t is the target word, w_{t+j} is a word in the context of t, and c is the context window size (number of words before and after the target word). The basic Skip-gram formulation defines $p(w_{t+j}|w_t)$ using the softmax function as in Equation 2.18. However, the traditional softmax formulation is computationally inefficient since it must sum across the entire vocabulary in order to evaluate the softmax function. Mikolov et al. show that the softmax formulation can be approximated through a hierarchical softmax approach (Morin and Bengio, 2005) and noise contrastive estimation (Gutmann and Hyvärinen, 2012). These approximations allow the efficient training of word embedding on a large corpus.

CBOW follows a similar intuition but is trained to predict the target word by its context words (or surrounding words). The objective of the CBOW is to maximize the following probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} \log p(w_t|w_{t+j}) \tag{2.21}$$

According to Mikolov et al. (2013b), Skip-gram yields better results on small datasets, and can better represent less frequent words, whereas CBOW trains more efficiently than Skip-gram and can better represent more frequent words.

2.3.4.2 GloVe

The word embeddings from Word2Vec rely on the local context (given as words before and after the target word) to represent the semantics of a word but do not account for the global statistics of the underlying corpus. Pennington et al. (2014) show that word embeddings can be also learned with global word statistics.

The GloVe model is based on the global word-to-word co-occurrence matrix denoted as C, whose entries $C_{i,j}$ represent the number of times the word j occurs in the context of the word i. To obtain word embeddings, GloVe factorizes this co-occurrence matrix to yield a lower-dimensional matrix, where each row yields a dense vector representation for the respective word. The matrix factorization corresponds to optimizing the following weighted least-squares objective:

$$J = \sum_{i,j=1}^{|V|} f(C_{i,j})(\boldsymbol{x}_i^T \tilde{\boldsymbol{x}}_j + \boldsymbol{b_i} + \tilde{\boldsymbol{b}}_j - \log C_{i,j})^2 \tag{2.22}$$

where $f(\cdot)$ is a weighting function that accounts for too frequent words, $\boldsymbol{x}_i \in \mathbb{R}^d$ is a word vector, $\tilde{\boldsymbol{x}}_i \in \mathbb{R}^d$ is a context word vector, $\boldsymbol{b}_i$ and $\tilde{\boldsymbol{b}}_i$ are bias terms. The final word vectors are constructed as the sum of $\boldsymbol{x}_i$ and $\tilde{\boldsymbol{x}}_i$.

2.3.4.3 FastText

Both word embedding techniques, Word2Vec and GloVe, rely on a fixed vocabulary V where each word of the vocabulary is assigned to a distinct vector. At the same time, the vocabulary size cannot be increased indefinitely due to limited computational resources. This can lead to the problem of words being *out-of-vocabulary* (OOV) that can occur especially for languages with large vocabularies and many rare words. The fastText method from Bojanowski et al. (2017) addresses the OOV problem by representing words as a bag-of-character n-grams. fastText builds upon the Skip-gram model (Mikolov et al., 2013b) and extends it with sub-word representations through character n-grams.

A character n-gram is a set of co-occurring characters within a given context window and a bag-of-character n-grams means that a word is represented by a sum of its character n-grams. Bojanowski et al. rely on boundary symbols $<$ and $>$ at the beginning and end of words to distinguish prefixes and suffixes from other character sequences. For example, the word *where* and $n = 3$ will yield the following character n-gram representation:

```
<wh, whe, her, ere, re>
```

The fastText method also includes the word itself in the set of n-grams, e.g., `<where>`. To obtain a vector representation for any word w, fastText computes the sum of the vector representations of the set of n-grams appearing in w.

2.3.4.4 Word Vector Pooling

While word embeddings can produce meaningful representations of words, this thesis is not concerned with single words but rather with sequences of words in the form of documents or sentences. To get from word to document embeddings, the most straightforward approach is to aggregate the embeddings of each word appearing in a document. For example, document embeddings can be computed as element-wise average, minimum, or maximum. These strategies are called average-pooling, min-pooling, and max-pooling, respectively. In addition to the aggregation, one can also concatenate different pooled embeddings. More sophisticated pooling strategies as well exist, e.g., the smooth inverse frequency model from Arora et al. (2017).

Pooling embeddings is a simple but effective approach for document embeddings. However, pooling completely ignores the order of the words in the document.

2.3.5 Paragraph Vectors

Paragraph Vectors (Le and Mikolov, 2014) extends the idea of Word2Vec (Mikolov et al., 2013b) to the learning of embeddings for word sequences of arbitrary length, e.g., paragraphs or documents. Paragraph Vectors is also referred to as Doc2Vec due to its popular implementation in the Gensim framework (Rehurek and Sojka, 2010). Similar to Word2Vec, Paragraph Vectors proposes two separate models to learn document embeddings: Distributed Memory Model of Paragraph Vectors (PV-DM) and Distributed Bag of Words version of Paragraph Vector (PV-DBOW).

In the PV-DM model, each paragraph is mapped to a unique vector and every word is also mapped to a unique vector. The paragraph vector and word vectors are concatenated to predict the next word in a context, similar to Equation 2.21. The contexts are fixed-length and sampled from a sliding window over the paragraph. The paragraph vector is shared across all contexts generated from the same paragraph but not across paragraphs.

PV-DBOW ignores the context words in the input but predicts words randomly sampled from the paragraph in the output. This has the advantage that word vectors do not need to be stored, i.e., the model requires less storage. PV-DBOW is analog to the Skip-gram model from Word2Vec.

2.3.6 Recurrent Neural Networks

A recurrent neural network (RNN) is a class of neural networks, which is made for processing sequential data. Originally proposed by Rumelhart et al. (1986), an RNN reuses its previous outputs as inputs in a recurrent fashion and shares the weights across the processing steps. RNNs can process variable-length sequences of inputs due to their internal state (memory). Since text can be represented as a sequence of words (or word embeddings), RNNs are often used in the context of NLP.

Vanilla recurrent neural networks. More formally, for each input $\boldsymbol{x}^{(t)}$ at the time step t an RNN computes the hidden state $\boldsymbol{h}^{(t)}$ and the output $\boldsymbol{y}^{(t)}$ as follows:

$$\begin{aligned} \boldsymbol{h}^{(t)} &= g_1(\boldsymbol{W}_a \boldsymbol{h}^{(t-1)} + \boldsymbol{W}_x \boldsymbol{x}^{(t)} + \boldsymbol{b}_h), \\ \boldsymbol{y}^{(t)} &= g_2(\boldsymbol{W}_y \boldsymbol{h}^{(t)} + \boldsymbol{b}_y) \end{aligned} \tag{2.23}$$

where $\boldsymbol{W}_x$, $\boldsymbol{W}_h$, $\boldsymbol{W}_y$, $\boldsymbol{b}_y$, and $\boldsymbol{b}_h$ are weights and bias terms that are shared across time steps and g_1, g_2 are activation functions. The hidden state $\boldsymbol{h}$ can be seen as a "memory" of the previous time steps in the sequence. The network weights can then be trained with the help of back-propagation through time (Rumelhart et al., 1986).

Long-short Term Memory (LSTM). Traditional RNNs often encounter the problem of vanishing and exploding gradient, to address this issue Hochreiter and Schmidhuber (1997) proposed the LSTM as a gated version of recurrent networks. The gates of an LSTM decide what information should be kept and what should be forgotten.

In contrast to the traditional RNN, the LSTM contains a forget gate $f^{(t)}$, input gate $i^{(t)}$, and output gate $o^{(t)}$, which are all functions of the current input $\boldsymbol{x}^{(t)}$ and the previous hidden state $\boldsymbol{h}^{(t-1)}$. The gates select which information to retain or overwrite depending on the previous cell state $\boldsymbol{c}^{(t-1)}$, the current input $\boldsymbol{x}^{(t)}$, and the current cell state $\boldsymbol{c}^{(t)}$:

$$\begin{aligned}
i^{(t)} &= \sigma(\boldsymbol{W}_i \boldsymbol{x}^{(t)} + \boldsymbol{U}_h \boldsymbol{h}^{(t-1)} + \boldsymbol{b}_i), \\
f^{(t)} &= \sigma(\boldsymbol{W}_f \boldsymbol{x}^{(t)} + \boldsymbol{U}_f \boldsymbol{h}^{(t-1)} + \boldsymbol{b}_f), \\
o^{(t)} &= \sigma(\boldsymbol{W}_o \boldsymbol{x}^{(t)} + \boldsymbol{U}_o \boldsymbol{h}^{(t-1)} + \boldsymbol{b}_o), \\
\boldsymbol{c}^{(t)} &= f^{(t)} \odot \boldsymbol{c}^{(t-1)} + i^{(t)} \odot \tanh(\boldsymbol{W}_c \boldsymbol{x}^{(t)} + \boldsymbol{U}_c \boldsymbol{h}^{(t-1)} + \boldsymbol{b}_c), \\
\boldsymbol{h}^{(t)} &= o^{(t)} \odot \tanh(\boldsymbol{c}^{(t)})
\end{aligned} \tag{2.24}$$

where $\boldsymbol{W}$ and $\boldsymbol{U}$ are weight matrices, $\boldsymbol{b}$ is a bias term, tanh and σ are activation functions, and $\odot$ is an element-wise multiplication. The gates discard irrelevant information which is then not back propagated in time. This mitigates the problem of vanishing or exploding gradients.

2.3.7 Transformer Language Models

The current state-of-the-art in most NLP tasks uses Transformer language models. In the following, we introduce the Transformer architecture and language models based on it.

2.3.7.1 Transformer

The Transformer (Vaswani et al., 2017) is a neural network architecture motivated by the goal of replacing the inherently sequential computation of RNNs with a more parallelizable approach based on self-attention (Bahdanau et al., 2014). At the core of a Transformer model is the Transformer layer (or block), which we introduce in the following:

A Transformer layer is a parameterized function class $f_\theta(\boldsymbol{x}) = \boldsymbol{z}$ that *transforms* the input $\boldsymbol{x} \in \mathbb{R}^{n \times d}$ into the output $\boldsymbol{z} \in \mathbb{R}^{n \times d}$. The input can be assumed to be a length-n sequence of d-word vectors. First, the input vector $\boldsymbol{x_i}$ is transformed into three vectors, query $Q^{(h)}(\boldsymbol{x_i})$, key $K^{(h)}(\boldsymbol{x_i})$, and value $V^{(h)}(\boldsymbol{x_i})$, through the learnable weight matrices $\boldsymbol{W}_q^{(h)}$, $\boldsymbol{W}_k^{(h)}$ and $\boldsymbol{W}_v^{(h)}$ and for $h \in H$ attention heads:

$$\begin{aligned}
Q^{(h)}(\boldsymbol{x_i}) = \boldsymbol{W}_q^T \boldsymbol{x_i}, \quad K^{(h)}(\boldsymbol{x_i}) = \boldsymbol{W}_k^T \boldsymbol{x_i}, \quad V^{(h)}(\boldsymbol{x_i}) = \boldsymbol{W}_v^T \boldsymbol{x_i}, \\
\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v \in \mathbb{R}^{d \times k}
\end{aligned} \tag{2.25}$$

Intuitively, in each of the attention heads the relation between queries and keys has a different meaning, e.g., syntactic similarity or distance in the input sequence.

The second step is calculating the self-attention score for word pairs in the sequence (Bahdanau et al., 2014). The score is the softmax function (Equation 2.18) over the dot product of the query vector with the key vector normalized by the square root of the dimension of the key vectors k.

$$\alpha_{i,j}^{(h)} = \text{softmax}_j \left(\frac{Q^{(h)}(\boldsymbol{x_i}) K^{(h)}(\boldsymbol{x_i})^T}{\sqrt{k}} \right) \tag{2.26}$$

The next step is the summation of value vectors V, weighted with the self-attention scores $\alpha_{i,1}, ..., \alpha_{i,n}$, over each attention head H.

$$\boldsymbol{u_i'} = \sum_{h=1}^{H} \boldsymbol{W}_{c,h}^T \sum_{j=1}^{n} \alpha_{i,j}^{(h)} V^{(h)}(\boldsymbol{x_j}), \quad W_{c,h} \in \mathbb{R}^{k \times d} \tag{2.27}$$

A MLP with ReLu activation and layer norm (Ba et al., 2016) computes the final output $\boldsymbol{z_i}$, where $\gamma_{1,2} \in \mathbb{R}^d$ and $\beta_{1,2} \in \mathbb{R}^d$ are layer norm parameters, and $\boldsymbol{W}_1 \in \mathbb{R}^{d \times m}$, $\boldsymbol{W}_2 \in \mathbb{R}^{m \times d}$ are MLP weight matrices:

$$\begin{aligned} \boldsymbol{u_i} &= \text{LayerNorm}(\boldsymbol{x_i} + \boldsymbol{u_i'}; \gamma_1, \beta_1), \\ \boldsymbol{z_i'} &= \boldsymbol{W}_2^T \text{ReLU}(\boldsymbol{W}_1^T \boldsymbol{u_i}), \\ \boldsymbol{z_i} &= \text{LayerNorm}(\boldsymbol{u_i} + \boldsymbol{z_i'}; \gamma_2, \beta_2) \end{aligned} \tag{2.28}$$

The computation of the $\boldsymbol{z_i}$ output concludes a single Transformer layer. The input $\boldsymbol{x_i}$ and the output $\boldsymbol{z_i}$ of a Transformer layer are both shaped equally. This allows the composition of L Transformer layers, each with their own parameters $f_{\theta_1} \cdot ... \cdot f_{\theta_L}(x) \in \mathbb{R}^{n \times d}$, into a full Transformer model. Depending on the use case, a Transformer model can be conceived as an encoder-only, decoder-only, or encoder-decoder model. For example, the machine translation approach from Vaswani et al. (2017) uses an encoder-decoder model with $L = 6$ encoder and $L = 6$ decoder Transformer layers. Such a large number of layers (or even more layers) are possible since a Transformer is essentially a series of matrix multiplication that can be performed in parallel.

2.3.7.2 BERT

The introduction of the Transformer model (Section 2.3.7.1) started a new paradigm for NLP: The initial self-supervised pretraining of a Transformer-based language model with millions or billions of parameters on large amounts of unstructured text is followed by fine-tuning the model on a much smaller but supervised and task-specific dataset. Devlin et al. (2019) popularized this paradigm with their Bidirectional Encoder Representations from Transformers (BERT). The pretraining and fine-tuning approach from BERT builds upon ideas introduced by semi-supervised sequence learning (Dai and Le, 2015), ELMo (Peters et al., 2018), ULMFiT (Howard and Ruder, 2018), GPT (Radford et al., 2019), and other work by the NLP community.

Model architecture. BERT is a multi-layer bidirectional encoder-only Transformer model. Bidirectional refers to the model processing the input sequence in both directions, i.e., left-context and right-context. As opposed to the original Transformer (Vaswani et al., 2017), which is an

encoder-decoder model, BERT is conceived to only encode the input sequence and does not generate (or decode) an output sequence, i.e., it is an encoder-only model. Devlin et al. provide BERT models in various sizes (different number of attention heads, layers, etc.), whereby `BASE` typically refers to 110M parameters and `LARGE` to 340M parameters.

Model inputs. BERT uses WordPiece (Wu et al., 2016) to convert an input text into tokens of a fixed vocabulary. Similar to fastText Section 2.3.4.3, WordPiece splits text into sub-word tokens and, therefore, does not suffer from the OOV problem. The first token of every sequence is marked with the special `[CLS]` token and at the end or between two sequences the `[SEP]` token is used. For example, the input text of *I have a new GPU!* would be converted into:

```
[CLS], i, have, a, new, gp, ##u, !, [SEP]
```

Each of the tokens is assigned to a token embedding $\boldsymbol{x_i} \in \mathbb{R}^d$ which is learned during training. By itself, the Transformer has no notion of textual position, Devlin et al. address this by adding a positional embedding $\boldsymbol{p_j}$ to the token embedding before feeding it into the Transformer model. They also include a segment embedding $\boldsymbol{s_t}$ to distinguish between two separated input sequences (for the *next sentence prediction* task – see below). Hence, the model input $\boldsymbol{x'_i}$ is computed as:

$$\boldsymbol{x'_i} = \boldsymbol{x_i} + \boldsymbol{p_j} + \boldsymbol{s_t} \tag{2.29}$$

Pretraining objectives. To pretrain BERT in a self-supervised manner, Devlin et al. introduce two pretraining objectives: Mask language modeling (MLM) and next sentence prediction (NSP). For MLM, a small ratio of the input tokens are randomly masked (Devlin et al. report 15% as the best ratio) and the model is trained to recover the masked token, i.e, predict the original token. For NSP, the model is fed with two sentences A and B and needs to predict whether B is the actual next sentence of A. Specifically, the NSP task corresponds to a sequence pair classification with the label classes *"is next sentence"* and *"is not next sentence"*. Both pretraining objectives do not require any labeled data and can be automatically constructed from unstructured text.

2.3.7.3 Other Transformer language models.

Today's NLP research is dominated by large language models based on the Transformer architecture. Various modifications and extensions to the Transformer have been proposed but also discussed in recent surveys (Lin et al., 2021; Narang et al., 2021; Tay et al., 2022a). In this section, we discuss approaches to Transformer language models relevant to this thesis:

Other pretraining objectives. Given that pretraining accounts for the majority of the training costs, several studies have investigated other pretraining objectives as alternatives to MLM and NSP. Joshi et al. (2020) mask a contiguous segment of the input sequence instead of only masking each token independently. Lan et al. (2020) propose a sentence-order prediction task that focuses on inter-sentence coherence. Aroca-Ouellette and Rudzicz (2020) combine various token- and sentence-level pre-training objectives, e.g., sentence ordering or term-frequency prediction. Liu et al. (2019) propose RoBERTa and an optimized BERT model that discards the NSP task as a pretraining objective and also is trained with larger batch sizes and more data. With the decoder-only language model XLNet, Yang et al. (2019) introduce a pretraining objective based on token order permutations in which they mask out attention weights rather than tokens in the

input sequence. ELECTRA (Clark et al., 2020) has in addition to MLM the pretraining objective of detecting replaced tokens in the input sequence. For this objective, Clark et al. use a generator that replaces tokens and a discriminator network that detects the replacements, whereby the generator and discriminator are both Transformer models.

Scalability. One major drawback of the self-attention mechanisms in Transformers is that it has a quadratic complexity with respect to the sequence length. Due to this reason, most Transformer-based language models, like BERT, have a limited sequence length of 512 tokens. Longformer (Beltagy et al., 2020) uses a sparse attention pattern that combines local and global information and scales linearly with the sequence length. Linformer (Wang et al., 2020c) approximate self-attention using a low-rank matrix and also achieve a linear complexity. Other work about scalable Transformers is concerned with distributed model training (Narayanan et al., 2021; Shoeybi et al., 2019) or faster inference (Kim and Awadalla, 2020).

Domain adaptations & pretraining corpora. Pretraining a language model on a domain-specific text has been shown to improve the downstream task performance (Gururangan et al., 2020). For example, BERT was pretrained on the English Wikipedia and the BooksCorpus (Zhu et al., 2015). If a task is concerned with a domain different from these corpora, the performance may be suboptimal. Thus, several domain-specific variations have been proposed.

SciBERT (Beltagy et al., 2019) is a variation of BERT tailored for scientific literature, which is pretrained on computer science and biomedical research papers. Covid-BERT (Chan, 2020) is the original BERT model but fine-tuned on the CORD-19 corpus. BioBERT (Lee et al., 2019) is another BERT model specialized in the biomedical domain. Holzenberger et al. (2020) and Chalkidis et al. (2020) present both LegalBERT, i.e., a BERT model pretrained on legal text.

This concludes the review of text-based representation methods. In the subsequent section, we continue with the discussion of representation methods for graph information.

2.4 Graph-based Representations

The documents that this thesis investigates do not only contain textual content but are also interconnected with each other. More specifically, Web pages are connected through (hyper-) links, and scientific or legal literature is connected through citations. On a technical level, Web links and citations can be both considered as edges that connect individual documents in a graph of documents, as illustrated in Figure 2.6. Viewing documents not just as pieces of textual content but also as a graph has a long tradition in library science (Garfield, 1972; Garfield, 1955; Kessler, 1963; Marshakova, 1973; Price, 1965; Small, 1973) and has been shown to be beneficial in recent deep learning-based approaches (Perozzi et al., 2017; Perozzi et al., 2014).

Since the thesis uses graph information, this section introduces general concepts of document graphs and reviews graph-based methods for representing documents and their similarity.

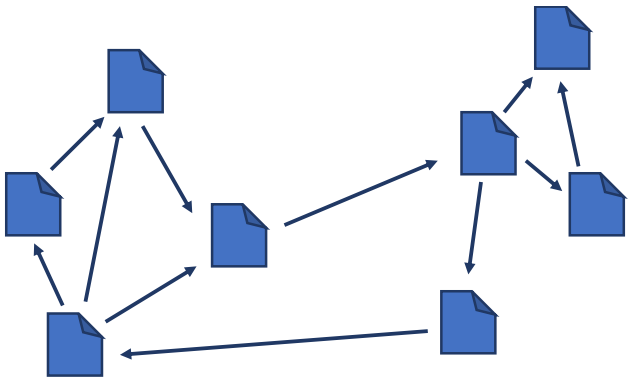


Figure 2.6: Illustration of a graph of documents. Documents are nodes that are connected through directed edges. An edge can be a citation or link. Its direction shows which document cites/links or is being cited/linked.

Graph terminology. A graph is a data structure denoted as $G = (V, E)$ that consists of a collection of vertices V (also called nodes, in our context, the vertices are typically documents) and a collection of edges E, represented as ordered pairs of vertices (u, v) (Cormen, 2009). As a generic data structure, graphs can represent various types of data. Next, we introduce two specific instances of graphs: citation graphs and Web link graphs.

Citation terminology. Citations are an essential part of research papers that reference prior publications (Smith, 1981). Following the definition from Egghe and Rousseau (1990), a reference in document B is a bibliographic note that describes document A. If document B contains a reference to document A, then A receives a citation from B. Stated otherwise, document B cites A, and A is cited by B (Figure 2.7). Scientific authors use citations to acknowledge concepts or methods that were used by the author (Smith, 1981), to express politeness and policy (Pasternack, 1969), and due to various other reasons (Teufel et al., 2006).

Link terminology. On the Web, links (or hyperlinks) are the equivalent of citations in the academic literature that connect Web pages with each other. However, the motivation for making a link on a Web page differs from the motivation behind citing a scientific article, even if their concepts may seem very similar (Thelwall and Wilkinson, 2004). In general, links and citations

serve the purpose of acknowledgment and are therefore analyzed for relevance judgments by many algorithms, e.g., PageRank (Page et al., 1998). Moreover, Web links are used for advertisements, navigational purposes, or in the context of link spam (Wu and Davison, 2005).

On a technical level, both citations and links represent edges in a graph. Hence, we refer in this thesis to *citations* or *links* when we want to highlight the current use case. When the discussion is use-case independent, e.g., in the context of a method, the terms *citations* and *links* are used interchangeably, or we refer to the generic graph terms of *edges* and *vertices*.

2.4.1 Direct Citations

Citations are often used as an indicator of semantic similarity. When two documents A and B (research papers or Web pages) are connected through a citation, A and B are generally assumed to be more semantically similar than two documents B and C that do not cite each other. Figure 2.7 visualizes this direct citation relationship of the documents A, B, and C. Since the similarity judgment depends on whether a citation exists or not, direct citations lead to a *binary* or *discrete* notion of similarity (similar or not).

Figure 2.7: Direct citations. Document A is cited by B, whereas B and C do not cite each other.

Despite the authors may cite without expressing similarity (Pasternack, 1969; Teufel et al., 2006), direct citations are often utilized as ground truth or gold standard for the evaluation of literature recommender systems. For example, the CiteSeer citation index[3] has been frequently used by researchers for evaluating research paper recommender systems (Caragea et al., 2013; Dong et al., 2009; He et al., 2010). Also, contrastive learning approaches like Citeomatic (Bhagavatula et al., 2018) or SPECTER (Cohan et al., 2020) rely on direct citations for the generation of positive and negative samples.

2.4.2 Bibliographic Coupling

Bibliographic coupling, introduced by Kessler (1963), measures the similarity of two documents as the number of their shared bibliographic items. Documents are *bibliographically coupled* if they cite one or more documents in common. The underlying assumption is that documents that cite the same literature are more likely to have the same subject. The degree of similarity is measured by the bibliographic coupling strength. For example, the coupling strength of two documents is three if they have three references in common. When documents do not share any references, their coupling strength is zero.

Even though its popularity in scientometrics (Jarneving, 2007), the bibliographic coupling method has been criticized in several ways. Martyn (1964) stated that bibliographic coupling indicates

[3] `https://csxstatic.ist.psu.edu/`, last accessed: 18/01/2023

a relationship between two documents but not necessarily their similarity. Small (1973) and Marshakova (1973) criticized as well the retrospective nature of bibliographic coupling. The references of a document do not change. Therefore, novel documents in the corpus are not reflected.

2.4.3 Co-Citations

The criticism of bibliographic coupling led Small (1973) and Marshakova (1973) to propose the co-citation similarity measure. Instead of focusing on the bibliography of the documents themselves, co-citations are concerned with the citations of two documents received by other documents. The number of papers citing two documents together defines the co-citation strength, i.e., the degree of similarity. Figure 2.8 illustrates one example in which the documents *A* and *B* have the co-citation strength of 2 since they are co-cited by *C* and *D*. The co-citation strength is influenced by how two documents are cited within the literature. Accordingly, co-citations are *prospective* in contrast to the retrospective bibliographic coupling.

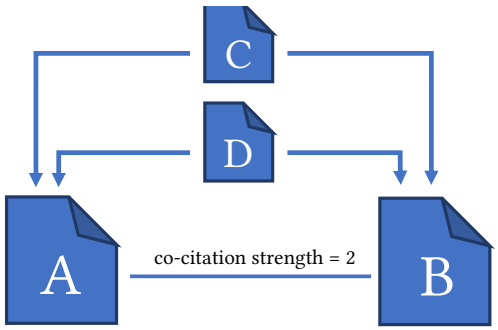


Figure 2.8: Co-citations. Documents *A* and *B* are *co-cited* by two other documents *C* and *D*, thus, *A* and *B* have the co-citation strength of 2. The co-citation strength is independent of whether *A* and *B* cite each other and is only defined through the external citations of *C* and *D*.

Various studies have shown the strength of co-citations for detecting similarities between research papers and other literature domains. For example, Chen (2017) conduct a systematic review of the scientific literature with the help of co-citation networks. Ferreira et al. (2016) rely on co-citations for clustering sub-fields of strategic management literature and for detecting emerging research topics. Jeong et al. (2014) identify authors working on related research topics by evaluating the co-citations of their publications. Woodruff et al. (2000) combine co-citations with textual information for a book recommender system.

2.4.4 Co-Citation Proximity Analysis

With co-citation proximity analysis (CPA), Gipp and Beel (2009) introduced an improvement over co-citations that utilizes the additional information provided by the citation marker and its position within the citing document. CPA is based on the assumption that when citation markers of co-cited documents are in close proximity, the documents are more likely to be similar (Figure 2.9). Using the additional proximity information has been shown to outperform the standard co-citation approach (Kim et al., 2016; Liu and Chen, 2011; Tran et al., 2009).

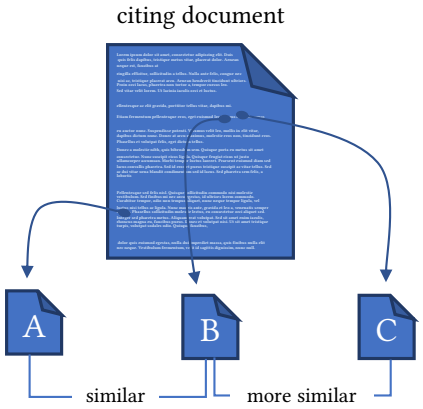


Figure 2.9: Co-citation proximity analysis (CPA). Documents *B* and *C* are more similar than *A* and *B* since their citation markers are located in close proximity.

CPA makes use of the increasing availability of full-text literature. Traditional citation indexes like Garfield (1964) did not provide access to the full-text information which is required to evaluate the proximity of citation markers.

To quantify the degree of similarity of co-cited documents, CPA assigns a numeric value, the co-citation proximity index (CPI), to each pair of documents co-cited in one or more citing documents. The CPI reflects the smallest distance between the citation markers of two co-cited documents within a citing document. In their original publication, Gipp and Beel distinguished five levels of co-citation proximity, each of which is assigned a static CPI: same sentence (CPI=1), same paragraph (CPI=$\frac{1}{2}$), same chapter (CPI=$\frac{1}{4}$), same journal issue or book (CPI=$\frac{1}{8}$), same journal, but a different issue (CPI=$\frac{1}{16}$). The CPA score is formed by summing up the proximity-weighted co-citations over all co-citing documents.

The static CPI definition of Gipp and Beel prohibits the application of CPA outside of the context of scientific literature. Therefore, our work on the similarity of Wikipedia articles (see Chapter 3) proposes a generalization of the citation proximity levels. In Schwarzer et al. (2016b), we define the link-position matrix $v_{i,j}$ of dimension $m \times m$ that stores the link position for all m documents. Specifically, the column for document $j, v_{*,j}$ holds the positions for links to other documents in words counted from the beginning of the document. Thus, the generalized CPI is defined as:

$$\mathrm{CPI}(a,b) = \sum_{j=1}^{m} \Delta_j(a,b)^{-\alpha} \tag{2.30}$$

$$\text{with } \Delta_j(a,b)^{-\alpha} = \begin{cases} |v_{a,j} - v_{b,j}|^{-\alpha} & v_{a,j} > 0 \wedge v_{b,j} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.31}$$

This definition states that for a document pair (a,b), the CPI is the sum of the proximity of their co-citations Δ_j, where the proximity is the link distance damped by an exponential tuning parameter α, which determines the influence of the distance. The hyperparameter α needs to be defined depending on the document type, i.e., the model needs to be optimized. Note that negative

values for α are counter-intuitive because a negative value of α would result in a weighting that prefers co-citations with a greater distance. Furthermore, the case of $\alpha = 0$ implies:

$$\text{CPI}(a,b) = \begin{cases} 1 & v_{a,j} > 0 \wedge v_{b,j} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.32}$$

In this specific case, CPI is independent of link distance and equivalent to the standard co-citation measure (Section 2.4.3) as only the number of co-citations is counted. Consequently, the proximity has no effect.

In Schwarzer et al. (2017), we extend the generalized CPI from Equation 2.30 with the inverse citation frequency:

$$\text{CPI}_{ICF}(a,b) = \sum_{j=1}^{|D|} \delta_j(a,b)^{-\alpha} * \log\left(\frac{|D| - n_a + 0.5}{n_a + 0.5}\right) \tag{2.33}$$

The first component is defined as in Equation 2.30. The second component is a factor that defines the specificity of article a based on the number of its received citations n_a. This factor is inspired by the Inverse Document Frequency of TF-IDF (Section 2.3.2), whereby we adapted the weighting schema from Okapi BM25 (Sparck Jones et al., 2000). Hence, we refer to the factor as inverse citation frequency that counteracts CPA's tendency to favor highly cited documents.

2.4.5 DeepWalk

The previously discussed methods, such as bibliographic coupling or CPA, are count-based methods that measure the pairwise similarity of documents. However, recent neural network-based approaches (Section 2.3.3) expect vector representations as their inputs, and, therefore, the count-based methods are suboptimal in the context of neural information processing. Moreover, vector representations have the advantage of being useful for applications other than similarity search, e.g., classification or clustering. This motivates graph-based representation learning that aims to derive a numerical vector representation $\boldsymbol{x} \in \mathbb{R}^d$ with d dimensions for each vertex v given the graph data $G = (V,E)$.

With DeepWalk, Perozzi et al. (2014) were the first to borrow word2vec's idea of learning word representations based on unstructured text (Section 2.3.4.1) and applying it to graph embeddings. DeepWalk utilizes truncated random walks on a graph to convert its graph data G into a sequence of vertices $v_1, v_2, ...v_c$ that can be modeled analog to word sequences in word2vec's Skip-gram approach. A random walk of length c and rooted at vertex v_i is a stochastic process that selects at random the next vertex v_{i+1} from the neighbors of vertex v_i. The random walk yields a sequence of vertices that can be thought of as short sentences generated by the context sliding window in word2vec. The representations for all vertices V, which corresponds to the vocabulary, can be learned with the objective of the Skip-gram model, which is to maximize the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log p(v_{t+j} | v_t) \tag{2.34}$$

where v_t is the target vertex, w_{t+j} is a vertex t walks apart from j, c is the length of the random walk. The training can then be efficiently performed analog to Mikolov et al. (2013b) with hierarchical softmax.

Perozzi et al. (2014) demonstrate DeepWalk's capabilities for classification tasks on several types of graph data, e.g., blogs, Flickr and YouTube. Other related studies rely also on DeepWalk. For instance, Chen et al. (2019b) and Guo et al. (2019) use random walks on the citation graph for citation recommendations. Li et al. (2017) use DeepWalk in a biomedical context to determine the similarity of diseases. Berahmand et al. (2021) extend DeepWalk with vertex attribute information for link prediction in social networks.

2.4.6 Walklets

Many real-world graphs are inherently hierarchical. For example, social networks reflect different scales from small (e.g., families) to medium (e.g., schools or companies) to large (e.g., nations). Similar hierarchical structures can be found in literature, e.g., research papers about a specific problem (small), papers from a field of study (medium), and a literature genre (large). Graph representations like DeepWalk neglect these multiple scales of relationships between vertices. Instead, they provide a "one-size fits all" approach where representations are independent of the hierarchy.

Walklets (Perozzi et al., 2017) explicitly encodes these multi-scale node relationships to capture hierarchical structures with the graph. Walklets learns a family of k successively coarser vertex representations $\boldsymbol{x}_1, \boldsymbol{x}_2, ...\boldsymbol{x}_k \in \mathbb{R}^d$ with d dimensions, where each $\boldsymbol{x}_k$ captures the view of the vertex at scale k. Perozzi et al. achieve the different scales by building upon DeepWalk but changing the sampling procedure of the random walks. Specifically, they choose to skip some of the vertices in the random walk, whereby the number of skipped vertices reflects the scale k. At inference time, the multi-scale representations can be leveraged (individually or combined) to provide a more comprehensive representation of the graph.

2.4.7 Other Graph Embeddings

Asides from DeepWalk and Walkets, many other techniques for graph embeddings have been proposed. For a comprehensive overview, we refer to the surveys from Cai et al. (2018), Goyal and Ferrara (2018), and Rahman and Azad (2021). Additional surveys from related topics are Wu et al. (2022), which reviews graph neural networks in the context of recommender systems, and Ji et al. (2022), which provides an overview of knowledge graphs and their representations.

Other prominent examples from graph embeddings are LINE (Tang et al., 2015), struc2vec (Ribeiro et al., 2017), and Node2Vec (Grover and Leskovec, 2016), which both improve upon DeepWalk (Perozzi et al., 2014). BoostNE (Li et al., 2019) is another technique to learn multi-scale vertex embeddings similar to Walklets (Perozzi et al., 2017) but it extends the matrix factorization approach with gradient boosting (Friedman, 2001).

2.5 Aspects in Information Processing

The term *aspect* originates from the Latin word *aspectus* that means "looking at" (Lewis, 1891). Thus, an aspect of something is the direction or perspective from which it is looked at. Following this meaning, the Merriam Webster dictionary defines an *aspect* as "a particular status or phase in which something appears or may be regarded".[4] Transferred to the context of this thesis, an aspect of a document is the perspective from which a user may look at the document's content.

In linguistics, the lexical aspect of a verb conveys information in which that verb is structured concerning time, and the grammatical aspect is an inherent feature of verbs or verb phrases, which is determined by the nature of the situation that the verb describes (Rothstein, 2016). In information processing research, the term "aspect" is less well-defined, leading to various alternative terms being interchangeably used in the literature. For example, aspects are also referred to as multi-senses (Mancini et al., 2017; Nguyen et al., 2017), multi-perspectives (He et al., 2015), facets (Mysore et al., 2021; Risch et al., 2021), or contexts (Hofmann et al., 2010). Depending on the research, these terms can have slightly different connotations and nuanced meanings while referring essentially to the same concept. For consistency reasons, we settle on the term of aspects throughout this thesis.

Table 2.1: Overview of literature concerned with aspects grouped by research task.

Task	References
Expert matching	Hofmann et al. (2010), Karimzadehgan and Zhai (2009), Karimzadehgan et al. (2008), Mirzaei et al. (2019), Pradhan and Pal (2020), Tang et al. (2010), and Zhang et al. (2020a)
Sentiment analysis	Alam et al. (2016), Basile et al. (2018), Chen et al. (2020), Ding et al. (2017), Do et al. (2019), Feng et al. (2019a), Hu and Liu (2004), Liu (2012), Liu et al. (2020), Nazir et al. (2022), Piryani et al. (2017), Pontiki et al. (2016), Pontiki et al. (2015), Pontiki et al. (2014), Poria et al. (2015), Poria et al. (2016), Rietzler et al. (2020), Ruder et al. (2016), Schouten and Frasincar (2016), Sun et al. (2019), Wang et al. (2016b), Xu et al. (2019), Yan et al. (2021), Zhang et al. (2021b), Zhang et al. (2022), and Zhang et al. (2021c)
Similarity	Bär et al. (2011), Bowman et al. (2015), Chen et al. (2018), Dagan et al. (2013), He et al. (2015), Nguyen et al. (2014a), Nguyen et al. (2018), and Saldias and Roy (2020)
Summarization	Amplayo et al. (2021), Angelidis et al. (2021), Cao and Wang (2021), Dang (2006), Fan et al. (2018), Frermann and Klementiev (2019), Kikuchi et al. (2016), Krishna and Srinivasan (2018), Maddela et al. (2022), and See et al. (2017)
Representation learning	Alshaikh et al. (2019), Camacho-Collados and Pilehvar (2018), Chan et al. (2018), Chen et al. (2019a), Jain et al. (2018), Kohlmeyer et al. (2021), Liao et al. (2020b), Mancini et al. (2017), Mysore et al. (2022), Nguyen et al. (2017), Risch et al. (2021), Schwarzenberg et al. (2019), and Zhang et al. (2021d)

[4] `https://www.merriam-webster.com/dictionary/aspect`, last accessed: 18/01/2023

This thesis's central topic, aspect-based document similarity, can be considered a niche task in the broader information processing literature. However, other related tasks also involve aspects. In the following section, we review these related tasks and how they consider aspects. An overview of the tasks and their corresponding literature is presented in Table 2.1.

2.5.1 Aspect-based Expert Matching

Finding the right expert is a problem that often occurs in the context of automatic peer reviewer matching (Wang et al., 2010). Given an item, which can be, for example, a submission to a publication venue and which is supposed to be reviewed by an expert, the objective is to assign the item to a reviewer with the best matching expertise. In this context, aspects play a crucial role since a single reviewer is potentially an expert in many topics and research areas. Also, the reviewed items can cover multiple aspects of a topic.

Karimzadehgan et al. (2008) were one of the first to criticize that most work on expert matching neglects "the multiple aspects of topics or expertise and all match the entire document to be reviewed with the overall expertise of a reviewer". The novel approach proposed by Karimzadehgan et al. accounts for aspects by modeling the sub-topics in the reviewer's documents and to be reviewed documents with probabilistic latent semantic analysis (Hofmann, 1999, PSLA) and matching both based on the latent sub-topic representations. Accordingly, this work considers aspects in the sense of sub-topics. Karimzadehgan and Zhai (2009) extend this multi-aspect approach with the constraint of review quota, making the approach more applicable to real-world scenarios. The constraint is enforced by casting the task as a linear programming problem that assigns reviewers based on the required expertise to review a document but also based on the coverage of the aspects of a document in a complementary manner subject to their review quota constraints. Tang et al. (2010) also work on constrained expert matching but include the authority of the reviewers in addition to topic and quota constraints. Technically, Tang et al. cast the problem to a convex cost flow problem.

Hofmann et al. (2010) relate aspects to users, i.e., they refer to the aspects as contextual factors about the experts and the expert-seeking users. For example, they find that important contextual factors are the topic of knowledge (working in the same area), organizational structure (position within the faculty), or familiarity (personal connection). The experiments from Hofmann et al. suggest combining user-related and content-based aspects yields the best results.

More recently, Mirzaei et al. (2019) introduced the idea of latent research areas for expert matching. Mirzaei et al. obtain latent research areas by clustering the term-based topic representations derived from the documents. Afterward, the matching is performed greedily based on the cosine distance between the latent research areas and the to-be-reviewed document. Zhang et al. (2020a) also treat research areas as the aspects of the expert matching task. However, they formulate the task differently, i.e., Zhang et al. perform a multi-label classification on the reviewer documents and the to-be-reviewed item, whereby the research areas act as label classes. The matching is then performed based on the similarity of the predicted labels. On the technical level, Zhang et al. rely on a recurrent neural network (Section 2.3.6) to encode documents. Instead of matching papers to reviewers, Pradhan and Pal (2020) match authors to potential collaborators. Their expert-to-expert matching approach accounts for a considerably larger number of aspects. For example, the matching from Pradhan and Pal is time-aware, authority-aware through H-Index, or considers prior collaborations.

In summary, the reviewed studies highlight the need to reflect the multi-aspect nature of documents and experts. All of the studies have found that combing multiple aspects is generally beneficial for task performance. However, the proposed approaches incorporate aspects only implicitly. For example, Mirzaei et al. (2019) cannot determine what exact research area is responsible for a match since they work with latent representations.

2.5.2 Aspect-based Sentiment Analysis

Aspect-based sentiment analysis is an extensively studied NLP task as several surveys show (Do et al., 2019; Liu et al., 2020; Nazir et al., 2022; Schouten and Frasincar, 2016; Zhang et al., 2022). In particular, a set of shared tasks contribute to the popularity of aspect-based sentiment analysis (Basile et al., 2018; Chen et al., 2020; Pontiki et al., 2016; Pontiki et al., 2015; Pontiki et al., 2014). Early work in sentiment analysis mainly focused on classifying the overall polarity of a written text. However, various applications require a more fine-grained approach. Specifically, the sentiment classification concerning a particular aspect (Liu, 2012). Given an example restaurant review "*The pizza is delicious, but the service is terrible*", it does not express an overall polarity but instead a positive sentiment towards the aspect *pizza* and negative sentiment towards *service*. In this example, the task would be to extract the aspect terms (*pizza* and *service*), the aspect category (e.g., food and people), and the corresponding sentiment polarities (*delicious* is positive, *terrible* is negative).

In early works on aspect-based sentiment analysis, noun phrase frequency-based approaches were used to extract aspect terms (Hu and Liu, 2004), where the assumption is that aspect terms are more likely to be repeated. This approach has the drawback that aspect terms may be implicitly stated as Hu and Liu find. Other approaches rely on rule-based methods (Piryani et al., 2017; Poria et al., 2016). But also topic modeling such as Latent Dirichlet Allocation (Blei et al., 2003, LDA) have been widely used (Alam et al., 2016; Poria et al., 2015). While the topic models are appropriated to detect aspects at the document level, the resulting topics are often too broad to reflect fine-grained aspects (Schouten and Frasincar, 2016).

Recently, deep learning-based token classification or sequence labeling approaches (similar to named entity recognition) have been the de facto standard for sentiment analysis. Feng et al. (2019a) and Ruder et al. (2016) utilize convolution neural networks, while Ding et al. (2017) and Wang et al. (2016b) apply LSTMs. Transformer language models are used in their vanilla form (Rietzler et al., 2020; Xu et al., 2019) and with task-specific modifications. For example, Sun et al. (2019) adopt BERT's sequence pair classification task for aspect-based sentiment classification, and Yan et al. (2021), Zhang et al. (2021b), and Zhang et al. (2021c) reformulate the task as a sequence-to-sequence generation task.

Even though the task of aspect-based sentiment analysis is conceptionally different from documenting similarity since only a single piece of text is considered, we still can borrow from the research findings of this task. Specifically, aspect terms may be implicitly stated (Hu and Liu, 2004), and topic models often do not reflect fine-grained aspects (Schouten and Frasincar, 2016). Moreover, Transformer language models can incorporate aspect information, e.g., through the sequence pair classification task as Sun et al. (2019) have shown (see Chapter 6 and 7).

2.5.3 Aspect-based Summarization

Text summarization approaches, both extractive (Ruan et al., 2022; Zheng and Lapata, 2019) and abstractive (Cachola et al., 2020; Calizzano et al., 2022), follow the assumption that salient content from single or multiple input documents is relevant (Erkan and Radev, 2004) and should be part of the generated summary. However, the notion of salience largely depends on user interest. For example, a user might only care about the *food* aspect in the summarization of restaurant reviews (Section 2.5.2). Consequently, summarization approaches have been proposed that control the summarization output such that the aspects a user is interested in are reflected. The literature also refers to these approaches as *controllable summarization*.

What is considered as an aspect is inconsistent in the summarization literature. There is approaches focussing on the output length (Kikuchi et al., 2016), textual style (Cao and Wang, 2021), or entities (Fan et al., 2018; Maddela et al., 2022). However, more relevant in the context of this thesis are studies about semantic aspects. The shared task from Dang (2006) aims to generate summaries that answer a complex question. Moreover, Dang distinguish between the granularity of the summary, e.g., general background information or specific details. Frermann and Klementiev (2019) propose an approach in which the input documents are segmented into aspect-specific parts, whereby aspects are perspectives in product reviews (e.g., product features) or topics in news articles (e.g., sports and politics). Technically, Frermann and Klementiev represent aspects as one-hot vectors and treat them as part of the vocabulary. The abstract summaries are then generated with sequence-to-sequence Pointer Generator network (See et al., 2017). Krishna and Srinivasan (2018) follow a similar approach but concatenate the one-hot aspect vectors to the word embeddings before feeding them into their model.

The abstractive approach from Amplayo et al. (2021) lets users provide one or more query aspects that control the output summary. The aspects are represented as a small set of words, e.g., for hotel reviews, the words could be *food*, *location*, or *cleanliness*. Amplayo et al. incorporate the aspects as special tokens in the input of a sequence-to-sequence language model. Angelidis et al. (2021) propose an extractive approach based on clustering. Based on individual sentences of the input documents, Angelidis et al. construct aspect-specific clusters, whereby the aspects are defined by query terms (similar to Amplayo et al.). To obtain the output summaries, the summary sentences are extracted only from those aspect-specific clusters.

The discussed aspect-based summarization approaches commonly rely on a fixed set of aspects that typically present a topic or question and correspond to a set of words. Having this word-based aspect definition requires the aspects to be explicitly mentioned in the input documents. As Hu and Liu (2004) already suggested, this can lead to suboptimal results.

2.5.4 Aspect-based Representations

In traditional representation learning (see Section 2.3 and 2.4), an item is commonly represented as a single point in the embedding space, i.e., a monolithic vector. Having a single point follows the geometric understanding of similarity (see Section 2.2.2). But this entangles the many aspects or meanings that an item can represent and makes them indiscriminative (Camacho-Collados and Pilehvar, 2018).

In contrast to this, alternative representation learning approaches incorporate aspect information. These aspect-based or multi-sense representations align with the similarity model of conceptual

spaces from Gardenfors (2004); see Section 2.2.2. Since these approaches disentangle monolithic vectors into aspect-specific subvectors, the literature also refers to them as *disentangled representation learning*. According to Higgins et al. (2018), disentangled representations are characterized by "the decomposition of a vector space into independent subspaces".

Even at the word level, vector representations should account for the different aspects, meanings, or senses a word can represent. Mancini et al. (2017) embed words and their senses in a joint vector space. Given a semantic network like BableNet (Navigli and Ponzetto, 2012), they connect each word in the corpus with zero, one, or more senses. Then, Mancini et al. extends the CBOW model (Section 2.3.4.1). In addition to the input of the word context window, they also have the sequence for the associated senses as input, which are then learned in the same ways as the word embeddings. Nguyen et al. (2017) propose to learn word embeddings as a weighted mixture of their sense embeddings. Their mixture model derives senses from the topics of an LDA topic model and adapts the Skip-gram model (Section 2.3.4.1). Alshaikh et al. (2019) learn *conceptual spaces* of word embeddings through clustering. Schwarzenberg et al. (2019) project word vectors into a concept space in which the dimensions correspond to predefined concepts. Liao et al. (2020b) formulate disentangled representations as a feature selection problem, whereby they transform the original word embeddings into six sub-spaces (the aspects are *artifact*, *location*, *animal*, *adjective*, *adverb*, and *unseen*). The underlying aspect labels are taken from WordStat[5]. Furthermore, Liao et al. emphasize two general advantages of aspect-based representations: First, the separate encoding of aspects intuitively allows manual examination. Second, each sub-space provides informative features, which one can select or discard specific sub-spaces depending on the downstream task. The survey from Camacho-Collados and Pilehvar (2018) summarizes additional aspect-based word embedding techniques.

In this thesis, the sentence- or document-level representations are of more relevance. Jain et al. (2018) were one of the first to introduce disentangled representations to documents. In their work about biomedical abstracts, they learn disentangled embeddings for the aspects of *populations*, *interventions*, and *outcomes* for clinical trials. For each aspect, Jain et al. train an aspect-specific encoder (CNN with gated token activations) on maximizing the similarity of documents, which are similar in the given aspect, and minimizing the similarity of dissimilar documents. The aspect-based similarity of documents is given as triplets (query, positive, and negative sample). The triplets are derived from an annotated aspect-based sentiment dataset (Section 2.5.2) or from aspect-based summarizations (Section 2.5.3). The results from Jain et al. suggest that their approach induces aspect-specific document representations, which are qualitatively interpretable and outperform the baselines in information retrieval tasks. Risch et al. (2021) present a similar approach, also focussed on the biomedical domain but relying on entity categories from a knowledge graph as aspect labels and an RNN encoder. Chen et al. (2019a) disentangle the syntax and the semantics within sentence representations by designing separate loss functions that address either syntax (paraphrases) or semantics (word positions). Zhang et al. (2021d) apply aspect-specific masking on weights and hidden activations of a BERT language model (Section 2.3.7.2) to obtain aspect-based representations. Zhang et al. show that their approach can disentangle aspects such as syntax from semantics and sentiment from genre.

Another line of work follows the principle of dividing a document first into aspect-specific segments and then computing the representation separately on a segment level. Chan et al. (2018) annotate the aspects of research papers in their abstracts, whereby *background*, *purpose*,

[5] `https://provalisresearch.com/`, last accessed: 18/01/2023

mechanism, and *findings* are considered as the aspects. Based on these annotations, Chan et al. found in a user study that the segment-level representations helped their participants in finding analogies between research papers more efficiently. Huang et al. (2020) apply the same segmentation approach as Chan et al. but to biomedical research papers. Kobayashi et al. (2018) classify sections into discourse facets and build document vectors for each facet. Mysore et al. (2022) address the problem of obtaining labels for aspect-based similarity by considering the context in which papers are co-cited as a supervised learning signal. Kohlmeyer et al. (2021) extract aspect words in books (each representing one or more of the aspects *location*, *time*, *style*, *atmosphere*, or *plot*) and then construct aspect embeddings with Paragraph Vectors.

But also aspect-based representations for other data structures or modalities are subject to research, e.g., knowledge graphs (Zhang et al., 2021a), audio (Luo et al., 2019) or images (Kulkarni et al., 2015; Locatello et al., 2019).

As the discussed literature suggests, a key challenge of aspect-based representation learning is the dependency on aspect information. The aspect information is either provided in the form of annotated data, which is costly to collect or is constructed with the help of silver standards (e.g., citations or data from unrelated tasks). Both sources provide a noisy learning signal. While for words various resources exist, such data is scarce for documents. Moreover, we see two major competing approaches, aspect-specific encoding and segmentation. Aspect-specific encoding is prefereable over segmentation, since splitting documents into segments breaks the document coherence and can hurt the performance of NLP models as Gong et al. (2020) showed.

2.5.5 Aspect-based Text Similarity

Aspect-based text similarity is the task of determining the similarity of two texts concerning one or multiple aspects. More formally, the task is about finding a function $\delta(d_a, d_b, a_i) = s$ that assigns for the texts d_a and d_b and the aspect a_i a similarity value $s \in \mathbb{R}$. We define aspect-based similarity as a general pairwise text comparison task without further restriction on what can be considered an aspect.

One example of a common NLP task that falls into this definition is the task of textual entailment or natural language inference (Bowman et al., 2015; Dagan et al., 2013). Textual entailment recognition involves assessing whether a given textual premise entails or implies a given hypothesis, i.e., while d_a and d_b are the premise and hypothesis respectively, the aspects are *entailment*, *contradiction*, and *neutral*. State-of-the-art approaches for textual entailment rely on large language models (Wang et al., 2021) and incorporate external knowledge such as lexical information (Chen et al., 2018).

Aside from textual entailment, aspect information is often neglected in NLP studies about text similarity. Bär et al. (2011) find that similarity is often ill-defined and just used as an "umbrella term covering quite different phenomena". Bär et al. suggest considering *content*, *structure*, and *style* as are the major aspects of textual similarity. The need for a more nuanced view of similarity is highlighted by the study of Nguyen et al. (2014a). In their study about narrative similarity, Nguyen et al. find that the different user groups differently perceive the similarity. Specifically, experts focus on the similarity of the plot, characters, and themes of narratives, whereas non-experts tend to perceive similarity instead in the context of genre and style. In a related study, Saldias and Roy (2020) also found that narrative similarity is perceived differently. One of the few NLP papers addressing a more nuanced view of similarity is the work from He

et al. (2015) in which sentence similarity is modeled from *multiple perspectives*. He et al. apply convolution filters with multiple granularities and window sizes to extract intermediate features, each representing one perspective. However, He et al. do not express aspects explicitly since the final sentence representations are pooled over individual aspect representations. Nguyen et al. (2018) provide an overview of additional methods for aspect-based text similarity.

Generally, the approaches discussed under Section 2.5.4 also apply to aspect-based text similarity. To be more precise, one can construct aspect-specific embeddings and then simply compute their vector similarity (Section 2.2.4).

2.6 Summary of the Chapter

This chapter introduced background knowledge and related work relevant to this thesis. We discussed recommender systems as the central application of this thesis and reviewed relevant recommender approaches including their strengths and weaknesses. In particular, we focussed on the two categories of user-based and content-based recommender systems. User-based approaches such as collaborative filtering face problems like the cold start problem or filter bubbles. To mitigate these problems or to circumvent the lack of user data, content-based approaches are applied. Especially in the context of digital libraries and literature recommendations, content-based approaches are commonly used as our review of related work showed. To reflect the diverse types of literature available in digital libraries, we reviewed existing works for recommending research papers, citations, legal documents, and books. We found that most works rely on aspect-free similarity and apply similarity measures without further specifications or detailed discussion about the meaning of similarity.

This chapter also reviewed similarity as a concept including theories and models of similarity from philosophy, psychology, and information theory. We found competing concepts and definitions of similarity and that there is no consensus on a single unified definition of similarity. Moreover, our review showed that the idea of aspect-based similarity is already integrated with philosophical concepts of similarity, e.g., the feature similarity model from Tversky (1977).

To determine document similarity, document semantics must be represented meaningfully. In particular, we reviewed machine learning-based representation methods distinguishing between text-based and graph-based representations. Our review of the methods has been conducted to support the analysis of our experimental results in the subsequent chapters. Throughout the review, we found that the recent methodological progress enables increasingly complex downstream tasks making aspect-based document similarity feasible in terms of the underlying methods.

To get inspired about approaches to integrate aspect information, we also reviewed related work about other aspect-based NLP tasks. In particular, aspect-based sentiment analysis is an extensively studied NLP task from which we can borrow methods like Sun et al.'s sequence pair classification approach. Similarly, we saw challenges like the lack of publically available gold standards and the need for constructing silver standards being recognized by the related work.

With the presentation of background knowledge and the discussion of related work, this chapter laid the foundation for the research of this thesis. The subsequent chapters revisit these foundations, e.g., Chapter 3 evaluates classical text-based and graph-based representations, Chapter 4 focusses on neural representations, or Chapter 8 implements aspect-based representations.

Part II

Aspect-free Document Similarity

Chapter 3

Wikipedia Article Recommendations

The previous chapter introduced and reviewed existing document representations and similarity measures. Based on these insights, this chapter contributes to Research Task **I** with an empirical evaluation of document similarity measures for Wikipedia article recommendations. This chapter's content is based on three publications (Ostendorff et al., 2021b; Schwarzer et al., 2017; Schwarzer et al., 2016b).

"Evaluating Link-based Recommendations for Wikipedia" by **Malte Schwarzer**, Moritz Schubotz, Norman Meuschke, Corinna Breitinger, Volker Markl, and Bela Gipp. In: *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries (JCDL)*, 2016.

"Citolytics: A Link-based Recommender System for Wikipedia" by **Malte Schwarzer**, Corinna Breitinger, Moritz Schubotz, Norman Meuschke, and Bela Gipp. In: *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys)*, 2017.

"A Qualitative Evaluation of User Preference for Link-Based vs. Text-Based Recommendations of Wikipedia Articles" by **Malte Ostendorff**, Corinna Breitinger, and Bela Gipp. In: *Towards Open and Trustworthy Digital Societies: 23rd International Conference on Asia-Pacific Digital Libraries (ICADL)*, 2021.

We select Wikipedia as our research subject and the generation of Wikipedia article recommendations as the task we want to evaluate. Wikipedia is a large and rapidly growing digital library. As of August 2022, all language-specific versions of Wikipedia combined contain approximately 59 million articles, of which six million are in English.[1] Wikipedia has grown by approximately 17,000 articles per month. On average, all Wikimedia projects received 22 billion page views (crawlers excluded) per month in 2022.[2] Despite Wikipedia's size, popularity, and rapid growth, little research has addressed the issue of improving information search in Wikipedia through an automated generation of article recommendations. When conducting the following experiments, Wikipedia relied entirely on manually created and curated links to related articles.

Our study compares co-citations (Section 2.4.3) to its proximity-weighted enhancement CPA (Section 2.4.4). We modify both approaches such that they use the internal Wikipedia links instead of citations to measure the similarity of Wikipedia articles. To complement the two graph-based methods, we also include the text-based MoreLikeThis (MLT) from the Apache Lucene framework, which is a widespread implementation of the vector space model and TF-IDF (Section 2.3.2). MLT or generally TF-IDF are commonly used by related work, e.g., Collins and Beel (2019), Ollivier and Senellart (2007), and Tran et al. (2009).

[1] `http://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia`, last accessed: 18/01/2023
[2] `https://stats.wikimedia.org`, last accessed: 18/01/2023

The study is conducted in two parts: (1) An offline evaluation based on click stream data and "See also" links as silver standards that enable a large-scale quantitative comparison of the methods. (2) A user study on a smaller subset of articles and their recommendations.

In the first part, we compare the three methods regarding the general ability to produce meaningful Wikipedia article recommendations. The offline evaluation aims to test whether co-citations and CPA can be transferred to links in Wikipedia articles despite being originally developed for citations. Moreover, the offline evaluation acts as a filter to select the two best methods that we will evaluate in the subsequent user study.

The second part's purpose is to answer a question that remains largely unexplored in today's literature: Are fundamentally different classes of recommendation algorithms, e.g., text- and graph-based methods, also perceived differently by users? If a noticeable difference can be observed among users, across what dimensions do the end-users of such recommendation algorithms perceive that the approaches differ for a given recommendation use case? Most studies dedicated to evaluating recommender systems use offline evaluations using statistical accuracy metrics or error metrics without gathering any qualitative data from users in the wild (Beel et al., 2016b). More recently, additional metrics have been proposed to measure more dimensions of user-perceived quality for recommendations, e.g., novelty (Gravino et al., 2019; Mendoza and Torres, 2020), diversity (Kunaver and Požrl, 2017; Yu et al., 2009), serendipity (De Gemmis et al., 2015; Ge et al., 2010; Kaminskas and Bridge, 2017; Kaminskas and Bridge, 2014), and overall satisfaction (Joachims et al., 2005; Maksai et al., 2015; Zhao et al., 2018). However, empirical user studies examining the perceived satisfaction with recommendations generated by different approaches remain rare. Given the emerging consensus on the importance of evaluating recommender systems from a user-centric perspective beyond accuracy alone (Ge et al., 2010), we identify a need for research to examine the user perception of fundamentally different recommendation classes.

Therefore, we perform a qualitative study to examine user-perceived differences and thus highlight the benefits and drawbacks of two contrasting recommendation approaches for Wikipedia articles. Specifically, the user study seeks to answer the following three research questions:

Research questions

RQ1: Is there a measurable difference in users' perception of the graph-based approach compared to the text-based approach? If so, what difference do users perceive?

RQ2: Do the approaches address different user information needs? If so, which user needs are best addressed by which approach?

RQ3: Does one approach show better performance for certain topical categories or article characteristics?

The remainder of this chapter is structured as follows: First, we introduce the general experimental methodology, i.e., the datasets, the evaluated methods, and the user study design. Subsequently, we present the results of offline evaluation in Section 3.2 and the results of user study in Section 3.3. Finally, we summarize the main findings of this chapter.

3.1 Methodology

The following section describes the methodology used for evaluating Wikipedia article recommendations with text-based and graph-based document similarity measures.

3.1.1 Dataset

Our dataset is a data dump of the English version of Wikipedia. The data dump was created in September 2014, consists of 4.6 million Wikipedia articles in XML (Wiki markup), and has a size of 99 GB. To get an overview of the dataset's composition and to enable a comparison with other collections, we present information on article length and the number of in-links. Figure 3.1 shows the distribution of words and in-links among articles.

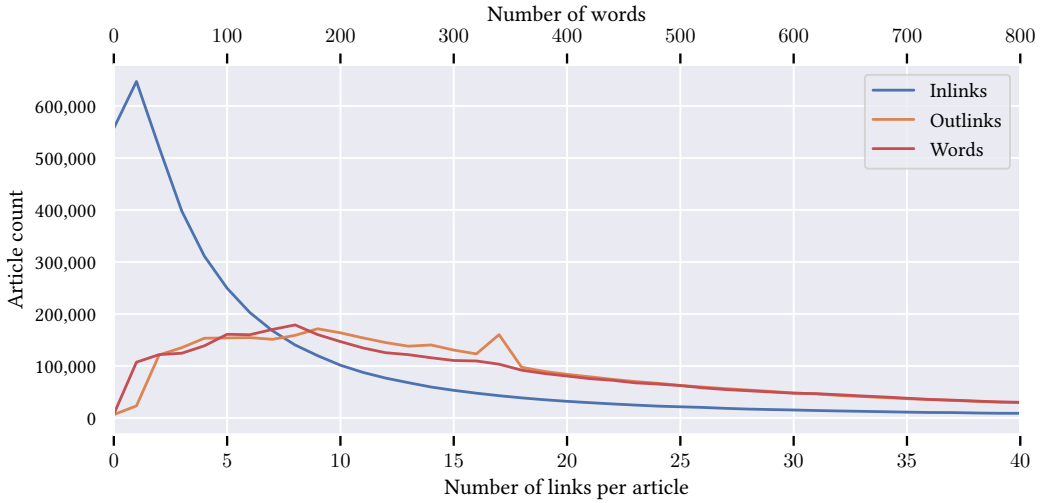


Figure 3.1: Distribution of word frequency (red), in-links (blue), and out-links (orange) of articles in our Wikipedia dataset.

On average, an article contains 740.54 words. The longest article contains 75,178 words. There is a consistently strong correlation between the number of out-links and the number of words for all article lengths. However, the distribution of in-links is heavily skewed. About 1.7 million of the 4.6 million articles have less than three in-links. On average, an article has 20.5 in-links. The most-linked article is "United States", which receives 392,494 in-links. As reported by Bellomi and Bonato (2005), Wikipedia articles with a high number of in-links are mainly about geopolitical topics, famous people, abstract nouns, or common words.

Our goal is a large-scale evaluation of the performance of similarity measures in recommending semantically similar Wikipedia articles. Instead of selecting several topics and defining topic-specific information needs, we want to understand how well the methods perform for the entire Wikipedia with its vast range of topics. Therefore, we define for our study a generalized information need: Recommend Wikipedia articles that interest a reader of the source article.

3.1.1.1 Silver Standards

Given the scope of our study, we require judgments for relevance that suit our information need, are available for large parts of the dataset and a broad range of topics, and are publicly accessible.

We derive two silver standards satisfying these requirements from analyzing (a) "See also" links and (b) clickstream data.

Unlike user studies, which are typically limited to a few hundred articles at most, these data sources allow an evaluation for 779,716 articles using "See also" links and 2.57 million articles using the clickstream dataset. Nonetheless, this evaluation approach has its shortcomings. "See also" links and clickstream data are only approximated relevance judgments. Therefore, we refer to them as silver standards, not gold standards. A silver standard is an approximation of a 'perfect' reference model. We are the first to apply both silver standards in the context of such evaluation.

"See also" links. A unique characteristic of Wikipedia articles is not only that they contain links to additional information in the form of internal references or external links but also that they contain so-called "See also" sections. The purpose of these sections is to provide links to other relevant Wikipedia articles, which results in these links acting as literature recommendations for readers.[3] Correspondingly, "See also" links are equivalent to a silver standard that allows a performance evaluation of a recommendation system. Therefore, we classify articles as relevant if the recommended article is listed in the "See also" section and as irrelevant otherwise. However, it is in this second assumption that we see a problem: We expect the "See also" links to be an incomplete reference model created by a few Wikipedia editors. We assume that the main objective of Wikipedia editors lies in creating textual content rather than providing useful literature recommendations, which means that if a recommended document is not included in the "See also" links, it can still be semantically similar and relevant to the readers. Therefore, we can only decide if a result is relevant but not if it is irrelevant. A true binary classification is not possible. Hence, we expect a precise true positive classification for articles that exist as "See also" links. However, many results could be classified as false negatives, even if a result is truly relevant because the recommendation is missing in the "See also" links.

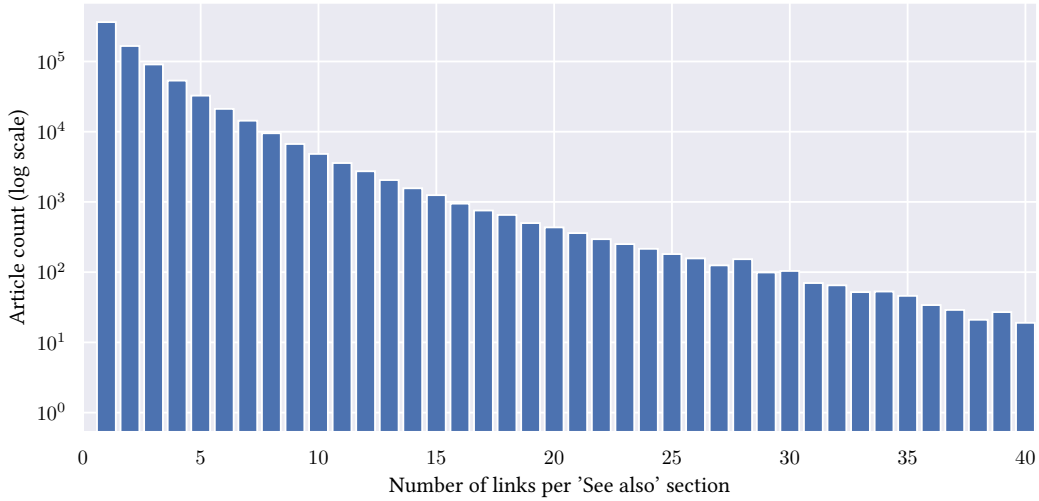


Figure 3.2: Number of links per "See also" section in Wikipedia articles.

[3] `https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Layout`, last accessed: 18/01/2023

We automatically extract the "See also" section and its links. Figure 3.2 shows the distribution of the number of "See also" links in our Wikipedia dataset. The dataset contains 779,716 Wikipedia articles with "See also" sections (17% of all articles), where each section contains 2.6 links on average. This low number of links per section additionally contributes to the incompleteness of relevance judgments, since the number of relevant recommendations that could be made from the Wikipedia corpus is likely greater than the number of available "See also" links.

Clickstreams. The publication of Wikipedia clickstreams by WikiResearch (Wulczyn and Taraborelli, 2015) allows us to use a second silver standard. The dataset contains clickstreams for 2,572,063 articles (56% of all articles) in the form of aggregated HTTP referrer information during the month of February 2015. The HTTP referrer indicates the page from which a user clicked to the article in question. Using this data, we can determine the number of clicks on out-links for articles. For outgoing links, which occur multiple times in an article, only the total number of clicks is provided. WikiResearch cleaned the dataset from computer-generated clicks (bot activity). However, Wulczyn and Taraborelli have noted that the filtering of bots chould be improved.We assume that the dataset contains some noise from bot activity, but we cannot quantify or reduce the noise level, since only aggregated clickstream data is available to us. In the future, WikiResearch plans to release more datasets, which would increase the value of clickstreams as a reference model. We consider the number of clicks on a link as a cardinal relevance classification regarding the linked article. The more often a link is clicked, the more relevant we assume the article to be. Whether this assumption holds true for all articles, and whether it is a major force driving clicks, has not been proven. Other factors can also affect the number of clicks, such as the descriptiveness value of the link, or the link's position within the article. A recently published study showed that the Click-Through-Rate decreases in proportion to the link's position from the top.[4]

The two silver standards differ in their conceptual properties: While the "See also" silver standard is created by Wikipedia editors; clicks are judgments for relevance by all readers. Moreover, clicks can only occur on links that exist in the article content. Such in-content links are also included for navigational purposes, while "See also" links are exclusively literature recommendations. The Wikipedia manual states to only add links in "See also" sections that do not exist in other parts of the article.

3.1.2 Evaluated Methods

We evaluate three methods: MoreLikeThis (MLT), Co-Citations (CoCit), and Co-Citation Proximity Analysis (CPA). CoCit can be considered as a special case of CPA (Section 3.2.1) and is only used in the offline evaluation. For the sake of transparency and reproducibility, we publish the source code used in our study on GitHub.[5]

MoreLikeThis. MoreLikeThis (MLT) is an implementation of TF-IDF (Section 2.3.2). To generate the MLT result sets, we use a Java application and an Elasticsearch cluster. The application consists of four sub-tasks: extracting articles from the Wikipedia XML dump, indexing them to Elasticsearch, performing the MLT queries, and storing all results as a CSV file.

[4] `http://meta.wikimedia.org/wiki/Research:Wikipedia_clickstream`, last accessed: 18/01/2023
[5] `https://github.com/wikimedia/citolytics`, last accessed: 18/01/2023

Co-Citation Proximity Analysis. We implement the CPA algorithm with the Apache Flink framework (Alexandrov et al., 2014) using the Java programming language. In contrast to MLT, CPA does not require an indexing process. Instead, the CPA results are directly generated from the Wikipedia XML dump. This requires extraction of the full link graph and performing CPI computation. These operations are expressed in the MapReduce programming model (Dean and Ghemawat, 2008). For completeness, we also resolve redirections for Wikipedia links that do not point directly to their destination.

The static classification of CPI values originally proposed by Gipp and Beel (2009) is unapplicable to our dataset. Wikipedia articles are not organized in journals, nor do they follow the structure of scientific documents. Thus, we introduce a new dynamic model of CPI that can be adjusted depending on the requirements of the test collection. Specifically, we redefine CPI as described in Equation 2.30 and use this CPI implementation in the offline evaluation. However, for the user study, we further modified the CPI to reflect the findings of the offline evaluation. The offline results revealed a tendency towards frequently linked Wikipedia articles. To account for that, we extend the generalized CPI with the inverse citation frequency, as described in Equation 2.33. The inverse citation frequency is inspired by the Inverse Document Frequency of TF-IDF (Section 2.3.2), whereby we adapted the weighting schema from Okapi BM25 (Sparck Jones et al., 2000). Section 2.4.4 presents details about the formal definitions of the CPI.

3.1.3 Evaluation Methdology

Each silver standard is evaluated separately to ensure that all Wikipedia articles contribute equally to the results, independent of an article's number of "See also" links or its popularity.

3.1.3.1 Evaluation Metrics

In the "See also" evaluation, we use the rank-based Mean Average Precision (MAP) score to quantify recommendation quality (Section 2.1.3). We calculate MAP for the 10 top-ranked results, i.e. $k = 10$. All articles are weighted equally in the final MAP score regardless of the article's number of "See also" links.

We also performed experiments that calculated the performance measure Mean Reciprocal Rank (MRR) in addition to MAP during the "See also" evaluation. Evaluating the approaches according to MAP or MRR yielded no significant differences in the performance relation of the approaches. Therefore, we chose to only report MAP results in this chapter, since we consider MAP as more representative of the performance of an approach with regard to all results.

In the clickstream evaluation, we measure recommendation performance using the Click-Through-Rate measure (CTR) for the top-k-results with k set to 1, 5, and 10 respectively (Section 2.1.3). Popular Wikipedia articles can generate more clicks than niche articles. Nevertheless, we followed the approach of equally weighting each article independent of its popularity.

"See also" Evaluation. We collect the data for the "See also" silver standard from the Wikipedia dump by filtering for sections titled "See also" and extracting the sections' links. We map the resulting dataset with the MLT and CPA results based on the article name. Lastly, we ensure that a "See also" link exists for each seed article.

Clickstream Evaluation. The data required for the clickstream evaluation is obtained from Wikiresearch as a CSV file. Therefore, no pre-processing is required. We assign the clickstream data to CPA and MLT results, i.e., we assign each article recommendation to the respective number of clicks on the link and its CTR. In the final evaluation process, we combine all result sets with the corresponding silver standards.

3.1.3.2 Computing Infrastructure and Runtime

The experiment is performed on a cluster of 10 IBM Power 730 (8231-E2B) servers. Each machine had two 3.7 GHz POWER7 processors with 6 cores (12 cores in total), 2 x 73.4 GB 15K RPM SAS SFF Disk Drive, 4 x 600 GB 10K RPM SAS SFF Disk Drive and 64 GB of RAM.

Table 3.1: Approximated runtimes for each task.

Task	Runtime
MoreLikeThis (Elasticsearch)	
· Indexing	7:30 hrs
· Retrieval	53:45 hrs
Co-Citation Proximity Analysis (Apache Flink)	
· Computing Results	7:45 hrs
Evaluation (Apache Flink)	
· See also links	0:45 hrs
· Clickstream	0:50 hrs

We rely on Apache Flink v0.8 and Hadoop v2.4.1. MLT is implemented using Elasticsearch v1.4.2. All versions were the latest stable releases at the time of the experiment. We use the software's default settings, i.e., neither Apache Flink nor Elasticsearch are optimized for runtime performance. Although we did not focus on runtime performance and none of the tested document similarity measures had been optimized, the difference in runtime between CPA and MLT, as listed in Table 3.1 shows that MLT involves a more extensive computation than CPA. This is conceptually obvious since the data volume for the recommendations based on words vs. links differs significantly. Also, MLT requires additional cleaning techniques such as stop word removal and TF-IDF weighting.

3.1.4 User Study Design

This section describes our user study methodology and the criteria for selecting the Wikipedia articles used in the study.

3.1.4.1 Study Design

Figure 3.3 shows the study design, including the seed articles used for recommendation generation, and the resulting data collected during the study.

Prior to our study, we create a sample of 40 seed articles covering a diverse spectrum of article types in Wikipedia. When selecting these seed articles, our aim was to cover diverse topics

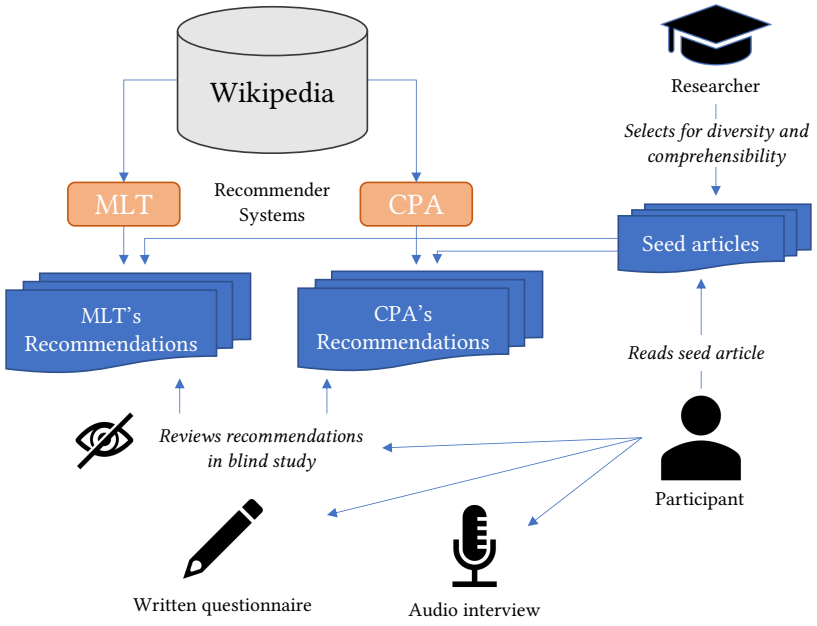


Figure 3.3: Overview of study design.

that nonetheless remain comprehensible to a general audience. To ensure comprehensibility, we exclude topics that would require expert knowledge to judge the relevance of recommendations, e.g., articles on mathematical theorems or regional historical events. Moreover, the seed articles featured diverse article characteristics, such as article length and article quality.[6] Annual page view statistics are measured with Wikipedia's page view tool[7] provided by the Wikimedia Foundation and are for the date range November 2016 - October 2017 (aligned with the dataset).

We distinguish the seed articles into four categories. First, according to their *popularity* (measured by page views) into either niche or popular articles, and second, according to the content of the article into either *generic*, i.e., reference articles typical of encyclopedias, or *named entities*, i.e., politicians, celebrities, or locations. We choose popularity as a criterion because, on average, popular articles receive more in-links from other articles. The results of the offline evaluation suggest that the number of in-links affected the performance of the graph-based CPA approach (Section 3.2). Moreover, we expect study participants to be more familiar with popular topics compared to niche articles. Therefore, users will be better able to verbalize their spontaneous information needs when examining a topic.

The 'article type' categories are chosen to study the effect that articles about named entities may have on MLT. Names of entities tend to be more unique than terms in articles on generic topics. Therefore, we expect that specific names may affect MLT's performance. Likewise, due to the nature of Wikipedia articles linking to generic topics, they may appear in a broader context than links to named entities. Thus, CPA's performance may also be affected. These considerations resulted in four article categories: (A) niche generic articles, (B) popular generic articles, (C)

[6] We judge the article quality using Wikipedia's vital article policy. `https://en.wikipedia.org/wiki/Wikipedia:Vital_articles`, last accessed: 18/01/2023

[7] Wikipedia Pageviews Analysis. `https://pageviews.toolforge.org/`, last accessed: 18/01/2023

niche named entities, and (D) popular named entities. Table 3.2 shows these four categories and the 40 seed articles selected for recommendation generation.

To perform our qualitative evaluation of user-perceived recommendation effectiveness, we recruit 20 participants. Participants are students and doctoral researchers from several universities in Berlin and the University of Konstanz. The average age of participants is 29 years. 65% of our participants say they spend more than an hour per month on Wikipedia, with the average being 4.6 hours spent on Wikipedia.

Our study contains both qualitative and quantitative data collection components. The quantitative component is in the form of a written questionnaire. This questionnaire asks participants about each recommendation set separately and elicited responses on a 5-point Likert scale. Some questions are tailored to gain insights into our research questions. The remainder of the questions adheres to the ResQue framework for user-centric evaluation (Pu et al., 2011). The qualitative data component is designed as a semi-structured interview. The interview contains open-ended questions that encourage participants to verbally compare and contrast the two recommendation sets. The participants are also asked to describe their perceived satisfaction. Resulting from this mixed methods study design, we could use the findings from the qualitative interviews to interpret and validate the results from the quantitative questionnaires. All interviews are audio-recorded with the permission of our participants.

3.1.4.2 Seed Articles

In this user study, each participant is shown four Wikipedia articles, one at a time. For each article, two recommendation sets, each containing five recommended articles, are displayed. One set is generated using CPA, while the other is generated using the MLT algorithm.

Each set of four Wikipedia articles is shown to a total of two participants to enable checking for the presence of inter-rater agreement. Participants are aware that recommendation sets are generated using different approaches, but they do not know the names of the approaches or the method behind the recommendations. We alternate the placement of the recommendation sets to avoid the recognition of one approach over the other and forming a potential bias based on placement. The seed Wikipedia articles are shown to participants via a tablet or a laptop. The participants are asked to read and scroll through the full article so that the exploration of the article's content is as natural as possible. We make the complete questionnaire and the collected data publicly available on GitHub.[8]

3.2 Offline Evaluation

In the following, we present the results of the offline evaluation. Before presenting the overall results, we report on the optimization of the CPI model. We conclude the offline evaluation with a manual evaluation that verifies the offline findings.

3.2.1 Optimizing the CPI Model

We employ a dynamic CPI model instead of the static CPI values proposed by Gipp and Beel (2009). Thus, we need to adjust the CPI for Wikipedia articles before comparing the approaches

[8] `https://github.com/malteos/wikipedia-article-recommendations`, last accessed: 18/01/2023

Table 3.2: Overview of seed articles selected for the study.

#	Article (Quality[6])	Words	#	Article (Quality[6])	Words
A	*Niche generic topics*		*C*	*Niche named entities*	
1	Babylonian mathematics (B)	3,825	21	Mainau (S)	567
2	Water pollution in India (S)	1,697	22	Lake Constance (C)	7,079
3	Transport in Greater Tokyo (C)	3,046	23	Spandau (C)	599
4	History of United States cricket (S)	3,610	24	Appenzell (C)	2,667
5	Firefox for Android (C)	4,821	25	Michael Müller (politician) (Stub)	602
6	Chocolate syrup (Stub)	391	26	Olympiastadion (Berlin) (C)	3,360
7	Freshwater snail (C)	1757	27	Theo Albrecht (S)	929
8	Touring car racing (S)	2550	28	ARD (broadcaster) (S)	2,397
9	Mudflat (C)	787	29	Kaufland (Stub)	680
10	Philosophy of healthcare (B)	3,804	30	Sylt Air (Stub)	110
B	*Popular generic topics*		*D*	*Popular named entities*	
11	Fire (C)	4,297	31	Albert Einstein (GA)	15,071
12	Basketball (C)	11,172	32	Hillary Clinton (FA)	28,645
13	Mandarin Chinese (C)	698	33	Brad Pitt (FA)	9,955
14	Cancer (B)	16,300	34	New York City (B)	30,167
15	Vietnam War (C)	32,847	35	India (FA)	16,861
16	Cat (GA)	17,009	36	Elon Musk (C)	11,529
17	Earthquake (C)	7,541	37	Google (C)	16,216
18	Submarine (C)	11,968	38	Star Wars (B)	16,046
19	Rock music (C)	19,833	39	AC/DC (FA)	10,442
20	Wind power (GA)	15,761	40	FIFA World Cup (FA)	7,699

with each other. We need to find a value for the α hyperparameter that achieves the best MAP score for the "See also" evaluation and the best CTR score for the clickstream evaluation.

To find the value for α that performs best for our dataset, we generate recommendations with CPA with α values ranging from -1 to 5 in 0.01 increments. Then, we evaluate the recommended top-k results with $k = 10$ of each batch by calculating the MAP and CTR scores (Figure 3.4). We find that CPA performs best in terms of MAP with α set to 0.81 and in terms of CTR with α set to 0.90 (see vertical lines in Figure 3.4). Thus, we use these optimized α values in the corresponding CPI models during the "See also" and clickstream evaluation.

The graph in Figure 3.4 also depicts the consistently lower performance of CoCit compared to CPA. CoCit is a special case of CPA with α set to zero (left line in the graph). Only for negative α values, CoCit performs better than CPA. Using negative α values would make CPA assign higher scores to more distant co-citations, effectively reversing the proximity notion of the CPA measure, and reducing its performance. As a result, the graph emphasizes the benefit of assigning higher scores to co-citations in closer proximity.

3.2.2 Overall Results

In the following, we present the offline evaluation results for the two silver standards discussed in Section 3.1.1.1. To be part of the "See also" evaluation, a Wikipedia article must contain a

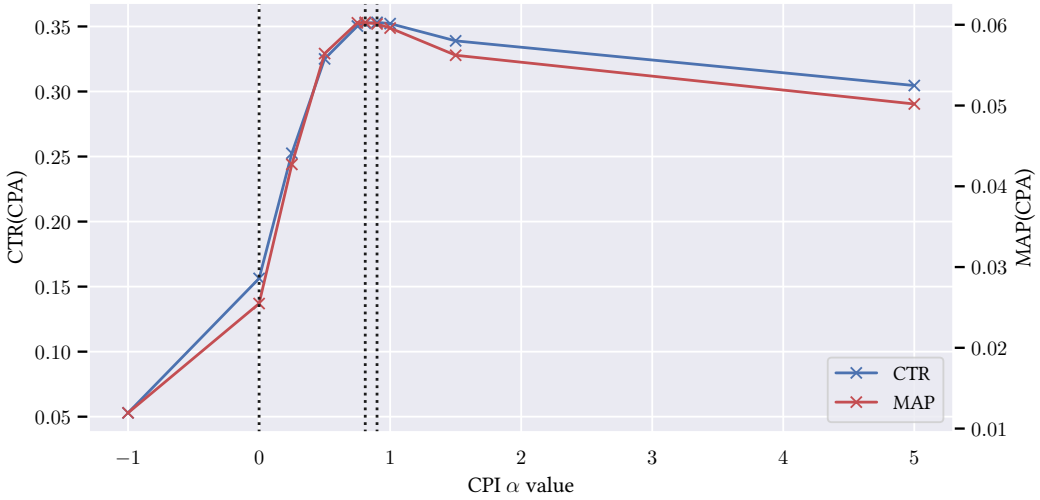


Figure 3.4: Optimization of the α parameter of CPA's CPI model. Performance measured in CTR and MAP w.r.t. α values. The maximum scores are achieved at $\alpha = 0.81$ (MAP) and $\alpha = 0.90$ (CTR), both represented with the two right vertical lines. The left vertical line represents the special case of $\alpha = 0$ that corresponds to the CoCit method.

"See also" section, which is true for 779,716 articles. To be part of the clickstream evaluation, clickstream data has to be available for the article in question, which is true for 2,572,063 articles.

To enable comparability of the evaluated similarity measures, the following sections report results for a "unified dataset", i.e., those articles for which all three evaluated measures recommended the same number of articles. For example, CoCit and CPA cannot generate recommendations for articles without in-links. Hence, we exclude articles without in-links from the unified dataset. This procedure reduces the dataset for the "See also" evaluation from 779,716 articles to 659,642 articles (-120,074 articles) and the dataset for the clickstream evaluation from 2,572,063 articles to 2,535,987 articles (-36,076 articles). To ensure that unifying the datasets does not skew the evaluation, we calculate all performance scores for CoCit, CPA, and MLT based on the sets of all related articles that the measures could identify. The maximum difference in any score was 1.3% (average number of clicks for CPA), and for most scores, less than 1% compared to the results of the unified dataset. For the interested reader, our GitHub repository includes the results for the unified dataset and the results for the set of all related articles.

3.2.2.1 "See also" Links

Table 3.3 shows that MLT performed better than CPA in terms of MAP and the average number of recommended relevant documents, while CoCit performed worst. The MAP score of CPA is less than half of MLT's score; CoCit's score is less than a quarter of MLT's score. The average number of relevant documents of MLT and CPA tripled from $k = 1$ to $k = 5$ and nearly quadrupled from $k = 1$ to $k = 10$. We expected significant performance differences between CoCit and CPA since the CPI optimization already showed that CoCit is an under-performing variation of CPA. On the other hand, we see an advantage of text-based MLT over the citation-based similarity measures when judging recommendation relevance using "See also" links.

Table 3.3: Results of the offline evaluation based "See also" links and clickstreams.

Metrics	CoCit	CPA	MLT
"See also" links			
Avg. relevant docs. ($k = 10$)	0.20	0.39	0.59
Avg. relevant docs. ($k = 5$)	0.12	0.27	0.43
Avg. relevant docs. ($k = 1$)	0.03	0.08	0.14
MAP ($k = 10$)	0.03	0.07	0.13
Clickstreams			
Avg. clicks ($k = 10$)	38.34	83.87	80.64
Avg. clicks ($k = 5$)	23.52	59.50	58.00
Avg. clicks ($k = 1$)	6.39	19.61	19.08
CTR ($k = 10$)	0.16	0.35	0.40

3.2.2.2 Clickstreams

Table 3.3 shows the CTR ranking of the clickstream evaluation. CPA accounts for more absolute clicks than MLT for any value of k, whereas MLT achieves the highest CTR. However, the ratio of the CTR scores of MLT and CPA (1.13) was significantly lower than that of the MAP scores of the two approaches (1.92). CoCit again performs worst concerning both scores.

The improved performance of CPA in this evaluation compared to the "See also" evaluation indicates that CPA performs better than MLT for popular articles, while MLT is more effective for niche articles. In the following, we present possible interpretations for this observation, which need further investigation.

Popular articles typically attract many visitors and thus impact the total click count more than niche articles. However, CTR values every article equally. Thus CTR does not reflect the comparably better performance of CPA for popular articles as strongly as the average number of clicks. Popular articles also tend to have more co-authors. Therefore, the collaboratively generated 'link set' contained within popular articles might be of higher relevance, thus generating higher numbers of clicks and CTRs. To be able to support this hypothesis, we would need to evaluate the performance with respect to indicators of article quality (Hu et al., 2007).

Additionally, popular articles likely receive more in-links affecting CPA performance. We further investigate this property in Section 3.2.3.2. Another cause for CPA performing better for popular articles might be that bots, i.e., computer-generated clicks, have a proportionally more significant impact on niche articles. Consequently, the quality of the silver standard for these articles might be lower than for articles of average popularity. As we explain in Section 3.1.1.1, we cannot quantify this effect since the data we used had been aggregated, thus preventing us from filtering bots on our own.

3.2.3 Impact of Article Properties

In this subsection, we provide details on evaluating CPA and MLT depending on article properties, such as the number of words and in-links. We omit CoCit in this evaluation since the "See also" and clickstream evaluations already showed inferior performance compared to CPA.

Figure 3.5 and 3.6 show the performance in terms of MAP and CTR for words and in-links. The graphs do not cover the full corpora: For the sake of visibility, we do not plot results for articles with more than 3,000 words (9.07% of the articles in the "See also" dataset, 5.90% of the articles in the clickstream dataset) or 400 in-links (2.66% of the articles in the "See also" dataset, 1.22% of the articles in the clickstream dataset).

3.2.3.1 Words

The performance plot for article length, see Figure 3.5, reveals some interesting results. First, we see that MLT consistently performs better than CPA when using MAP, but when using CTR, the performance ranking varies depending on the number of words. For articles with less than around 1,400 words, MLT is superior. Otherwise, CPA performs slightly better.

Second, MLT's and CPA's MAP and CTR graphs show similar tendencies, but with one exception: MLT's MAP scores for very short articles (30-50 words) are exceptionally high but drop sharply for slightly longer articles (60-150 words). For articles with more than approximately 150 words, MLT's MAP and CTR scores increase steadily and peak at article lengths of approximately 250 words. For articles longer than 250 words, MLT's MAP and CTR scores steadily decline. On the contrary, CPA's MAP and CTR scores increase until approximately 400 words. Beyond this word count, the CPA's MAP and CTR scores remain relatively stable.

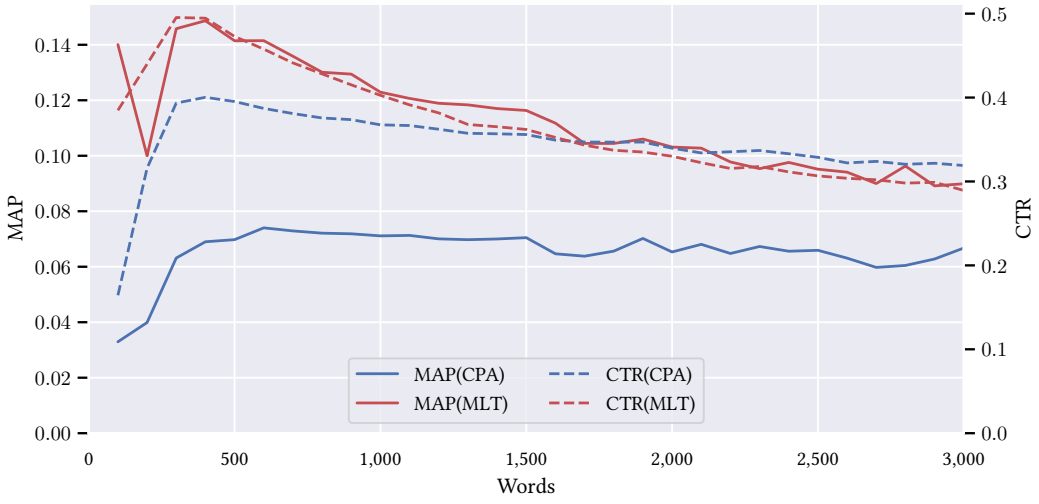


Figure 3.5: Offline evaluation results for CPA (blue) and MLT (red) for the number of words per article and measured as MAP (solid line) and as CTR (dashed line).

MLT's performance is more strongly affected by article length than CPA's. Short articles simply offer less data for both a text-based and graph-based similarity assessment. If an article contains few words, it is challenging to determine topic-defining keywords and find other articles with matching topics. Short articles also typically have fewer in-links, e.g., because they are stubs.

Therefore, MLT and CPA require an article length of approximately 250 or more words to perform well. MLT's MAP peaks for articles with around 50 words is an outlier phenomenon. Such very short articles typically contain only a single sentence on one topic, a list, a table, or specific vocabulary. Therefore, such articles often allow an accurate text-based similarity assessment.

While CPA reaches a relatively stable performance in MAP and CTR, MLT's MAP and CTR scores decline steadily for articles with 450 words or more. Long articles often cover several subtopics, which decrease the performance of text-based similarity approaches like MLT. The vocabulary of subtopics can vary, thus making it difficult to determine a set of words representing the breadth of topics present in the article. CPA's performance is hardly affected by article length, given a critical mass of in-links has been reached. This result is intuitive given that CPA's performance exclusively depends on in-links.

3.2.3.2 In-Links

Figure 3.6 shows the plot of MAP and CTR scores depending on the number of in-links. Both MLT and CPA performed best for approximately 20 in-links. For more in-links, the performance declines steadily as the number of in-links increases. This plot also shows a change in the CTR performance ranking of CPA and MLT. For less than 50 in-links, MLT performs better; for more than 50 in-links, CPA performs better. Compared to the CTR ranking, the ranking according to MAP does not change.

In-links as a data source are essential for graph-based similarity measures but do not directly affect text-based similarity measures. Seeing MLT perform better than CPA in terms of CTR for articles with less than 20 in-links is therefore intuitive. It is also intuitive that CPA's CTR scores increase as the number of in-links increases in the range of 0 to 20 in-links.

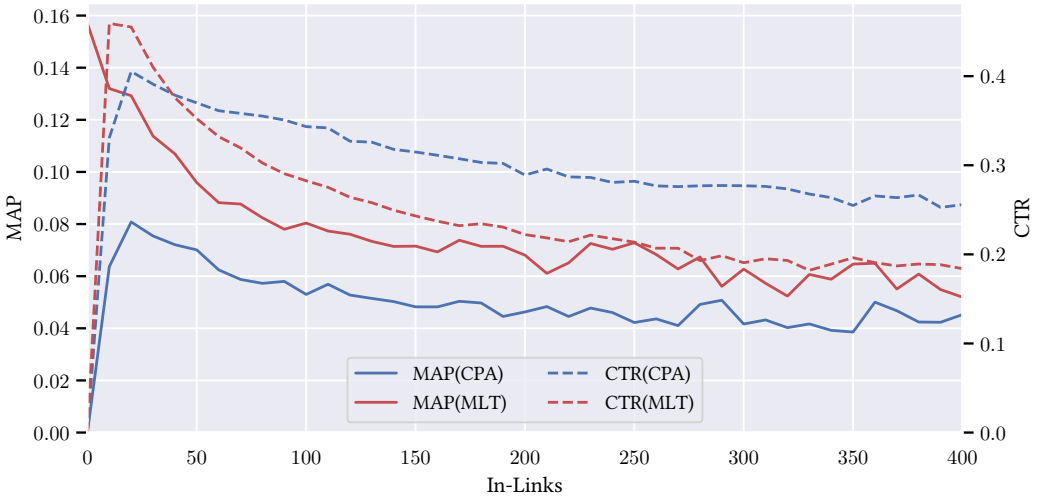


Figure 3.6: Offline evaluation results for CPA (blue) and MLT (red) for the number of in-links per article and measured as MAP (solid line) and as CTR (dashed line).

The reason that CPA's CTR scores peak at 20 in-links and decline thereafter and MLT's CTR scores decline steadily as the number of in-links increases may not be as intuitive. We attribute this behavior to the nature of articles that receive many in-links. Such articles typically cover broad

topics, e.g., countries, as Bellomi and Bonato (2005) also reported. We explain in Section 3.2.3.1 that text-based similarity measures like MLT perform worse for such articles than for articles with narrowly similar topics. Figure 3.6 also demonstrates that graph-based measures like CPA perform worse for broad-topic articles because such articles receive in-links from many topically diverse articles. This diversity of in-links reduces the likelihood that the article in question is frequently co-cited in closer proximity with other articles, reducing CPA's performance.

3.2.4 Manual Sample Analysis

To test the validity of "See also" links and clickstreams as reference datasets, we manually evaluate a small and random subset of Wikipedia articles. From these articles, we present and discuss three exemplary articles: Technical University of Berlin (Table 3.4), Newspaper (Table 3.6), and Elvis Presley (Table 3.5). We chose the articles for their diversity and comprehensibility. Tables 3.4-3.6 list the recommendations of CoCit, CPA, and MLT with the corresponding rank and click counts. Recommendations that are part of the "See also" links are bold.

3.2.4.1 Technical University of Berlin

The article about the Technical University of Berlin (TUB) includes information about the university's history, campus, organization, and a list of notable alumni and professors. Both graph-based measures, CoCit and CPA, recommend two articles, which are included in the "See also" links and received clicks (Humboldt University of Berlin and Free University of Berlin, bold in Table 3.4). The MLT results have a clear focus on the term "University" as the central topic since all recommended articles are about universities but from other cities and countries.

Table 3.4: Recommendations for "Technical University of Berlin" with total of 596 clicks and with the "See also" links: Hertie School of Governance, Berlin University of the Arts, Free University of Berlin, Humboldt University of Berlin, Berlin School of Economics and Law, Beuth University of Applied Sciences Berlin

Rank	CoCit (clicks)	CPA (clicks)	MLT (clicks)
1	Germany (0)	Germany (0)	Technical Uni. of Sofia (0)
2	Berlin (20)	Berlin (20)	University of Economics Varna (0)
3	**Humboldt Uni. of Berlin** (42)	**Humboldt Uni. of Berlin** (42)	Vilnius College of Tech. and Design (0)
4	Ludwig Maximilian Uni. of Munich (0)	RWTH Aachen Uni. (0)	Braunschweig Uni. of Technology (0)
5	World War II (0)	Technische Uni. München (0)	Technical Uni. of Gabrovo (0)
6	United States (0)	Charlottenburg (0)	Chemnitz Uni. of Technology (0)
7	RWTH Aachen University (0)	Mathematics (0)	Technische Uni. Ilmenau (0)
8	**Free Uni. of Berlin** (0)	**Free Uni. of Berlin** (0)	Technical Uni. of Dortmund (0)
9	Heidelberg Uni. (0)	Habilitation (0)	Dresden Uni. of Technology (0)
10	Mathematics (0)	Ludwig Maximilian Uni. of Munich (0)	Technological Uni. Hpa-An (0)

In this case, it can be said that the best results are produced by the CPA algorithm, followed by CoCit and MLT. While the CPA results can all be considered relevant, the MLT approach produces a list of irrelevant institutions. For example, the University of Economics Varna in Bulgaria or the Technological University Hpa-An in Myanmar. Opposed to that, the universities considered relevant by the CPA approach are all well-known Universities in the region with a strong technical focus, similar to the Technical University of Berlin.

In this example, the poor performance of MLT can be explained by the weakness of text-based approaches where a strong emphasis lies on overlapping terms in the documents. Text describing a university is usually similar, given those generic characteristics such as the number of students, etc. are described, automatically leading to a "high" text-based similarity. Possibly, Wikipedia authors reused text when writing the article about the university in Burma. Citation-based approaches are not affected by this text reuse issue.

Table 3.5: Recommendations for "Elvis Presley" with total 92,379 clicks and with the "See also" links: Honorific nicknames in popular music, Elvis Presley Enterprises, List of best-selling music artists, Personal relationships of Elvis Presley, List of artists by number of UK Albums Chart number ones, List of artists by the total number of UK number one singles

Rank	CoCit (clicks)	CPA (clicks)	MLT (clicks)
1	AllMusic (0)	The Beatles (247)	Sun Studio (0)
2	The Beatles (247)	Frank Sinatra (139)	From Elvis in Memphis (516)
3	Billboard magazine (0)	Johnny Cash (140)	List of songs recorded by Elvis Presley on the Sun label (240)
4	United States (52)	Jerry Lee Lewis (73)	Peter Guralnick (0)
5	Frank Sinatra (139)	RCA Records (175)	Colonel Tom Parker (1175)
6	The Rolling Stones (0)	Rock and roll (306)	The Blue Moon Boys (100)
7	Billboard Hot 100 (12)	Heartbreak Hotel (720)	Elvis Presley's Army career (619)
8	Johnny Cash (140)	Jailhouse Rock song (260)	Jailhouse Rock film (1132)
9	Cliff Richard (0)	Roy Orbison (96)	I Want You, I Need You, I Love You (83)
10	Bob Dylan (77)	United States (52)	Elvis Presley albums discography (6084)

3.2.4.2 Elvis Presley

The biographical Wikipedia article about the American singer and actor Elvis Presley is relatively long. The article contains 24,298 words, received 5,834 in-links, and provided 92,379 out-clicks.

None of the articles recommended by any approach were part of the "See also" links, but most recommendations are related to the topic. The topics CoCit and CPA recommended are broader than MLT's results. Furthermore, CoCit's recommendations for the articles "AllMusic", an online music database, and "Billboard magazine" are notable: Even though both articles are music-related, they lack a direct connection to Elvis Presley. These recommendations were caused by links not belonging to the actual article text, e.g., info boxes or the article footer.

Table 3.6: Recommendations for "Newspaper" with total of 4,516 clicks and with the "See also" links: List of newspaper comic strips, Lists of newspapers

Rank	CoCit (clicks)	CPA (clicks)	MLT (clicks)
1	United States (0)	Broadsheet (59)	The Daily Courier Arizona (0)
2	Broadsheet (59)	Magazine (119)	Online newspaper (142)
3	English language (0)	Tabloid newspaper format (35)	History of British newspapers (168)
4	Tabloid newspaper format (35)	United States (0)	List of newspapers in the United States by circulation (0)
5	Race and ethnicity in the United States Census (0)	Publishing (0)	Newspaper circulation (23)
6	The New York Times (118)	English language (0)	Midland Daily News (0)
7	New York City (0)	Journalist (32)	The Huntsville Times (0)
8	World War II (0)	Book (11)	Decline of newspapers (0)
9	Magazine (119)	Comic strip (37)	The Leaf-Chronicle (0)
10	United Kingdom (0)	Radio (0)	The Ann Arbor News (0)

3.2.4.3 Newspaper

The "Newspaper" article contains general information on newspapers as periodical publications, historical development, categories, formats, and other newspaper-related topics. The article consists of 6,313 words and is linked to 7,611 other articles. The "See also" section includes two links to newspaper-related lists: "List of newspaper comic strips" and "Lists of newspapers".

MLT, CPA, and CoCit all fail to recommend any of the "See also" links, which is not surprising since the two "See also" links point to another list. Despite all articles recommended by MLT being newspaper-related, they were also overly narrow and irrelevant for the broad and internationally-oriented "Newspaper" article. MLT recommends articles on actual newspaper publications, e.g., "The Daily Courier Arizona", or "Midland Daily News"; However, these publications are so provincial that they will be irrelevant to most readers. In contrast to MLT, CPA recommends a broader range of related topics, for instance, newspaper formats ("Tabloid", "Magazine", "Broadsheet") or other media types ("Book", "Comic strip", "Radio"). Two of the CPA recommendations ("United States" and "English language") are not topically relevant. CoCit recommended many irrelevant articles from the geopolitical category ("United States", "New York City", etc.)

3.2.4.4 Summary of Manual Evaluation

The results presented for these three examples were typical of other articles examined. MLT tended to recommend topically more narrow articles compared to the graph-based approaches. CPA usually produced more relevant recommendations than CoCit. We observed that the recommendations were of a different nature for each approach. While CPA's recommendations were consistently plausible, MLT tended to recommend obscure articles. For example, MLT recommended a University in Myanmar (Technological University Hpa-An) for the article 'Tech-

nical University Berlin' or an internationally virtually unknown newspaper ('The Daily Courier Arizona') at rank 1.

The result of the manual evaluation showed that CPA recommends topically broader articles but with consistent relevance compared to the often niche results of MLT. However, because this evaluation approach is highly subjective and dependent on a user's specific information need, we invite the reader to examine the examples as well as additional results available in the repository to make a judgment.

3.2.5 Discussion of Offline Evaluation

We derive the following findings from the offline evaluation. In the "See also" evaluation, the text-based MLT measure recommended more relevant articles and achieved higher MAP than both graph-based measures. CPA followed at second rank and clearly outperformed the third-ranked CoCit in this evaluation. Links outside of the article text, e.g., in information boxes or article footers, were a source of irrelevant CoCit and CPA results since such links are commonly less related to the article's topic.

For example, CoCit recommended the "AllMusic" article at the top rank in the article about Elvis Presley (Table 3.5). Downranking or ignoring these links in future studies should improve the performance of the graph-based similarity measures. Such a procedure would correspond to the stop word removal in MLT. For the CPA approach, adjusting the CPI weighting scheme could reduce the effect of such Wikipedia-specific unrelated results. For instance, the quantification of citation proximity should be adjusted for article length or the number of in-links an article receives. Such normalization can downrank links to general articles that are frequently co-cited but have no topical relevance, e.g., geopolitical articles such as "United States".

Recommendations by CPA consistently achieved the highest number of clicks in the clickstream evaluation. MLT followed at the second rank and CoCit at the third rank in this regard. Yet, MLT achieved slightly higher CTR scores than CPA in this evaluation, with CoCit again following at rank three. These results indicate that traditional text-based methods are a well-performing "general purpose" approach for recommending semantically similar Wikipedia articles regardless of specific article properties. CPA is better suited to recommending popular articles. Due to Wikipedia's collaborative approach to article curation, popular articles are typically longer and of higher quality.

Our manual evaluation of samples also indicated that CPA and MLT have different strengths that are not adequately reflected by the "See also" silver standard. The graph-based approaches, especially CPA, tended to recommend articles from a broader range of related topics compared to MLT. For instance, for the seed article "Newspaper" MLT mostly recommended actual newspapers, e.g., "The Daily Courier Arizona" (Table 3.6). On the other hand, CPA recommended more generally related topics, e.g., newspaper formats such as "Tabloid" or "Broadsheet". In our perception, CPA and MLT performed similarly well in identifying semantically similar articles, yet the type of similarity differs.

Two advantages of the graph-based measures over the text-based measure are their significantly lower run time requirement (Table 3.1) and their language independence. Citation or link analysis can be performed for texts in any language and can also be employed for retrieving texts across languages (when links are used across languages). Text-based measures like MLT are language-dependent.

Summarizing our findings, we conclude that text-based and graph-based approaches address different aspects of the content of Wikipedia articles. The advantage of one source of information over the other depends on the information need of the user. If a user is interested in articles that address a specific topic in a single language and from a relatively narrow perspective, the text-based recommendations from MLT likely suit the user's information need better than graph-based recommendations. If the user desires a broader overview of a topic and wants to see articles in different languages, or if the user values factors like article popularity and quality, then graph-based recommendations fulfill these requirements better than text-based recommendations.

Ultimately, a combined approach that includes graph-based, text-based, and potentially other document similarity measures is likely to achieve the best recommendation quality.

3.3 User Study Evaluation

In this section, we summarize and discuss the results of our user study on Wikipedia articles. At first, we present the primary findings, which provide answers to our three research questions, before illustrating them with participants' quotes. Subsequently, we discuss secondary findings that arose from coding the participants' responses and go beyond the research questions we initially set out to answer.

3.3.1 Primary Results

Our user study finds several differences in the reader's perception of the graph-based approach compared to the text-based approach. A notable difference could be identified especially in the perceived degree of 'similarity' of the recommendations. Participants are significantly more likely to agree with the statement 'the recommendations are more similar to each other' (see 1.6 in Figure 3.7) for the MLT approach. 73% of responses 'agree' or 'strongly agree' (58 out of 80 responses) with this statement, compared to only 36% of the responses for the CPA approach (29 out of 80). Keep in mind that each of the 40 seed articles is examined by two participants, resulting in 80 responses. A question about whether the articles being recommended 'matched with the content' of the source article (see 1.1) is answered with a similar preference, with a significantly higher portion of the responses indicating 'strongly agree' or 'agree' for the MLT approach (73%) and only 38% of responses choosing the same response for the CPA approach.

Overall, users perceive recommendations of CPA as more familiar (see 1.3). They feel less familiar (1.4) with the recommendations made by MLT. We find that this difference is observed by nearly all participants and can be attributed to how MLT considers textual similarity. MLT generally focuses on overlapping terms, while CPA utilizes the co-occurrence of links. The offline evaluation results already suggested that this leads to diverging recommendations (Section 3.2).

3.3.1.1 Perceived Difference between CPA and MLT

The participants observed that the methodological difference between the approaches affects their recommendations. In the questionnaire, participants express 48 times that the articles recommended by CPA are more diverse, i.e., less similar, compared to the seed article (Figure 3.8). MLT's recommendations are found to be diverse only 13 times. Regarding the similarity of recommendations, the outcome is the opposite.

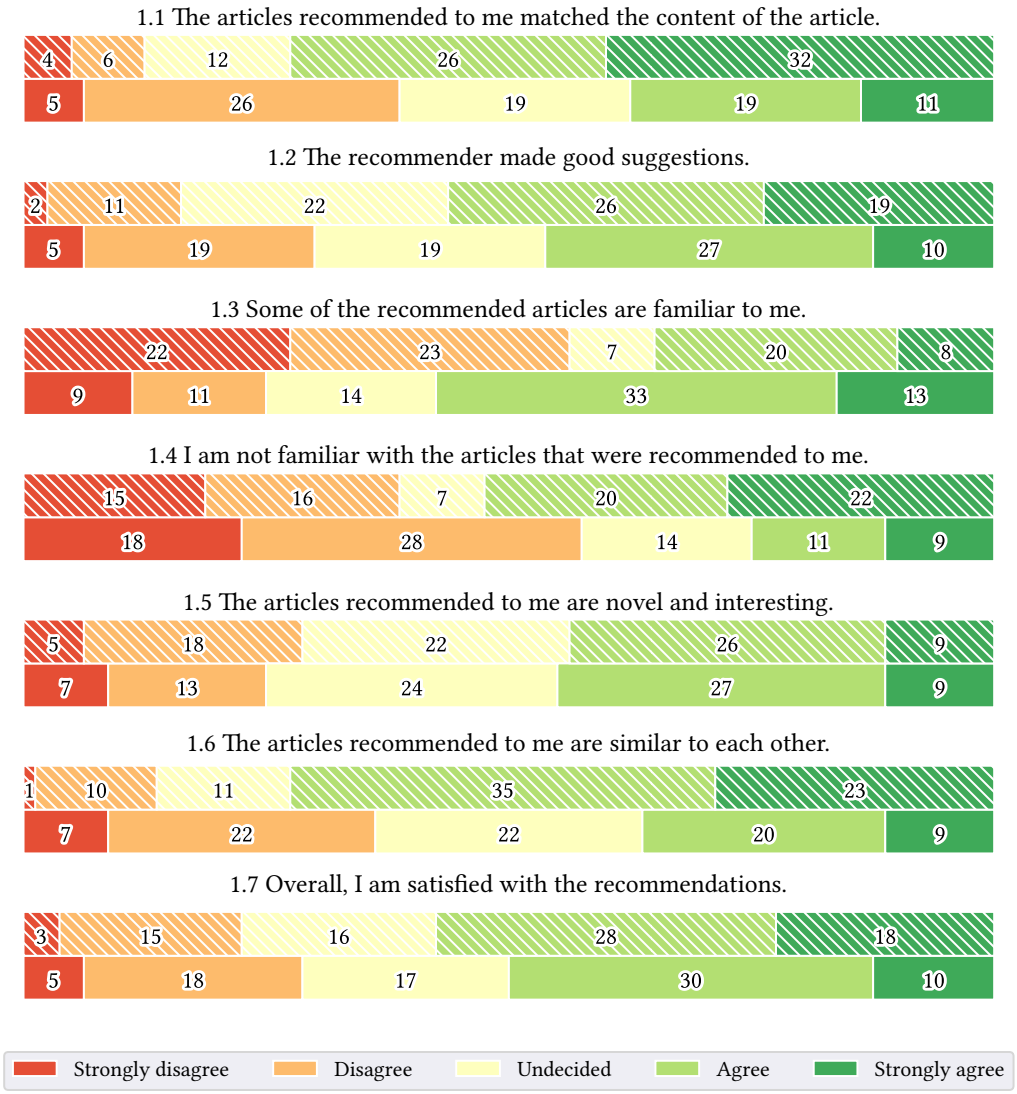


Figure 3.7: Responses for MLT (dashed) and CPA (solid) on a 5-point Likert scale.

The participants' answers also indicate the difference between MLT's and CPA's recommendations. Participant P20 explains that *"approach A [CPA] is more an overview of things and approach B [MLT] is focusing on concrete data or issues and regional areas"*. CPA providing an "overview of things" is not favorable for all participants as they describe different information needs. For example, participant P20 prefers MLT's recommendation since *"it is better to focus on the details"*. Some participants attributed the recommendations' similarity (or diversity) to terms co-occurring in the seed title and recommended articles. For instance, participant P15 finds MLT's recommendations for *Star Wars* to be more similar because *"Star Wars is always in the title [of MLT's recommendations]"*. Participant P17 also assumes a direct connection between the seed and MLT's recommendations *"I'd guess recommendations of A [MLT] are already contained as a link in the source article"*.

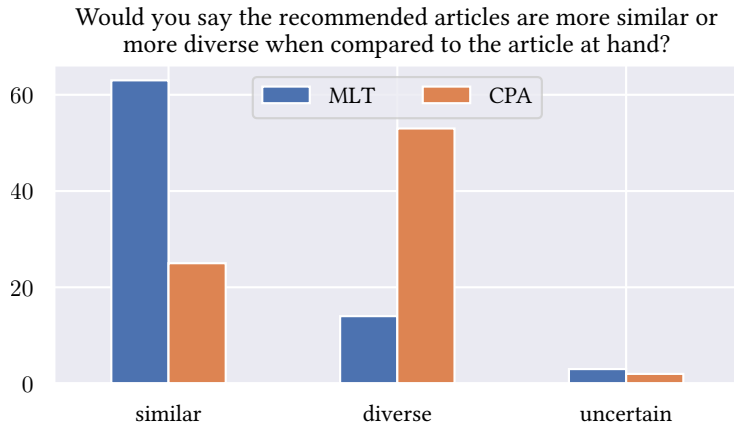


Figure 3.8: Diversity or similarity.

Nonetheless, participants struggle to put the observed difference between MLT and CPA in words, although they notice categorical differences in the recommendation sets. Participant P19 said *"I can see a difference, but I don't know what the difference is"*. Similarly, participant P20 finds that *"they [MLT and CPA] are both diverse to the same extent but within a different scope"*.

Concerning the overall relevancy of the recommendations, MLT outperforms CPA. In total, the participants agree or strongly agree 45 times that MLT made 'good suggestions' (see 1.2), whereas only 37 times the same is stated for CPA. Similarly, MLT's overall satisfaction is slightly higher (46 times agree or strongly agree) than CPA (40 times; see 1.7).

3.3.1.2 Information Need

The participants are also aware of relevancy depending on their individual information needs. When asked about the 'most relevant recommendation', the participants' answers contain the words 'depends' or 'depending' ten times. Participant P15 states that *"if I want a broader research I'd take B [CPA] but if I'd decide for more punctual research I would take A [MLT] because it is more likely to be around the submarine and because in B [CPA] I also get background information"*. Similarly, participant P13 would click on a recommendation as follows: *"if you're looking for a specific class/type of snails, then this [MLT] could be one, but if you're just looking to get an overview of aquatic animals, then probably you would click on the other approach [CPA]"*. In summary, the participants agree that CPA provides 'background information' that is useful to 'get an overview of a topic', while MLT's recommendations are perceived as 'more specific' and having a 'direct connection' to the seed article.

The most commonly expressed information needs for articles on science and technology are understanding how technology works or looking up a definition. For individual articles, participants express the need to find dates relating to an individual and understand their contributions to society. For 'niche' topics, users are slightly more likely to state the desire to discover sub-categories on a topic, which implies wishing to move from a broader overview to a more fine-grained and in-depth examination of the topic.

The subjectiveness in the perception of recommendations is reflected in the inter-rater agreement. On average, the participants who review the same articles have a Cohen's kappa of $\kappa = 0.14$, which corresponds to a slight agreement. The inter-rater agreement increases to a "fair agreement"

(Landis and Koch, 1977) when we move from a 5-point to a 3-point Likert scale, i.e., possible answers are 'agree', 'undecided', or 'disagree'. A low agreement indicates that the perception of recommendation highly depends on the individual's prior knowledge and information needs.

3.3.1.3 Article Characteristics

In the methodology section, we define article 'types' according to article popularity, length, and breadth into the four categories 'popular generic', 'niche generic', 'popular named entities', and 'niche named entities'. These categories have no observable impact on user's preference for one recommendation approach over the other.

However, we find that the user-expressed information need, for example, the desire to identify related articles that are either more broadly related or more specialized, has a measurable impact on the user's preference for the recommendation method. For *popular generic articles* on science and technology, e.g., the article on wind power, the most frequently expressed information needs are understanding how technology works or looking up definitions.

For articles in the categories 'popular generic' and 'niche generic', we could observe that the information needs expressed by our readers are more *broad*. For example, they want to find definitions for the topic at hand, more general information to understand a topic in its broader context, or examples of sub-categories on a topic. There is no observable difference between the specified categories of information need for 'popular' vs. 'niche' generic articles.

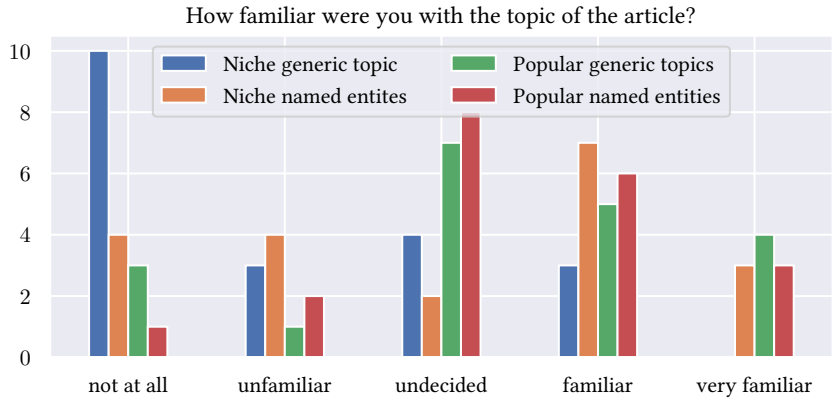


Figure 3.9: Familiarity with article topic.

Given our initial classification of the selected 40 Wikipedia articles, the empirical questionnaire data shows that 'niche' entities are, on average, more familiar to the participants than we initially expected (Section 3.1.4.1). This is especially the case for named entities from niche topics, many of which are rated as familiar to the participants. This may be because our participants are from Germany and are thus familiar with many of these articles, despite the articles reporting on regional German topics, e.g., Spandau, Mainau. On the other hand, users rated niche generic topics as less familiar, which is in line with what we expected.

Furthermore, both popular generic topics and popular named entities are less often classified as 'unfamiliar' by the participants than they are classified as 'neutral or familiar'. Lastly, one notable finding is that Wikipedia listings, e.g., *List of rock genres* or *List of supermarket chains in Germany*, are found to be the most relevant recommendations in some cases. Our CPA

implementation intentionally excludes Wikipedia listings from its recommendation sets. Thus, the implementation needs to be revised accordingly.

3.3.2 Secondary Results

The analysis of the participants' interviews led us to the following secondary findings:

Figure 3.10: User's satisfaction depends on interest and familiarity, i.e., for all articles or only the articles which are very interesting or familiar to the user.

3.3.2.1 Effect of User's Interest on Recommendations

Figure 3.10 shows that MLT outperforms CPA in recommendation satisfaction if participants 'strongly agree' with the article topic being (a) interesting or (b) being at least 'familiar' to them. This is likely the case because users are more versed in judging the relevance of the text-based recommendations of the MLT approach if they already have more in-depth knowledge of a topic. For example, one participant observes CPA's recommendations of *Renewable energy* for *Wind power* as the seed article: *"Renewable energy is least relevant because everybody knows something about it [Renewable energy]"*. Sinha and Swearingen (2002) have already shown the previous familiarity with an item as a confounding factor on a user 'liking' a recommendation. Interestingly, this trend is no longer observable in cases when participants only 'agree' but without strong conviction that the articles are interesting or familiar to them. In these cases, the MLT and CPA approaches are seen as more equal, with the CPA approach taking a slight lead.

3.3.2.2 User-based Preferences

Our findings confirm the subjectiveness of recommendation performance since we observe user-based preferences. For instance, participant P2 only agrees or strongly agrees with MC 1.2 'The recommender made good suggestions' and 1.7 'Overall, I am satisfied with the recommendations' for CPA but never gives the same answers for MLT. However, participant P3 shows the opposite preference, i.e., only MLT makes good suggestions according to P3. The remaining participants have more balanced preferences. In terms of MC 1.2 and 1.7, nine participants have a tendency to prefer MLT, while six participants preferred CPA, and five participants do not show any particular preference for one of the two recommendation approaches.

3.3.2.3 Perception of Novelty & Serendipity of Recommendations

To gain insights into the user-perceived novelty, we ask participants to rate the following statements: 'I am not familiar with the articles that were recommended to me' and 'The articles recommended to me are novel and interesting'. While novelty determines how unknown recommended items are to a user, serendipity is a measure of the extent to which recommendations positively surprise their users (Ge et al., 2010; Kaminskas and Bridge, 2017). For instance, participant P19 answers *"approach A [MLT] shows me some topics connected with the article I read but with more special interest - they are about "Healthcare", and B [CPA] actually changes the whole topic. B [CPA] offers totally different topics."* CPA's recommendations are generally found to be more serendipitous. For the question regarding an 'unexpected recommendation among the recommendation set' CPA receives a positive answer 41 times, compared to only 23 times for MLT. The perceived novelty also makes participants click on recommendations. Among others, participant P1 explains that *"there are more [MLT] articles that I would personally click on because they are new to me."* Similarly, participant P16 states that they would *"click first on Star Wars canon, because I don't know what it is"*.

3.3.2.4 Trust & (Missing) Explanations

Although the questionnaire is not designed to investigate the participants' trust in the recommendations, many answers relate to this topic. When users were asked about the relevance of recommendations, some participants expressed that there "must be a connection" between the article at hand and a recommendation and that they just "do not know what is has to do with it". Others are even interested in topically irrelevant recommendations. For example, they express *"it interests me why this is important to the article I am reading"*. Similarly, a participant says that they might click on a recommendation *"because I do not know what it has to do with [the seed article]"*. Such answers are more frequent for CPA recommendations since they are more broadly related than MLT's more narrow topical similarity. In some cases, there is no semantic similarity. Yet, participants often do not recognize a recommendation as irrelevant. Instead, they say it is their fault for not knowing how the recommendation is relevant to the seed article. This behavior indicates a high level of trust from the participants in the recommender system.

3.3.3 Discussion of User Study

The user study demonstrates that MLT and CPA differ in their ability to satisfy specific user information needs. Furthermore, our study's participants are capable of perceiving a systematic difference between the two approaches.

CPA is found to provide an 'overview of things' with recommendations more likely to be unfamiliar to the participants and less likely to match the content of the seed article. In contrast, MLT is found to 'focus on the details'. Participants also feel that MLT's recommendations matched the content of the seed article more often. At the same time, participants perceive CPA's recommendations as more diverse, while MLT's recommendations are more similar to each other. So CPA and MLT, being conceptually different approaches and relying on different data sources, lead to unique differences in how their recommendations are perceived. In terms of overall satisfaction with recommendations, most participants expressed a preference for MLT over CPA. MLT is based on TF-IDF and, therefore, its recommendations are centered around specific terms (e.g., P15: *"Star Wars is always in the title"*). In contrast, CPA relies on the co-occurrence of links. According to CPA, two articles are considered related when they are mentioned in the

same context. Our results show that this leads to more distantly related recommendations, which do not necessarily share the same terminology. Given that the participants experience the two recommendation approaches differently, a hybrid combination of text and graph information is preferable depending on the context.

Moreover, the differently perceived recommendations show the shortcoming of the notion of similarity. Both approaches, CPA and MLT, are developed to retrieve semantically similar documents, which they indeed do (Gipp and Beel, 2009; Jones, 1973). However, their recommendations address different aspects of the article content. Both approaches convey a different notion of similarity. A recommended article that provides an 'overview' can be considered similar to the seed article. Equivalently, a 'detailed' recommendation can also be similar to the seed but in a different context. Our qualitative interview data shows how users perceive these two similarity measures differently. These findings align with the work from Bär et al. (2011), which finds that text similarity inherits different dimensions.

We also find that either CPA's or MLT's recommendations are liked or disliked depending on the individual participant's preferences. Some participants even express a consistent preference for one method over the other. However, a strict preference was the exception. We could also not identify any direct relation between the user or article characteristics and the preference for one method. At this point, more user data as in a user-based recommender system would be needed to tailor the recommendations to the user's profile. Purely content-based approaches such as CPA and MLT lack this ability (Beel et al., 2016b; Jannach et al., 2010; Lenhart and Herzog, 2016). The only option would be to allow users to select their preferred recommendation approach through the user interface depending on their information needs.

The participants' answers also reveal trust in the quality of the recommendations, although the trust was not always justified. Participants would assume a connection between the seed article and the recommended article just because it is recommended by the system. Instead of holding the recommender system accountable for non-relevant recommendations, participants find themselves responsible for not understanding a recommendation's relevance. To not disappoint this trust, recommender systems should provide explanations that help users understand why a particular item is recommended. Also, explanations would help users to understand the connections between seed and recommendations. Explainable recommendations are a subject of active research (Kunkel et al., 2019; Zhang and Chen, 2020). However, most research focuses on user-based approaches. Explainable content-based approaches are an unexplored research area, although methods such as CPA or MLT would also benefit from explanations.

Despite the insights of our user study to elicit users' perceived differences in recommendation approach performance, the nature of our evaluation has several shortcomings. With 20 participants, the study is limited in size. Consequently, our quantitative data points suggest a difference that is not statistically significant. When consulting only our offline evaluation and quantitative questionnaire data, one could assume that MLT and CPA are comparable in some aspects since their average scores are similarly high. The discrepancies between CPA and MLT only become evident when analyzing the written and oral explanations of live users. This highlights that recommender system research should not purely rely on offline evaluations (Beel et al., 2016b).

3.4 Summary of the Chapter

In this chapter, we focussed on Research Task **I** and compared existing document similarity measures. In particular, this chapter investigated classical similarity measures, the text-based MLT (a TF-IDF implementation; Section 2.3.2) and the graph-based CPA (and CoCit; Section 2.4.4). The similarity measures are evaluated for the task of recommending Wikipedia articles. We conducted a large-scale offline evaluation (Section 3.2) and a user study (Section 3.3) to compare MLT and CPA in a quantitative and qualitative manner.

The offline evaluation found that the graph-based and text-based methods have complementary strengths. While the text-based MLT method performed well in identifying closely related articles, the graph-based CPA, which consistently outperformed CoCit, was better suited for identifying a broader spectrum of related articles and popular articles that typically exhibit a higher quality. In terms of evaluation metrics, MLT achieved better MAP and CTR scores, whereas CPA's recommendations led to the highest number of absolute clicks. Consequently, both evaluation revealed no significant difference in recommendation accuracy of MLT and CPA.

Our user study with 20 participants confirmed these findings. The users were generally more satisfied with the recommendations generated by text-based MLT, whereas CPA's recommendations were perceived as more novel and diverse. The methodological difference between CPA and MLT, i.e., based on either text or links, was reflected in their recommendations and noticed by the participants. Depending on information needs or user-based preferences, one recommender approach was preferred over the other. Thus, we suggest combining both approaches in a hybrid system since both address different information needs. However, the challenge for such a hybrid approach would be making different notions of MLT's and CPA's semantic similarity accessible to the users of a recommender system. Moreover, their notions of similarity lack a proper definition; hence, their similarity cannot be explicitly stated. Does MLT yield a *topic-specific* similarity? Or are CPA's recommendations *concept-similar*? In the words of Goodman (1972), to what aspects does the similarity of MLT or CPA relate? We will investigate these questions in the subsequent chapters of this thesis.

Aside from comparing MLT and CPA, this chapter made additional contributions. We introduced the first implementation of CPA for a hyperlink use case. We adopted CPA's CPI model from the academic literature to analyze links. To conduct the large-scale offline evaluation, we proposed two novel silver standards based on Wikipedia's "See also" sections and a comprehensive clickstream dataset as estimators of the relevance of Wikipedia articles.

Chapter 4

Legal Literature Recommendations

This chapter continues the work on Research Task **I** from the previous chapter. But in contrast to Chapter 3, which focuses on rather traditional methods, this chapter investigates a large variety of state-of-the-art document representation techniques for the task of finding semantically similar court decisions. The content of this chapter is published in Ostendorff et al. (2021a).

"Evaluating Document Representations for Content-Based Legal Literature Recommendations" by **Malte Ostendorff**, Elliott Ash, Terry Ruas, Bela Gipp, Julian Moreno-Schneider, and Georg Rehm. In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law (ICAIL)*, 2021.

Legal professionals, e.g., lawyers and judges, frequently invest considerable time in finding relevant literature (Lastres, 2013). More so than most other domains, in law, there are high stakes for finding the most relevant information (documents) as that can drastically affect the outcome of a dispute. A case can be won or lost depending on whether or not a supporting decision can be found. Recommender systems assist in the search for relevant information. However, research and development of recommender systems for legal corpora pose several challenges. Recommender system research is known to be domain-specific, i.e., minor changes may lead to unpredictable variations in the recommendation effectiveness (Beel et al., 2016a). Likewise, legal English is a peculiarly obscure and convoluted variety of English with the widespread use of common words with uncommon meanings (Mellinkoff, 1963). Recent language models may not be equipped to handle legal English since they are pretrained on generic corpora like Wikipedia or cannot process lengthy legal documents due to their limited input length. This raises the question of whether the recent advances in recommender systems and NLP research and underlying techniques also apply to the legal domain.

In this chapter, we empirically evaluate 25 document representation methods, analyze the results for the aforementioned possible issues, and answer the following research questions:

Research questions

RQ1: What document representation method performs best on the task of finding semantically similar court decisions?

RQ2: Can individual methods be combined in a hybrid manner to improve the overall performance?

RQ3: How do document properties such as length or citations affect the performance?

To answer these research questions, we evaluate state-of-the-art document representations in a literature recommender use case for each method. The methods are distinguished into three categories: (1) word vector-based, (2) Transformer-based, and (3) graph-based methods. Furthermore, we evaluate additional hybrid variations of individual methods. Our primary evaluation metric comes from two silver standards on United States case law that we extract from Open

Case Book and Wikisource. The relevance annotations from the silver standards are provided for 2,964 documents.

In this chapter, we present the following main contributions:

1. We propose and make available two silver standards as benchmarks for legal recommender system research.
2. We conduct a quantitative evaluation of 25 methods, the majority of which have not been previously investigated for legal literature, and validate our results qualitatively.
3. We demonstrate that a simple hybrid combination of text-based and graph-based methods can further improve the recommendation performance.

The remainder of this chapter is structured as follows: First, we introduce the general experimental methodology, i.e., the datasets and the evaluated methods. Subsequently, we present the evaluation based the two silver standards, starting with the overall results in Section 4.2.1, impact of document properties in Section 4.2.2, coverage and similarity of recommendations in Section 4.2.3, and manual analysis in Section 4.2.4. In Section 4.3, we discuss the results of all evaluations. Finally, we summarize the main findings of this chapter.

4.1 Methodology

This section describes our quantitative evaluation of 25 methods for legal document recommendations. The methods are evaluated with a specific recommendation scenario in mind.

Recommendation scenario. The recommendations are consumed in the following context: The user, a legal professional, needs to research a particular decision, e.g., to prepare a litigation strategy. Based on the decision at hand, the system recommends other decisions to its users such that the research task is easy to accomplish. The recommendation is relevant when it covers the same topic or provides essential background information, e.g., it overruled the seed decision (Opijnen and Santos, 2017).

4.1.1 Datasets

Most of the related works (Section 2.1.1) evaluate recommendation relevance by asking domain experts to provide subjective annotations (Boer and Winkels, 2016; Kumar et al., 2011; Mandal et al., 2017; Winkels et al., 2014). Especially in the legal domain, these expert annotations are costly to collect and, therefore, their quantity is limited. For the same reason, expert annotations are rarely published. Consequently, the research is difficult to reproduce (Beel et al., 2016a). In the case of United States court decisions, such expert annotations between documents are also not publicly available. We construct two ground truth datasets from publicly available resources allowing the evaluation of more recommendations to mitigate the mentioned problems of cost, quantity, and reproducibility.

Open Case Book. With Open Case Book, the Harvard Law School Library offers a platform for making and sharing open-licensed casebooks.[1] The corpus consists of 222 casebooks containing 3,023 cases from 87 authors. Each casebook contains a manually curated set of topically

[1] `https://opencasebook.org`, last accessed: 18/01/2023

Table 4.1: Distribution of relevance annotations for Open Case Book and Wikisource.

	Relevance annotations per document						
Datasets ↓	**Mean**	**Std.**	**Min.**	**25%**	**50%**	**75%**	**Max.**
Open Case Book	86.42	65.18	2	48	83	111	1590
Wikisource	130.01	82.46	1	88	113	194	616

related court decisions, which we use as relevance annotations. The casebooks cover a range from broad topics (e.g., *Constitutional law*) to specific ones (e.g., *Intermediary Liability and Platforms' Regulation*). The decisions are mapped to full-texts and citations are retrieved from the Caselaw Access Project (CAP).[2] After duplicate removal and the mapping procedure, relevance annotations for 1,601 decisions remain.

Wikisource. We use a collection of 2,939 United States Supreme Court decisions from Wikisource as ground truth (Wikisource, 2020). The collection is categorized in 67 topics like *antitrust*, *civil rights*, and *amendments*. We map the decisions listed in Wikisource to the corpus from CourtListener.[3] The discrepancy between the two corpora decreases the number of relevance annotations to 1,363 court decisions.

Relevance classification. We derive a binary relevance classification from Open Case Book and Wikisource. When decisions A and B are in the same casebook or category, A is relevant for B and vice versa. Conversely, A and B are irrelevant recommendations for each other when they do not share the same casebook or category. Table 4.1 presents the distribution of relevance annotations. This relevance classification is limited since a recommendation might still be relevant despite not being assigned to the same topic as the seed decision. Thus, we consider the Open Case Book and Wikisource annotations as a silver standard rather than a gold standard.

4.1.2 Evaluation Methdology

The evaluation is conducted with a k nearest neighbor search in the embedding space of each method. We evaluate each method for its ability to represent any legal document d in our corpus as a numerical vector $\boldsymbol{d} \in \mathbb{R}^s$ with s denoting the vector size. First we obtain the vector representations (or document embeddings) for all documents in our corpus. To retrieve the recommendations for a seed or query document d_q, we compute the cosine similarities of the vectors of $\boldsymbol{d_q}$ and all other documents in the corpus. Finally, we select the top $k = 5$ documents with the highest similarity through a k nearest neighbor search of $\boldsymbol{d}_q$. We set $k = 5$ due to the user interface (Ostendorff et al., 2020a) into which the recommendations will be integrated.

Mean Average Precision (MAP) is the primary, and Mean Reciprocal Rank (MRR) is the second evaluation metric (Section 2.1.3). We compute MAP and MRR over a set of queries Q, whereby Q is equivalent to the seed decisions with $|Q_{\text{WS}}| = 1363$ available in Wikisource and $|Q_{\text{OCB}}| = 1601$ for Open Case Book.

[2] `https://case.law`, last accessed: 18/01/2023

[3] `https://courtlistener.com`, last accessed: 18/01/2023

In addition to the accuracy-oriented metrics, we evaluate the recommendations' coverage and Jaccard set similarity. The coverage is defined as in Equation 2.5 and the Jaccard similarity in Equation 2.13. Coverage and Jaccard similarity are computed for two methods a and b over their recommendation sets R_a and R_b.

4.1.3 Evaluated Methods

In the experiments, we evaluate 25 methods that we devide into three categories: Word vector-, Transformer-, and graph-based methods.

4.1.3.1 TF-IDF Baseline

As a baseline method, we use the sparse document vectors from TF-IDF[4] (Section 2.3.2) with $\boldsymbol{d} \in \mathbb{R}^{500,000}$, which are commonly used in related works (Kumar et al., 2011; Nanda et al., 2019) and performed well in our experiments with Wikipedia articles (Chapter 3).

4.1.3.2 Word Vector-based Methods

We evaluate the word vector-based methods GloVe (Pennington et al., 2014), fastText (Bojanowski et al., 2017; Joulin et al., 2017), and Paragraph Vectors (Le and Mikolov, 2014) (see Section 2.3.4 for details on these methods).

- Paragraph Vectors: A distributed bag-of-words model jointly trained on Open Case Book and Wikisource with $\boldsymbol{d} \in \mathbb{R}^{300}$ using a window size of 5, and the default hyperparameters from the Gensim framework (Rehurek and Sojka, 2010).
- GloVe: Pretrained GloVe word vectors $\boldsymbol{w} \in \mathbb{R}^{300}$ as provided by Pennington et al.; trained on Wikipedia and Gigaword.
- fastText: Pretrained fastText word vectors $\boldsymbol{w} \in \mathbb{R}^{300}$ as provided by Bojanowski et al.; trained on Wikipedia, UMBC webbase corpus and statmt.org news dataset.
- $\text{fastText}_{\text{Legal}}$ and $\text{GloVe}_{\text{Legal}}$: Custom legal word vectors based on GloVe and fastText but trained on the joint court decision corpus extracted from Open Case Book and Wikisource.[5]

To obtain a document vector $\boldsymbol{d}$ with GloVe and fastText, we compute the weighted average over its word vectors, $\boldsymbol{w_i}$, whereby the number of occurrences of the word i in d defines the weight c_i. Having word vectors from a generic corpus and our own legal corpus allows the investigation of the cross-domain applicability of the methods.

4.1.3.3 Transformer-based Methods

In the second method category, we employ language models from deep contextual text representations based on the Transformer architecture (Section 2.3.7.1):

- BERT (Devlin et al., 2019): The original model provided by Devlin et al. that was pretrained on Wikipedia and BookCorpus.

[4] We use the TF-IDF implementation from the scikit-learn framework (Pedregosa et al., 2011).
[5] The legal word vectors can be downloaded from our GitHub repository.

- Legal-JHU-BERT-base (Holzenberger et al., 2020): A BERT base model but fine-tuned on legal text from the CAP corpus.
- Legal-AUEB-BERT-base (Chalkidis et al., 2020): Another legal BERT model fine-tuned on the CAP corpus but also on other corpora (court cases and legislation from United States and European Union, and contracts).
- RoBERTa (Liu et al., 2019): An improved BERT variation trained on more data and with larger batches, and without the next sentence prediction task for pretraining.
- Sentence-BERT and Sentence RoBERTa (Reimers and Gurevych, 2019): Sentence Transformers are fine-tuned BERT and RoBERTa models in a Siamese setting (Bromley et al., 1993) to derive semantically meaningful sentence embeddings that can be compared using cosine similarity. The evaluated Sentence Transformers variations are *nli*- or *stsb*-version that are either fine-tuned on the SNLI and MNLI dataset (Bowman et al., 2015; Williams et al., 2018) or fine-tuned on the STS benchmark (Cer et al., 2017).
- Longformer (Beltagy et al., 2020): A Transformer with an attention mechanism that scales linearly with sequence length, allowing longer documents to be processed. We use the pretrained Longformer models as provided by Beltagy et al., which are limited to 4096 tokens (as opposed to the 512 tokens from the other Transformer models).

All Transformer models apply mean-pooling to derive document vectors. We experimented with other pooling strategies, but they yield significantly lower results. These findings agree with Reimers and Gurevych (2019). We investigate each Transformer in two variations depending on their availability and with respect to model size and document vector size (base with $\boldsymbol{d} \in \mathbb{R}^{768}$ and large with $\boldsymbol{d} \in \mathbb{R}^{1024}$).

4.1.3.4 Graph-based Methods

We explore graph-based methods utilizing the legal citation graph in which documents are nodes and edges correspond to citations to generate document vectors (see Section 2.4 for details on graph-based methods). The citation graph embeddings have the same vector size as the word-based methods with $\boldsymbol{d} \in \mathbb{R}^{300}$.

- DeepWalk (Perozzi et al., 2014): One of the first methods that borrowed word2vec's idea and applied it to graph embeddings.
- Walklets (Perozzi et al., 2017): Extends DeepWalk but explicitly encodes multi-scale node relationships to capture community structures with the graph.
- BoostNE (Li et al., 2019): A matrix factorization-based embedding technique combined with gradient boosting. Li et al. have shown that BoostNE performs well with citation graphs.
- Poincaré (Nickel and Kiela, 2017): An embedding method that is based on the hyperbolic space of the Poincaré ball model rather than the Euclidean space as in the aforementioned methods. Embeddings produced in hyperbolic space are naturally equipped to model hierarchical structures (Krioukov et al., 2010). Poincaré seems in particular promising since such hierarchical structures can also be found in the legal citation graph in the form of different topics or jurisdictions,

DeepWalk, Walklets, and BoostNe are implemented with the Karate Club framework (Rozemberczki et al., 2020)

4.1.3.5 Variations & Hybrid Methods

Given the conceptional differences in the evaluated methods, each method has its strength and weakness. The performance of a method may vary when document characteristics change. For further insights into these differences, we evaluate all methods with limited text, vector concatenation, and score summation:

Limited token count. Unlike the Transformers, the word vector-based methods have no maximum number of input tokens. Whether an artificial limitation of the document length improves or decreases the results is unclear. Longer documents might add additional noise to the representation and could lead to worse results (see Section 3.2.3.1). To make these two method categories comparable, we include additional variations of the word vector-based methods that are limited to the first 512 or 4096 tokens of the document. For instance, the method *fastText$_{Legal}$ (512)* is artificially limited to only the first 512 tokens.

Vector concatenation. Aside from the text length constraints, we explore hybrid methods that utilize text and citation information. Each of the single methods above yields a vector representation $\boldsymbol{d}$ for a given document d. We combine methods by concatenating their vectors. For example, the vectors from fastText $\boldsymbol{d}_{\text{fastText}}$ and Poincaré $\boldsymbol{d}_{\text{Poincaré}}$ can be concatenated as in Equation 4.1:

$$\boldsymbol{d} = \boldsymbol{d}_{\text{fastText}} || \boldsymbol{d}_{\text{Poincaré}} \tag{4.1}$$

The resulting vector size is the sum of the concatenated vector sizes, e.g., $s = 300 + 300 = 600$. Recommendations based on the concatenated methods are retrieved in the same fashion as the other methods, with cosine similarity.

Score summation. Moreover, we combine methods by adding up their cosine similarities as already done by Wang et al. (2016a). The combined score of two methods is the sum of the individual scores, e.g., for method x and method y the similarity of two documents d_a and d_b is computed as in Equation 4.2. Methods with score summation are denoted with $x + y$, e.g., Poincaré + fastText$_{\text{Legal}}$.

$$sim(\boldsymbol{d_a}, \boldsymbol{d_b}) = sim(\boldsymbol{d}_{\text{x}_a}, \boldsymbol{d}_{\text{x}_b}) + sim(\boldsymbol{d}_{\text{y}_a}, \boldsymbol{d}_{\text{y}_b}) \tag{4.2}$$

Lastly, we integrate citation information into Sentence Transformers analog to the fine-tuning procedure proposed by Reimers and Gurevych (2019). Based on the citation graph, we construct a dataset of positive and negative document pairs. Two documents d_a, d_b are considered positive samples when they are connected through a citation. Negative pairs are randomly sampled and do not share any citations. Sentence-Legal-AUEB-BERT-base is the Sentence Transformer model with Legal-AUEB-BERT-base as the base model and trained with citation information.

To summarize, we selected 25 methods ranging from traditional TF-IDF over Transformer models to citation graph embeddings. The selection represents a set of popular techniques that are

successfully applied for other NLP tasks (Sun et al., 2019; Yan et al., 2021) or other application domains like research papers (Ali et al., 2020; Mohamed Hassan et al., 2019) but so far had little impact on legal applications.

4.2 Evaluation

The results of the comparison of the 25 methods are shown in this section. We start by presenting the overall results of the offline evaluation based on Wikisource and Open Case Book. To obtain further insights, we also analyze the results concerning document length and citation count, compare the coverage and similarity of the recommendations, and validate our quantitative findings with a manual analysis of a randomly sampled recommendation set. For the offline evaluation, we obtain a list of recommendations for each input document and each method and then compute precision, recall, MRR, MAP, and coverage accordingly.

Table 4.2: Overall scores for top $k = 5$ recommendations from Open Case Book and Wikisource as precision, recall, MRR, MAP and coverage for the 25 methods. The methods are divided into baseline, word vector-based, Transformer-based, graph-based, and hybrid. High scores according to the exact values are underlined (or **bold** for category-wise).

Datasets →	**Open Case Book**					**Wikisource**				
Methods ↓	**Prec.**	**Rec.**	**MRR**	**MAP**	**Cov.**	**Prec.**	**Rec.**	**MRR**	**MAP**	**Cov.**
TF-IDF	0.320	0.032	0.363	0.020	0.487	0.318	0.026	0.389	0.015	0.446
Paragraph Vectors	0.555	0.056	0.729	0.049	**0.892**	0.477	0.036	0.629	0.030	**0.841**
fastText	0.532	0.053	0.713	0.045	0.811	0.422	0.031	0.581	0.025	0.772
$\text{fastText}_{\text{Legal}}$	**0.574**	**0.059**	**0.739**	**0.050**	0.851	**0.478**	**0.037**	**0.631**	**0.031**	0.815
$\text{fastText}_{\text{Legal}}$ (512)	0.394	0.037	0.591	0.028	0.835	0.433	0.034	0.587	0.027	0.809
$\text{fastText}_{\text{Legal}}$ (4096)	0.552	0.054	0.727	0.045	0.867	0.466	0.035	0.620	0.029	0.817
GloVe	0.536	0.054	0.702	0.046	0.814	0.412	0.033	0.577	0.026	0.789
$\text{GloVe}_{\text{Legal}}$	0.564	0.057	0.724	0.048	0.834	0.461	0.037	0.621	0.030	0.804
BERT-base	0.253	0.021	0.428	0.015	0.815	0.323	0.021	0.485	0.015	0.784
BERT-large	0.270	0.022	0.443	0.016	0.841	0.364	0.023	0.530	0.018	0.794
Legal-JHU-BERT-base	0.295	0.025	0.482	0.018	0.848	0.371	0.027	0.537	0.020	0.796
Legal-AUEB-BERT-base	0.331	0.028	0.506	0.021	0.884	**0.401**	**0.027**	**0.573**	**0.022**	0.813
Longformer-base	0.382	0.033	0.572	0.026	**0.892**	0.329	0.020	0.514	0.016	**0.841**
Longformer-large	**0.419**	**0.039**	**0.614**	**0.031**	0.885	0.360	0.023	0.535	0.018	0.826
RoBERTa-large	0.305	0.026	0.481	0.019	0.843	0.387	0.026	0.553	0.020	0.782
Sentence-BERT-large-nli	0.206	0.018	0.352	0.013	0.872	0.273	0.017	0.443	0.012	0.782
BoostNE	0.258	0.022	0.442	0.016	0.800	0.248	0.016	0.398	0.013	0.832
DeepWalk	0.267	0.028	0.473	0.021	0.818	0.364	0.030	0.533	0.025	**0.856**
Poincaré	0.447	**0.044**	0.629	**0.036**	**0.930**	0.465	0.038	0.598	**0.031**	0.837
Walklets	**0.448**	0.043	**0.636**	0.035	0.816	**0.470**	**0.038**	**0.611**	0.031	0.826
Poincaré ‖ $\text{fastText}_{\text{Legal}}$	0.473	0.048	0.656	0.041	0.737	**0.505**	**0.041**	0.638	**0.035**	0.818
Longformer-large ‖ $\text{fastText}_{\text{Legal}}$	0.451	0.043	0.642	0.035	0.876	0.383	0.025	0.547	0.020	0.829
Poincaré + $\text{fastText}_{\text{Legal}}$	**0.571**	**0.058**	**0.746**	**0.050**	0.860	0.497	0.040	**0.646**	0.034	0.835
Poincaré + Longformer-large	0.419	0.039	0.630	0.033	0.885	0.360	0.023	0.548	0.019	0.826
Sent.-Legal-AUEB-BERT-base	0.438	0.039	0.603	0.031	**0.917**	0.471	0.038	0.602	0.032	**0.849**

4.2.1 Overall Results

Table 4.2 presents the overall evaluation metrics for 25 methods and the two datasets. From the non-hybrid methods, $\text{fastText}_{\text{Legal}}$ yields the highest MAP score with 0.05 on Open Case Book, whereas on Wikisource, $\text{fastText}_{\text{Legal}}$, Poincaré, and Walklets all achieve the highest MAP score of 0.031. The hybrid method of Poincaré || $\text{fastText}_{\text{Legal}}$ outperforms the non-hybrids for Wikisource with 0.035 MAP. For Open Case Book, the MAP of Poincaré + $\text{fastText}_{\text{Legal}}$ and $\text{fastText}_{\text{Legal}}$ are equally high.

We compared in total 41 methods but we remove 16 less insightful methods from Table 4.2 for better comprehensibility (the results for the excluded methods can be found in the supplementary materials[9]). From the word vector-based methods, we discard the 512 and 4096 tokens variations of Paragraph Vectors, GloVe and $\text{GloVe}_{\text{Legal}}$, as they show a similar performance deterioration as $\text{fastText}_{\text{Legal}}$. The base versions of some Transformers are also excluded in favor of the better-performing large versions. Similarly, only the Sentence-BERT-large-nli version of the Sentence Transformers is shown, since all other Sentence Transformers yielded a poor performance. For the hybrid variations, we show only the best methods. We also tested Node2Vec (Grover and Leskovec, 2016) but exclude it given its low MAP scores.

Regarding the word vector-based methods, we see that the methods which are trained on the legal corpus (Paragraph Vectors, $\text{fastText}_{\text{Legal}}$, $\text{GloVe}_{\text{Legal}}$) perform similarly well with a minor advantage by $\text{fastText}_{\text{Legal}}$. Moreover, there is a margin between the generic and legal word vectors even though the legal word vectors are trained on a small corpus compared to ones from the generic vectors. The advantage of Paragraph Vectors over TF-IDF is consistent with the results from related work, e.g., Mandal et al. (2017). Limiting the document length to 512 or 4096 decreases the effectiveness of $\text{fastText}_{\text{Legal}}$. A limit of 512 tokens decreases the MAP score to 59% compared to all tokens on Open Case Book. With 4096 tokens, the performance decline is only minor (90% compared to all tokens). The token limitation effect is also larger on Open Case Book than Wikisource. The 4096 tokens version of $\text{fastText}_{\text{Legal}}$ even outperforms all Transformer methods.

Longformer-large is the best Transformer for Open Case Book with 0.031 MAP. For Wikisource, Legal-AUEB-BERT achieves the highest MAP of 0.022, closely followed by Legal-JHU-BERT. The Longformer's theoretical advantage of processing 4096 instead of 512 tokens does not lead to better results for Wikisource, for which even BERT scores the same MAP of 0.018. We generally observe that large models outperform their base counterparts.[6] Likewise, RoBERTa has higher scores than BERT, as Liu et al. (2019) suggested. From the Transformers category, Sentence Transformers yield the worst results. We assume that fine-tuning the models on datasets like NLI or STSB does not increase the performance since the models do not generalize well to other domains. However, the language model fine-tuning from Legal-JHU-BERT and Legal-AUEB-BERT improves performance, whereby Legal-AUEB-BERT generally outperforms Legal-JHU-BERT. For Open Case Book, Legal-AUEB-BERT is the best model in the Transformer category in terms of MAP even though it is only used as the base version.

In the citation category, Poincaré and Walklets are the best methods by a large margin. For Wikisource, the two graph-based methods achieve the same MAP of 0.031 as $\text{fastText}_{\text{Legal}}$. Compared to the word vector-based methods, the citation methods generally perform better on Wikisource than on Open Case Book.

[6] Legal-JHU-BERT and Legal-AUEB-BERT are only available as base versions.

Combining text and citations improves the recommendation performance, as we can see in the category of hybrid methods. For Open Case Book, the score summation Poincaré + fastText$_{\text{Legal}}$ has the same MAP of 0.05 as fastText$_{\text{Legal}}$ but a higher MRR of 0.746. The MRR of Poincaré + fastText$_{\text{Legal}}$ is even higher than the MRR of its sub-methods Poincaré (0.629) and fastText$_{\text{Legal}}$ (0.739) individually. The concatenation of Poincaré $\|$ fastText$_{\text{Legal}}$ is with 0.035 MAP the best method on Wikisource. Using citation as a training signal as in Sentence-Legal-AUEB-BERT also improves the performance but not as much as concatenation or summation. When comparing the three hybrid variations, score summation achieves overall the best results. In the case of Wikisource, the concatenation's scores are below its sub-methods, while summation has at least the best sub-methods score. Moreover, combining a pair of text-based methods such as Longformer-large and fastText$_{\text{Legal}}$ never improves its sub-methods.

4.2.2 Impact of Document Properties

To complement the overall results, we also evaluate the impact of document length and citation count on the recommendation performance.

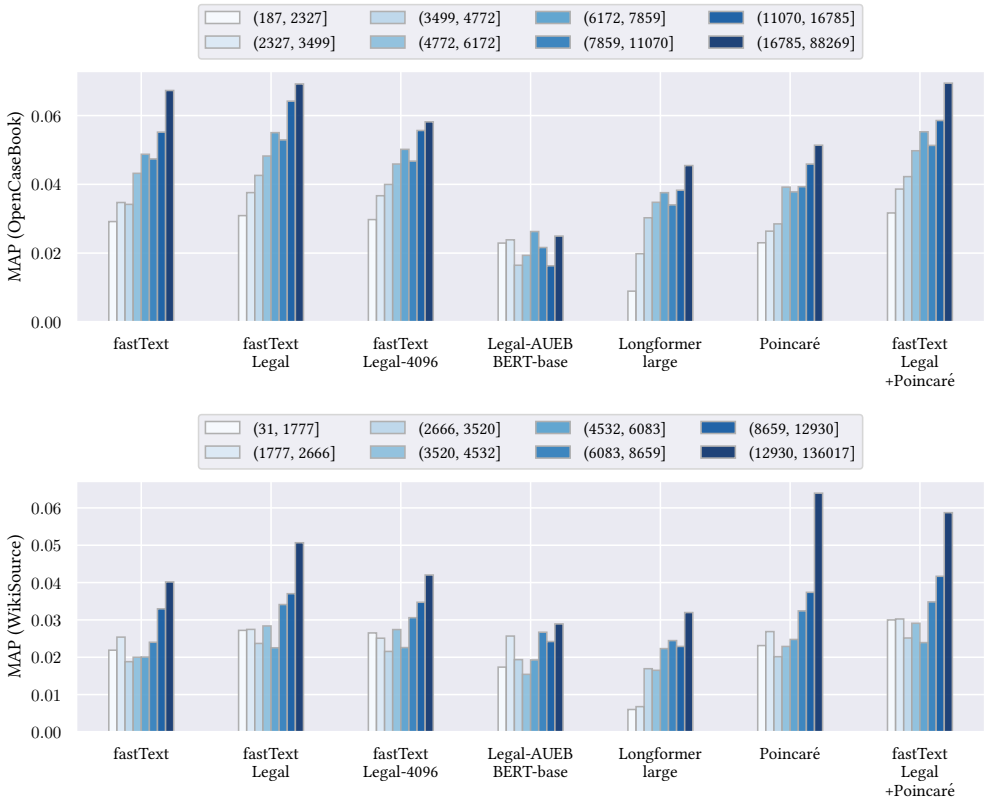


Figure 4.1: MAP scores with respect to words in the seed document of Open Case Book (top) and Wikisource (bottom). The more words, the better the results, no peak at medium length. fastText$_{\text{Legal}}$ outperforms both Legal-BERT and Longformer even for short documents.

4.2.2.1 Document Length

The effect of the document length on the performance in terms of MAP is displayed in Figure 4.1. We group the seed documents into eight equal-sized buckets (each bucket represents an equal number of documents) depending on the word count in the document text to make the two datasets comparable. The interval of each bucket is shown in the legend.

Both datasets (Open Case Book and Wikisource) present a similar outcome. The MAP increases as the word count increases. Table 4.2 presents the average overall documents; therefore, the overall best method is not equal to the best method in some subsets. For instance, Paragraph Vectors achieve the best results for several buckets, e.g., 4772-6172 words in Open Case Book or 6083-8659 words in Wikisource (the results of Paragraph Vectors are not shown in the figure). The text limitation of $\text{fastText}_{\text{Legal}}$ (4096 tokens) in comparison to fastText is also clearly visible. The performance difference between the two methods increases as the document length increases. For the first buckets with less than 4096 words, e.g., 187-2327 words in Open Case Book, one could expect no difference since the limitation does not affect the seed documents in these buckets. However, we observe a difference since target documents are not grouped into the same buckets. Remarkable is that the performance difference for very long documents is less substantial. When comparing Longformer-large and Legal-AUEB-BERT, we also see an opposing performance shift with changing word count. While Legal-AUEB-BERT's scores are relatively stable throughout all buckets, Longformer depends more on the document length. On the one hand, Longformer performs worse than Legal-AUEB-BERT for short documents, i.e., 187-2327 words in Open Case Book, and 31-1777 words in Wikisource. On the other hand, for documents with more words, Longformer mostly outperforms Legal-AUEB-BERT by a large margin. The graph-based method Poincaré is as well affected by the document length. However, this effect is due to a positive correlation between word count and citation count.

4.2.2.2 Citation Count

Figure 4.2 shows the effect of the number of in- and out-citations (i.e., edges in the citation graph) on the MAP score based on equal-sized buckets. The citation analysis for Wikisource confirms the word count analysis. More data correlates with higher MAP scores. The only exception can be found for Open Case Book, where the performance of the graph-based methods peaks at 31-51 citations and even decrease at 67-89 citations. When comparing Poincaré and Walklets, no superior method and no dependency pattern are visible. The performance effect on DeepWalk is more substantial. The number of citations must be above a certain threshold to allow DeepWalk to achieve competitive results. For Open Case Book, the threshold is at 51-67 citations, and for Wikisource, it is at 30-50 citations. Figure 4.2 also shows the on average higher MAP of Poincaré + $\text{fastText}_{\text{Legal}}$ in comparison to the other approaches. Graph-based methods require citations to work, whereas text methods do not have this limitation (see 0-14 citations for Open Case Book). When no citations are available, citation graph-based methods cannot recommend documents, whereas the text methods still work (see 0-14 citations for Open Case Book).

Our graph-based methods use only a fraction of the true citation data (70,865 citations in Open Case Book and 331,498 citations in Wikisource), because of our limitations to the documents available in the silver standards. For comparison, the most-cited decision[7] from CourtListener

[7] `https://www.courtlistener.com/opinion/111170/strickland-v-washington/`, last accessed: 18/01/2023

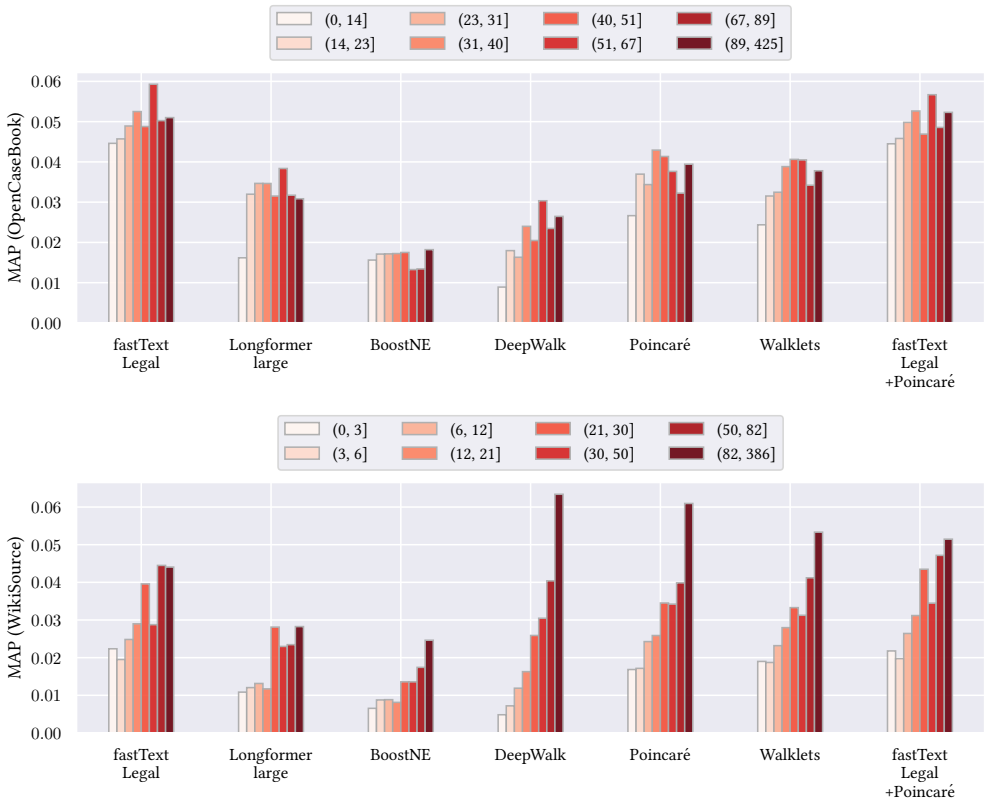


Figure 4.2: MAP scores with respect to citation count of seed documents for Open Case Book (top) and Wikisource (bottom). Among graph-based methods, Poincaré and Walklets perform on average the best, while DeepWalk outperforms them only for Wikisource and when more than 82 citations are available (rightmost bucket).

(the underlying corpus of Wikisource) has 88,940 citations, whereas in experimental data of Wikisource the maximum number of in- and out-citations is 386. As a result, we expect the graph-based methods, especially DeepWalk, to work even better when applied to the full corpus.

4.2.3 Coverage and Similarity of Recommendations

In addition to the accuracy-oriented metrics, Table 4.2 reports also the coverage of the recommendation methods. Recommender systems for an expert audience should not focus on a small set of the most popular items but rather provide high coverage of the whole item collection. However, coverage alone does not account for relevancy; therefore, it must be contextualized with other metrics, e.g., MAP.

Overall, two graph-based methods yield the highest coverage for both datasets, i.e., Poincaré for Open Case Book and DeepWalk for Wikisource. In particular, Poincaré has not only a high coverage but also high MAP scores. However, the numbers do not indicate that graph-based methods have generally a higher coverage since the text-based Paragraph Vectors or Longformer-base also achieve considerably high coverage. The lowest coverage has by far the TF-IDF baseline. Notable, the hybrid methods with concatenation and summation have a different effect on the coverage as on the accuracy metrics. While the hybrid methods generally yield a higher

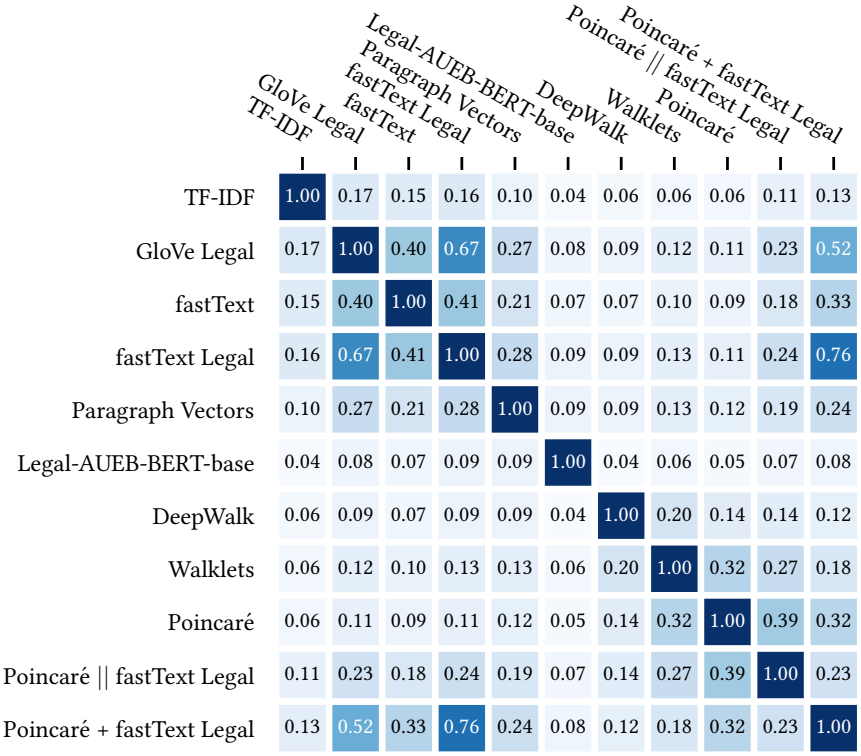


Figure 4.3: Jaccard index for similarity or diversity of two recommendation sets (average over all seeds from the two datasets). Most overlap can be found among the word vector based methods.

MAP, their coverage is lower compared to their sub-methods. Only, the Sentence-Legal-AUEB-BERT-base yields a higher coverage compared to Legal-AUEB-BERT-base.

Besides the coverage, we also analyze the similarity or diversity of the recommendations between pairs of methods. Figure 4.3 shows the similarity measured as the Jaccard index for selected methods. Method pairs with $J(a,b) = 1$ have identical recommendations, whereas $J(a,b) = 0$ means no common recommendations. Generally speaking, the similarity of all method pairs is considerably low ($J < 0.8$). The highest similarity can be found between a hybrid method and one of its sub-methods, e.g., Poincaré + fastText$_{\text{Legal}}$ and fastText$_{\text{Legal}}$ with $J = 0.76$. Apart from that, substantial similarity can be only found between pairs from the same category. For example, the pair of the two text-based methods of GloVe$_{\text{Legal}}$ and fastText$_{\text{Legal}}$ yields $J = 0.67$. Graph-based methods tend to have lower similarity compared to the text-based methods, whereby the highest Jaccard index between two graph-based methods is achieved for Walklets and Poincaré with $J = 0.32$. But also within the text-based category, we see little overlap when comparing for example Legal-AUEB-BERT with fastText$_{\text{Legal}}$. Like the coverage metric, the Jaccard index should be considered in relation to the accuracy results. GloVe$_{\text{Legal}}$ and fastText$_{\text{Legal}}$ yield equally high MAP scores, while having also a high recommendation set similarity. In contrast, the MAP for Wikisource from fastText$_{\text{Legal}}$ and Poincaré is equally high, too. However, their recommendation's similarity is low with $J = 0.11$. Consequently, fastText$_{\text{Legal}}$ and Poincaré provide relevant recommendations that are diverse from each other. This explains the good performance of their hybrid combination.

4.2.4 Manual Sample Analysis

Due to the lack of openly available gold standards, we rely only on silver standards. Thus, we conduct an additional qualitative evaluation with domain experts to estimate the quality of our silver standards.

Table 4.3: Open Case Book example recommendations from $\text{fastText}_{\text{Legal}}$ and Poincaré for *Mugler v. Kansas* with relevance annotations by the silver standard (S) and domain expert (D).

	#	Recommendations	Year	S	D
$\text{fastText}_{\text{Legal}}$	1	Yick Wo v. Hopkins	1886	N	N
	2	Munn v. Illinois	1876	Y	Y
	3	LS. Dealers' & Butchers' v. Crescent City LS.	1870	N	Y
	4	Butchers' Benevolent v. Crescent City LS.	1872	Y	Y
	5	Lochner v. New York	1905	Y	Y
Poincaré	1	Yick Wo v. Hopkins	1886	N	N
	2	Allgeyer v. Louisiana	1897	Y	Y
	3	Calder v. Wife	1798	N	N
	4	Davidson v. New Orleans	1877	Y	Y
	5	Muller v. Oregon	1908	Y	Y

Table 4.3 and 4.4 lists one of the randomly chosen seed decisions (Mugler vs. Kansas[8]), and five recommended similar decisions, each from $\text{fastText}_{\text{Legal}}$ and Poincaré. In Mugler vs. Kansas (1887), the court held that Kansas could constitutionally outlaw liquor sales with constitutional issues raised on substantive due process (Fourteenth Amendment) and takings (Fifth Amendment). We provide a detailed description of the cases and their relevance annotations in the Appendix of Ostendorff et al. (2021a).

The sample verification indicates the overall usefulness of both text-based and graph-based methods and does not contradict our quantitative findings. Each of the identified cases has a legally important connection to the seed case (either the Fourteenth Amendment or Fifth Amendment), although it is difficult to say whether the higher-ranked cases are more similar along an important topical dimension. The rankings do not appear to be driven by facts presented in the case as most of them have not to do with alcohol bans. Only Kidd vs. Pearson (1888) is about liquor sales as the seed decision.

Also, the samples do not reveal considerable differences between text- and graph-based similarity. The lack of this difference contradicts the findings from the Wikipedia experiments in Chapter 3. This could indicate that citations are differently used in the legal literature compared to Wikipedia, which is most likely true, or that the manual analysis was performed on too few samples to produce representative results. Regarding the silver standards, the domain expert annotations agree in 14 of 20 cases (70%). In only two cases, the domain expert classifies a recommendation as irrelevant despite being classified as relevant in the silver standard.

[8] `https://www.courtlistener.com/opinion/92076/mugler-v-kansas/`, last accessed: 18/01/2023

Table 4.4: Wikisource example recommendations from fastText$_{\text{Legal}}$ and Poincaré for *Mugler v. Kansas* with relevance annotations by the silver standard (S) and domain expert (D).

	#	Recommendations	Year	S	D
fastText$_{\text{Legal}}$	1	Kidd v. Pearson	1888	N	Y
	2	Lawton v. Steele	1894	N	Y
	3	Yick Wo v. Hopkins	1886	N	N
	4	Geer v. Connecticut	1896	N	Y
	5	Groves v. Slaughter	1841	Y	N
Poincaré	1	Rast v. Van Deman & Lewis Co.	1916	Y	N
	2	County of Mobile v. Kimball	1881	N	N
	3	Brass v. North Dakota Ex Rel. Stoeser	1894	Y	Y
	4	Erie R. Co. v. Williams	1914	Y	Y
	5	Hall v. Geiger-Jones Co.	1917	Y	Y

4.3 Discussion

Our experiments explore the applicability of the latest advances in representation learning research to the use case of legal literature recommendations. Existing studies on legal recommendations typically rely on small-scale user studies and are therefore limited in the number of approaches that they can evaluate (Section 2.1.1). For this study, we utilize relevance annotations from two publicly available sources, i.e., Open Case Book and Wikisource. These annotations not only enable us to evaluate the recommendations of 2,964 documents but also the comparison of in total 41 methods and their variations of which 25 methods are presented in this chapter.

Our extensive evaluation shows a large variance in the recommendation performance. Such a variance is known from other studies (Beel et al., 2016a). There is no single method that yields the highest scores across all metrics and all datasets. Despite that, fastText$_{\text{Legal}}$ is the best single method on average. fastText$_{\text{Legal}}$ yields the highest MAP for Open Case Book, while for Wikisource only hybrid methods outperform fastText$_{\text{Legal}}$. Also, the coverage of fastText$_{\text{Legal}}$ is considerably high for both datasets. Simultaneously, fastText$_{\text{Legal}}$ is robust to corner cases since neither very short nor very long documents reduce fastText$_{\text{Legal}}$'s performance substantially. These results confirm the findings from Arora et al. (2017) that average word vectors are "simple but tough-to-beat baseline". Regarding baselines, our TF-IDF baseline yields one of the worst results. In terms of accuracy metrics, only some Transformers are worse than TF-IDF, but especially TF-IDF's coverage is the lowest by a large margin. With a coverage below 50%, TF-IDF fails to provide diverse recommendations that are desirable for legal literature research.

The transfer of research advances to the legal domain is one facet of our experiments. Thus, the performance of Transformers and citation embeddings is of particular interest. Despite the success of Transformers for many NLP tasks, Transformers yield the worst results on average for representing lengthy documents written in legal English. The other two method categories, word vector-based and graph-based methods, surpass Transformers.

The word vector-based methods achieve overall the best results among the non-hybrid methods. All word vector-based methods with in-domain training, i.e., Paragraph Vectors, $\text{fastText}_{\text{Legal}}$, and $\text{GloVe}_{\text{Legal}}$, perform similarly well with a minor advantage by $\text{fastText}_{\text{Legal}}$. Their similar performance aligns with the large overlap among their recommendations. Despite a small corpus of 65,635 documents, the in-domain training generally improves the performance as the gap between the out-of-domain fastText and $\text{fastText}_{\text{Legal}}$ shows. Given that the training of custom word vectors is feasible on commodity hardware, in-domain training is advised. More significant than the gap between in- and out-of-domain word vectors is the effect of limited document lengths. For Open Case Book, the $\text{fastText}_{\text{Legal}}$ variation limited to the first 512 tokens has only 52% of the MAP of the full-text method. For Wikisource, the performance decline also exists but is less significant. This effect highlights the advantage of the word vector-based methods in that they derive meaningful representations of documents with arbitrary lengths.

The evaluated Transformers cannot process documents of arbitrary length but are either limited to 512 or 4096 tokens. This limitation contributes to Transformers' low performance. For instance, Longformer-large's MAP is almost twice as high as BERT-large's MAP on Open Case Book. However, for Wikisource, both models yield the same MAP scores. For Wikisource, the in-domain pretraining has a larger effect than the token limit since Legal-AUEB-BERT achieves the best results among the Transformers. Regarding the Transformer pretraining, the difference between Legal-JHU-BERT and Legal-AUEB-BERT shows the effect of the two pretraining approaches. The corpora and the hyperparameter settings used during pretraining are crucial. Even though Legal-JHU-BERT was exclusively pretrained on the CAP corpus, which has a high overlap with Open Case Book, Legal-AUEB-BERT still outperforms Legal-JHU-BERT on Open Case Book.

Another reason for the poor performance of the Transformer models is that their embedding space suffers from being anisotropic, as the work from Li et al. (2020) suggests. An anisotropic embedding space is poorly defined in some areas making a similar search more error-prone. Given these findings, we expect the performance of Transformers could be improved by increasing the token limit beyond the 4096 tokens, by additional in-domain pretraining, and by addressing anisotropic issues (e.g., through a contrastive learning objective as shown in Chapter 5). Such improvements are technically possible but add significant computational effort. In contrast to word vectors, Transformers are not trained on commodity hardware but on GPUs. Especially long-sequence Transformers such as the Longformer require GPUs with large memory. Such hardware may not be available in production deployments. Moreover, the computational effort must be seen in relation to the other methods. Put differently, even $\text{fastText}_{\text{Legal}}$ limited to 512 tokens outperforms all Transformers.

Concerning the citation embeddings, we consider Poincaré, closely followed by Walklets, as the best method. In particular, the two methods outperform the other citation methods for documents even when only a few citations are available, which makes them attractive for legal research. Poincaré also provides the highest coverage for Open Case Book, emphasizing its quality for literature recommendations. For Wikisource, DeepWalk has the highest coverage despite yielding generally low accuracy scores. As Figure 4.2 shows, DeepWalk's MAP score improves substantially as the number of citations increases. Therefore, we expect that DeepWalk and other citation methods would perform even better when applied to larger citation graphs.

While the sample analysis reveal no considerable difference between text-based and graph-based similarity (Section 4.2.4), the overall recommendation set similarity between the graph-based

methods and the text-based methods is remarkably low (Figure 4.3). This indicates that the different method categories yield also a different notion of document similarity since they produce fundamentally different recommendations. This finding aligns with our qualitative comparison of TF-IDF and CPA in Chapter 3. Also, it motivates the hybrid combination of text-based and graph-based methods.

Related work has already shown the benefit of hybrid methods for literature recommendations (Bhattacharya et al., 2020a; Wiggers and Verberne, 2019). Our experiments confirm these findings. The simple approaches of score summation or vector concatenation can improve the results. In particular, Poincaré + $\text{fastText}_{\text{Legal}}$ never deteriorates the performance. Instead, it improves the performance for corner cases in which one of the sub-methods perform poorly. Vector concatenation has mixed effects on the performance, e.g., positive effect for Wikisource and negative effect for Open Case Book. Using citations as training data in Sentence Transformers can also be considered a hybrid method that improves the recommendation performance. However, this requires additional effort to train a new Sentence Transformer model.

In general, our results highlight the benefit of combing text and graph information for legal literature recommendations. Poincaré $\|$ $\text{fastText}_{\text{Legal}}$ achieves the best precision, recall, and MAP on Wikisource. Poincaré + $\text{fastText}_{\text{Legal}}$ is on par with $\text{fastText}_{\text{Legal}}$ on Open Case Book in terms of MAP (best MRR), when scores are rounded to three decimals. Coverage is also high for both hybrid methods. These improvements can be achieved even with simple approaches like score summation or vector concatenation, which come only with a small computational overhead compared to the Transformer models. Thus, hybrid methods are generally advisable.

As we discuss in Section 4.1.1, we consider Open Case Book and Wikisource rather a silver than a gold standard. With the qualitative evaluation, we mitigate the risk of misinterpreting the quantitative results, whereby we acknowledge our small sample size. The overall agreement with the domain expert is high. The expert tends to classify more recommendations as relevant than the silver standards, i.e., relevant recommendations are missed. This explains the relatively low recall from the quantitative evaluation. In a user study, we would expect only minor changes in the ranking of methods with similar scores, e.g., $\text{fastText}_{\text{Legal}}$ and $\text{GloVe}_{\text{Legal}}$. The overall ranking among the methods would remain the same. The benefit of our silver standards is the number of available relevance annotations. The number of annotations in related user studies is up to 50 annotations rather low. Instead, our silver standards provide magnitude more annotations for recommendation relevance. Almost 3,000 relevance annotations enable evaluations regarding text length, citation count, or other properties that would be otherwise magnitudes more difficult. Similarly, user studies are difficult to reproduce as their data is mostly unavailable (Beel et al., 2016a). The open license of the silver standards allows sharing of all evaluation data and, therefore, contributes to more reproducibility. In summary, the proposed datasets bring great value to the field, overcoming eventual shortcomings.

4.4 Summary of the Chapter

This chapter empirically evaluated 25 document representation methods in the context of legal literature recommendations. Following Chapter 3, this chapter continued comparing existing aspect-free document similarity methods as formulated in Research Task **I**. As opposed to Chapter 3 that studied primarily classical count-based approaches like CPA, this chapter focused exclusively on evaluating vector representation techniques. We discarded CPA in favor of state-

of-the-art graph embedding methods and also due to its lower performance compared to TF-IDF. We conducted the study based on the common content-based recommendation approach of first learning vector representations from text and graph information and then recommending the k nearest neighbors based on the cosine similarity of their vector representations.

We evaluated the 25 methods over two document corpora containing 2,964 documents (1,601 from Open Case Book and 1,363 from Wikisource), which differentiates our study from the small-scale studies previously conducted in the legal domain. We underpinned our findings with a sample-based qualitative evaluation. Our analysis of the results revealed $\text{fastText}_{\text{Legal}}$ (averaged fastText word vectors trained on our in-domain corpora) as the best-performing single method.

In particular, the results also showed that graph-based and text-based recommendations have a low overlap and that the individual methods are vulnerable to certain dataset characteristics like text length and the number of available citations. To mitigate the weakness of single methods and to increase recommendation diversity, we proposed simple hybrid methods like the score summation of $\text{fastText}_{\text{Legal}}$ and Poincaré that outperformed all individual methods. The hybrid methods not only improved the accuracy-oriented evaluation metrics with little computational overhead but also increased the coverage of the recommendations. Thus, hybrid methods are generally advisable. Combining methods improves the recommendations since the individual methods implicitly address different aspects of the document content. This outcomes confirms the findings from Chapter 3 and shows that state-of-the-art document representations and legal literature recommendations are also affected by the lack of aspect information.

Although there are limitations in the experimental evaluation due to the lack of openly available ground truth data, we could draw meaningful conclusions about the behavior of text-based and graph-based document embeddings in the context of legal document recommendation. The chapter's source code, models, and datasets are openly available.[9]

[9] `https://github.com/malteos/legal-document-similarity`, last accessed: 18/01/2023

Chapter 5

Hybrid Research Paper Representations

The last two chapters have shown that text-based and graph-based methods address different aspects of the document content and that a simple hybrid combination of text and graph information, such as vector concatenation or score summation, already improves recommendation performance. In this chapter, we focus on the Research Task **II** and design a method, which combines text and graph information. To be precise, we will explore how citation information can be incorporated into a text-based document encoder for research papers. Having an encoder, which produces document representations from text input alone, has the advantage that it can also be applied in use cases where no or only little graph information is available. The approaches from Chapter 4 like vector concatenation would not work under these settings. This chapter's content is based on Ostendorff et al. (2022b).

"Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings" by **Malte Ostendorff**, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. In: *The 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

Large pretrained language models achieve state-of-the-art results on many NLP tasks (Rogers et al., 2020). However, the sentence or document embeddings derived from these language models are of lesser quality compared to simple baselines like fastText (as shown in Chapter 4), as their embedding space suffers from being anisotropic (Li et al., 2020). In other words, their embedding space is poorly defined in some areas.

One approach that has recently gained attention is the combination of language models with contrastive fine-tuning to improve the semantic similarity between document representations (Gao et al., 2021; Wu et al., 2020). These contrastive methods learn to distinguish between pairs of similar and dissimilar documents (positive and negative samples). Recent works showed that the selection of these positive and negative samples is crucial for efficient contrastive learning of document representations (Rethmeier and Augenstein, 2021a; Rethmeier and Augenstein, 2021b; Shorten et al., 2021; Tian et al., 2020b).

Building upon these findings, this chapter focuses on learning hybrid document representations from text and citations for research papers. The core distinguishing feature of the scientific domain is the presence of citation information that complement the textual information. Existing methods like SciBERT (Beltagy et al., 2019) pretrain a Transformer language model on domain-specific text but neglect citations. The current state-of-the-art SPECTER by Cohan et al. (2020) uses citation information to generate positive and negative samples for contrastive fine-tuning of a SciBERT language model. SPECTER relies on 'citations by the query paper' as a discrete signal for similarity, i.e., positive samples are cited by the query while negative ones are not cited.

However, SPECTER's use of citations has its pitfalls. Considering only one citation direction may cause positive and negative samples to collide since a paper pair could simultaneously be treated as a positive and negative instance. Also, relying on a single citation as a discrete similarity

signal is subject to noise, for example, when citations may reflect politeness and policy rather than semantic similarity (Pasternack, 1969) or related papers lack a direct citation (Gipp and Beel, 2009). This discrete cut-off to similarity is counter-intuitive to *continuous* similarity-based learning. Instead, the generation of *non-colliding* contrastive samples should be based on a continuous similarity function that allows us to find semantically similar papers, even without direct citations. This chapter introduces the SciNCL approach (**sci**entific document **n**eighborhood **c**ontrastive **l**earning) that addresses the aforementioned issues by generating contrastive samples based on citation embeddings. Citation embeddings incorporate the full citation graph and provide a continuous, undirected, and less noisy similarity signal that generates arbitrary easy-to-hard positive and negative samples. To validate these assumptions, this chapter seeks to answer the following research questions.

Research questions

RQ1: Are samples generated from neighboring citation embeddings more suitable than samples from discrete citations for the contrastive learning of scientific document representations?

RQ2: How does the difficulty of contrastive samples affect the quality of the learned document representations and the training efficiency?

We conduct extensive experiments based on the SciDocs benchmark to provide answers to these research questions. Specifically, we compare SciNCL against existing state-of-the-art document representation methods and analyze the effect of its hyperparameters.

In summary, this chapter makes the following main contributions:

1. We propose neighborhood contrastive learning for scientific document representations with citation graph embeddings (SciNCL) based on contrastive learning theory insights.
2. We sample positive (similar) and negative (dissimilar) papers from the k nearest neighbors in the citation graph embedding space, such that positives and negatives do not collide but are also hard to learn.
3. We compare against the state-of-the-art approach SPECTER (Cohan et al., 2020) and other strong methods on the SciDocs benchmark and find that SciNCL outperforms SPECTER on average and on 9 of 12 metrics.
4. Finally, we demonstrate that with SciNCL, using only 1% of the triplets for training, starting with a general-domain language model, or training only the bias terms of the model is sufficient to outperform the baselines.

This chapters's code and models are publicly available.[1]

The remainder of this chapter is structured as follows: First, we introduce the general methodology, i.e., the concept of contrastive neighborhood learning, the datasets, and the evaluated methods. Subsequently, we present the overall results in Section 5.2.1, the analysis of sample difficulty in Section 5.2.2, and other ablations in Section 5.2.3. In Section 5.3, we discuss the results of all evaluations. Finally, we summarize the main findings of this chapter.

[1] `https://github.com/malteos/scincl`, last accessed: 18/01/2023

5.1 Methodology

Our goal is to learn citation-informed representations for scientific documents. To do so, we sample three document representation vectors and learn their similarity. For a given query paper vector $\boldsymbol{d}^{Q}$, we sample a positive (similar) paper vector $\boldsymbol{d}^{+}$ and a negative (dissimilar) paper vector $\boldsymbol{d}^{-}$. This produces a 'query, positive, negative' triplet $(\boldsymbol{d}^{Q}, \boldsymbol{d}^{+}, \boldsymbol{d}^{-})$ – represented by (★, ✚, ▬) in Figure 5.1. To learn paper similarity, we need to define three components: how to calculate document vectors $\boldsymbol{d}$ for the loss over triplets $\mathscr{L}$ (Section 5.1.1), how citations provide similarity between papers (Section 5.1.2), and how negative and positive papers $(\boldsymbol{d}^{-}, \boldsymbol{d}^{+})$ are sampled as (dis-)similar documents from the neighborhood of a query paper $\boldsymbol{d}^{Q}$ (Section 5.1.2.1).

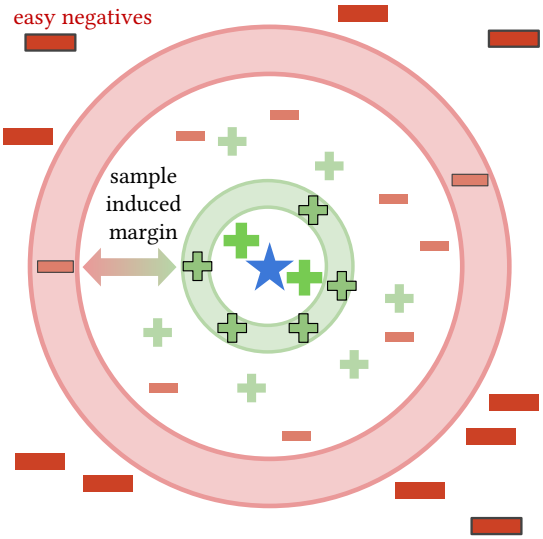


Figure 5.1: Starting from a query paper ★ in a citation graph embedding space. Hard positives ✚ are citation graph embeddings that are sampled from a similar (close) context of ★ but are not so close that their gradients collapse easily. Hard (to classify) negatives ▭ (red band) are close to positives (green band) up to a *sampling induced margin*. Easy negatives ▬ are very dissimilar (distant) from the query paper ★.

5.1.1 Contrastive Learning

Before diving into the methodological details, we introduce essential background knowledge about contrastive learning that was not covered in Chapter 2.

Contrastive learning pulls representations of similar data points (positives) closer together, while representations of dissimilar documents (negatives) are pushed apart. A common contrastive objective is the triplet loss (Schroff et al., 2015) that SPECTER used for scientific document representation learning, as described below. However, as Musgrave et al. (2020) and Rethmeier and Augenstein (2021a) point out, contrastive learning objectives work best when specific requirements are respected:

1. Views of the same data should introduce new information, i.e., the mutual information between views should be minimized (Tian et al., 2020b). We use citation graph embeddings to generate contrast label information that supplement text-based similarity.

2. For training time and sample efficiency, negative samples should be hard to classify but should also not collide with positives (Saunshi et al., 2019).
3. Recent works like Khosla et al. (2020) and Musgrave et al. (2020) use multiple positives. However, positives need to be consistently close to each other (Wang and Isola, 2020), since positives and negatives may otherwise collide, e.g., Cohan et al. (2020) consider only 'citations by the query' as similarity signal and not 'citations to the query'. Such unidirectional similarity does not guarantee that a negative paper (not cited by the query) may cite the query paper and thus could cause collisions the more we sample. Papers cited by the query are positives, and papers not cited by the query are negatives. When papers citing the query are not considered as positives, collisions can occur.

Our method treats citing and being cited as positives (Requirement 2), while it also generates hard negatives and hard positives (Requirement 2+3). Hard negatives are close but do not overlap positives (red band in Figure 5.1). Hard positives are close, but not trivially close to the query document (green band in Figure 5.1). The sample-induced margin (space between the red and green band in Figure 5.1) ensures that contrastive samples do not collide.

Triplet mining. Triplet mining remains a challenge in NLP due to the discrete nature of language, making data augmentation less trivial than computer vision (Gao et al., 2021). Examples of augmentation strategies are translation (Fang et al., 2020), or word deletion and reordering (Wu et al., 2020). Positives and negatives can be sampled based on the sentence position within a document (Giorgi et al., 2021). Gao et al. (2021) utilize supervised entailment datasets for the triplet generation. Language- and text-independent approaches are also applied. Kim et al. (2021) use intermediate BERT hidden state for positive sampling, and Wu et al. (2021) add noise to representations to obtain negative samples. Xiong et al. (2020) present an approach similar to SciNCL where they sample hard negatives from the k nearest neighbors in the embedding space derived from the previous model checkpoint. While Xiong et al. rely only on textual data, SciNCL also integrates citation information which is especially valuable in the scientific context as Cohan et al. (2020) has shown.

Scientific document representations. Scientific document representations based on Transformers (Vaswani et al., 2017) and pretrained on domain-specific text dominate today's scientific document processing. There are SciBERT (Beltagy et al., 2019), BioBERT (Lee et al., 2019) and SciGPT2 (Luu et al., 2021), to name a few. Recent works modify these domain-specific language models to support cite-worthiness detection (Wright and Augenstein, 2021) or fact-checking (Wadden et al., 2020).

Aside from text, citations are a valuable signal for the similarity of research papers. Paper (node) representations can be learned using the citation graph (Grover and Leskovec, 2016; Perozzi et al., 2014; Wu et al., 2019). Especially for recommendations of papers or citations, hybrid combinations of text and citation features are often employed (Brochier et al., 2019; Han et al., 2018; Holm et al., 2022; Jeong et al., 2020; Yang et al., 2015). We discuss in Section 2.1.1 additional research paper and citation recommendation approaches.

Closest to SciNCL are Citeomatic (Bhagavatula et al., 2018) and SPECTER (Cohan et al., 2020). While Citeomatic relies on bag-of-words for its textual features, SPECTER is based on SciBERT. Both leverage citations to learn a triplet-based document embedding model, whereby positive samples are papers cited in the query. Easy negatives are random papers not cited by the query.

Hard negatives are citations of citations – papers referenced in positive citations of the query but are not cited directly by it. Citeomatic also uses a second type of hard negatives, which are the nearest neighbors of a query that are not cited by the query.

Unlike our approach, Citeomatic does not use the neighborhood of citation embeddings but instead relies on the actual document embeddings from the previous epoch. Despite being related to SciNCL, the sampling approaches employed in Citeomatic and SPECTER do not account for the pitfalls of using discrete citations as a signal for paper similarity. The work presented in this chapter addresses the aforementioned issue.

Cross-modal transfer. SciNCL transfers knowledge across modalities, i.e., from citations into a language model. According to Cohan et al. (2020), SciNCL can be considered as a "*citation-informed Transformer*". In the literature, such a cross-modal transfer learning is applied for various modalities (Kaur et al., 2021): text-to-image (Socher et al., 2013), RGB-to-depth image (Tian et al., 2020a), or graph-to-image (Wang et al., 2018). While the aforementioned methods incorporate cross-modal knowledge through joint loss functions or latent representations, SciNCL transfers knowledge through the contrastive sample selection, which we found to be superior to the direct transfer approach; see Appendix in Ostendorff et al. (2022b).

Learning objective. Given the textual content of a document d (a research paper), the goal is to derive a dense vector representation $\boldsymbol{d}$ that best encodes the document information and can be used in downstream tasks. A Transformer language model f (Beltagy et al., 2019, SciBERT) encodes documents d into vector representations $f(d) = \boldsymbol{d}$. The input to the language model is the title and abstract separated by the `[SEP]` token.[2] The final layer hidden state of the `[CLS]` token is then used as a document representation $f(d) = \boldsymbol{d}$.

Training with a masked language modeling objective alone has been shown to produce sub-optimal document representations, as shown by related work (Gao et al., 2021; Li et al., 2020) and by our study on legal recommendations (Chapter 4). Thus, similar to the state-of-the-art method SPECTER (Cohan et al., 2020), we continue training the SciBERT model (Beltagy et al., 2019) using a self-supervised triplet margin loss (Schroff et al., 2015):

$$\mathscr{L} = \max\left\{\|\boldsymbol{d}^Q - \boldsymbol{d}^+\|_2 - \|\boldsymbol{d}^Q - \boldsymbol{d}^-\|_2 + \xi, 0\right\}$$

Here, ξ is a slack term ($\xi = 1$ as in SPECTER) and $\|\Delta \boldsymbol{d}\|_2$ is the L^2 norm, used as a distance function. However, the SPECTER sampling method has significant drawbacks. We will describe these issues and our contrastive learning theory-guided improvements in detail below in Section 5.1.2.

5.1.2 Citation Neighborhood Sampling

Compared to the textual content of a paper, citations provide an outside view of a paper and its relation to the scientific literature (Elkiss et al., 2008), which is why citations are traditionally used as a similarity measure in library science (Kessler, 1963; Small, 1973). Our experiments in Chapter 3 also reveal that users perceive text-based and citation (graph-based) similarity differently. However, using citations as a discrete similarity signal, as done in Cohan et al.

[2] Cohan et al. (2019) evaluated other inputs (venue or author) but found the title and abstract to perform best.

(2020), has its pitfalls. Their method defines papers cited by the query as positives, while papers citing the query could be treated as negatives. This means that *positive and negative learning information collides* between citation directions, which Saunshi et al. (2019) have shown to deteriorate performance. Furthermore, a cited paper can have low similarity with the citing paper given the many motivations a citation can have (Teufel et al., 2006). Likewise, a similar paper might not be cited.

To overcome these limitations, we learn citation embeddings first and then use the citation neighborhood around a given query paper d^Q to construct similar (positive) and dissimilar (negative) samples by using the k nearest neighbors. This builds on the intuition that nodes connected by edges should be close to each other in the embedding space (Perozzi et al., 2014). Using citation embeddings allows us to (1) sample paper similarity on a continuous scale, which makes it possible to (2) define hard-to-learn positives, as well as (3) hard or easy-to-learn negatives. Points (2-3) are essential for efficient contrastive learning, as described below in Section 5.1.2.1.

5.1.2.1 Positives and Negatives Sampling

Positive samples. A positive sample d^+ should be semantically similar to the query paper d^Q, i.e., sampled close to the query embedding $\boldsymbol{d}^Q$. Additionally, as Wang and Isola (2020) find, positives should be sampled from comparable locations (distances from the query) in embedding space and be dissimilar enough from the query embedding to avoid gradient collapse (zero gradients). Therefore, we sample c^+ positive (similar) papers from a close neighborhood around query embedding $\boldsymbol{d}^Q$ $(k^+ - c^+, k^+]$, i.e. the green band in Figure 5.1. When sampling with k nearest neighbors search, we use a small k^+ to find positives and later analyze the impact of k^+ in Figure 5.2.

Negative samples. Negative samples can be divided into easy ▬ and hard ▭ negative samples (light and dark red in Figure 5.1). The sampling of hard negatives is known to improve contrastive learning (Bucher et al., 2016; Wu et al., 2017). However, we make sure to sample hard negatives (red band in Figure 5.1) such that they are close to potential positives but do not collide with positives (green band) by using a tunable 'sampling induced margin'. We do so since Saunshi et al. (2019) showed that the sampling of hard negatives only improves performance *if the negatives do not collide with positive samples* since collisions make the learning signal noisy. That is, in the margin between hard negatives and positives, we expect positives and negatives to collide. Thus we avoid sampling from this region. To generate a diverse self-supervised citation similarity signal for contrastive document representation learning, we also sample easy negatives that are farther from the query than hard negatives. For negatives, the k^- should be large when sampling via k nearest neighbors to ensure samples are dissimilar from the query paper.

5.1.2.2 Citation Graph Embeddings

We train a graph embedding model f_c on citations extracted from the Semantic Scholar Open Research Corpus (S2ORC; Lo et al., 2020) to get citation embeddings C. We utilize PyTorch BigGraph (Lerer et al., 2019), which allows for training on large graphs with modest hardware requirements. The resulting graph embeddings perform well using the default training settings from Lerer et al. (2019), but given more computational resources, careful tuning may produce even better-performing embeddings. Nonetheless, we conducted a narrow parameter search based on link prediction, as reported in Ostendorff et al. (2022b).

5.1.2.3 Sampling Strategies

As described in Section 5.1.2 and 5.1.2.1, our approach aims to improve upon the method by Cohan et al. (2020). Therefore, we reuse their sampling parameters (5 triplets per query paper) and then further optimize our method's hyperparameters. Specifically, we generate the same amount of $(\boldsymbol{d}^Q, \boldsymbol{d}^+, \boldsymbol{d}^-)$ triplets per query paper as SPECTER to train the triplet loss (Cohan et al., 2020). This means we generate c^+=5 positives (as explained in Section 5.1.2.1). We also generate five negatives, comprised of three easy negatives c^-_{easy}=3 and two hard negatives c^-_{hard}=2, as described in Section 5.1.2.1.

Below, we describe three strategies (I-III) for sampling triplets. These either sample neighboring papers from citation embeddings (I), by random sampling (II), or using a combination of both strategies (III). For each strategy, let c' be the number of samples for either positives c^+, easy negatives c^-_{easy}, or hard negatives c^-_{hard}.

(I) k nearest neighbors. Assuming a given citation embedding model f_c and a search index (e.g., FAISS Section 5.1.5), we run $kNN(f_c(d^Q), C)$ and take c' samples from a range of the $(k-c', k]$ nearest neighbors around the query paper d^Q with its neighbors N=$\{n_1, n_2, n_3, \dots\}$, whereby neighbor n_i is the i-th nearest neighbor in the citation embedding space. For instance, for c'=3 and k=10 the corresponding samples would be the three neighbors descending from the tenth neighbor: n_8, n_9, and n_{10}. To reduce computing effort, we sample the neighbors N only once via $[0; \max(k^+, k^-_{\text{hard}})]$, and then generate triplets by range-selection in N; i.e. positives $= (k^+ - c^+; k^+]$, and hard negatives $= (k^-_{\text{hard}} - c^-_{\text{hard}}; k^-_{\text{hard}}]$.

(II) Random sampling. Sample any c' papers without replacement from the corpus.

(III) Filtered random. Like (II) but excluding the papers that are retrieved by k nearest neighbors, i.e., all neighbors within the largest k are excluded.

The k nearest neighbors sampling introduces the hyperparameter k that allows for the *controlled sampling of positives or negatives* with different difficulty (from easy to hard depending on k). Specifically, in Figure 5.1 the hyperparameter k defines the tunable *sample induced margin* between positives and negatives, as well as the width and position of the positive sample band (green) and negative sample band (red) around the query sample.

5.1.3 Datasets

We train and evaluate SciNCL on the following datasets.

5.1.3.1 Evaluation Dataset

We evaluate on the SciDocs benchmark (Cohan et al., 2020). A key difference from other benchmarks is that embeddings are the input to the individual tasks without explicit fine-tuning. The SciDocs benchmark consists of the following four tasks:

- **Document classification** (CLS) with Medical Subject Headings (Lipscomb, 2000) and Microsoft Academic Graph labels (Sinha et al., 2015) evaluated with the F1 metric.

- **Co-views and co-reads** (USR) prediction based on the L2 distance between embeddings. Co-views are papers viewed in a single browsing session. Co-read refers to a user accessing the PDF of a paper. Both user activities are evaluated using Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (nDCG).
- **Direct and co-citation** (CITE) prediction based on the L2 distance between the embeddings. MAP and nDCG are the evaluation metrics.
- **Recommendations** (REC) generation based on embeddings and paper metadata. An offline evaluation with historical clickthrough data determines the performance using Precision@1 (P@1) and nDCG.

The final SciDocs score is computed as average overall metrics and all tasks.

5.1.3.2 Training Datasets

The experiments mainly compare SciNCL against SPECTER on the SciDocs benchmark. However, we found 40.5% of SciDocs's papers leaking into SPECTER's training data. The leakage affects only the unsupervised paper data but not the gold labels – see Appendix in Ostendorff et al. (2022b). To be transparent about this leakage, we train SciNCL on two datasets:

SPECTER replication (w/ leakage). We replicate SPECTER's training data and its leakage. Unfortunately, SPECTER provides neither citation data nor a mapping to S2ORC, on which our citation embeddings are based. We successfully map 96.2% of SPECTER's query papers and 83.3% of the corpus from which positives and negatives are sampled to S2ORC. To account for the missing papers, we randomly sample papers from S2ORC (without the SciDocs papers) such that the absolute number of papers is identical with SPECTER.

S2ORC subset (w/o leakage). We select a random subset from S2ORC that does not contain any of the mapped SciDocs papers. This avoids SPECTER's leakage but also makes the scores reported in Cohan et al. (2020) less comparable. We successfully map 98.6% of the SciDocs papers to S2ORC. Thus, only the remaining 1.4% of the SciDocs papers could leak into this training set in the worst case.

The details of the dataset creation are described in Ostendorff et al. (2022b). Both training sets yield 684K triplets (same count as SPECTER). Also, the ratio of training triplets per query remains the same (Section 5.1.2.3). Our citation embedding model is trained on the S2ORC citation graph. In *w/ leakage*, we include all SPECTER papers even if they are part of SciDocs, the remaining SciDocs papers are excluded (52.5 nodes and 463M edges). In *w/o leakage*, all mapped SciDocs papers are excluded such that we avoid leakage also for the citation embedding model (52.4M nodes and 447M edges).

5.1.4 Evaluated Methods

We compare against the following baselines:

- Randomly initialized embeddings
- Paragraph Vectors / DBOW (Le and Mikolov, 2014)
- Universal Sentence Encoder (Cer et al., 2018, USE)

- SIF / fastText (Arora et al., 2017) - document representations generated by removing the first principal component of aggregated scientific fastText embeddings
- BERT (Devlin et al., 2019) - a state-of-the-art LLM pretrained on general-domain text
- BioBERT (Lee et al., 2019) - a BERT variation for biomedical text
- SciBERT (Beltagy et al., 2019) - a BERT variation for scientific text
- CiteBERT (Wright and Augenstein, 2021) - a SciBERT variation fine-tuned on cite-worthiness detection
- Sentence-BERT (Reimers and Gurevych, 2019) - model that uses negative sampling based on Wikipedia sections to tune BERT for document embeddings
- DeCLUTR (Giorgi et al., 2021) - scientific language with contrastive fine-tuning based on sentence positions
- SGC (Wu et al., 2019) - the graph-convolution approach using the citation information
- Citeomatic (Bhagavatula et al., 2018)
- SPECTER (Cohan et al., 2020)

If not otherwise mentioned, all BERT variations are used in their `base-uncased` versions.

Furthermore, we compare against *Oracle SciDocs* which is identical to SciNCL except that its triplets are generated based on SciDocs's validation and test set using the gold labels. For example, papers with the same MAG labels are positives, and papers with different labels are negatives. Similarly, the ground truth for the other tasks is used, e.g., clicked recommendations are considered as positives. In total, this procedure creates 106K training triplets for *Oracle SciDocs*.[3] Accordingly, *Oracle SciDocs* represents an estimate for the performance upper bound that can be achieved with the current setting (triplet margin loss and SciBERT encoder).

5.1.5 Implementation Details

We replicate the training setup from SPECTER as close as possible. We implement SciNCL using Huggingface Transformers (Wolf et al., 2020), initialize the model with SciBERT's weights (Beltagy et al., 2019), and train via the triplet loss as defined in Section 5.1.1. The optimizer is Adam with weight decay (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) and learning rate $\lambda=2^{-5}$. To explore the effect of computing efficient fine-tuning we also train a BitFit model (Ben Zaken et al., 2022) with $\lambda=1^{-4}$ (Section 5.2.3).

We train SciNCL on two NVIDIA GeForce RTX 6000 (24G) for 2 epochs (approx. 24 hours of training time) with batch size 8 and gradient accumulation for an effective batch size of 32 (same as SPECTER). The graph embedding training is performed on an Intel Xeon Gold 6230 CPU with 60 cores and takes approx. 6 hours. The k nearest neighbors strategy is implemented with FAISS (Johnson et al., 2021) using a flat index (exhaustive search) and takes less than 30min for indexing and retrieval of the triplets.

[3] We under-sample triplets from the classification tasks to ensure a balanced triplet distribution over the tasks.

5.2 Evaluation

This section presents the evaluation of SciNCL and its hyperparameter optimization.

Table 5.1: Results on the SciDocs test set. With replicated SPECTER training data, SciNCL surpasses the previous best avg. score by 1.8 points and outperforms the baselines in 9 of 12 task metrics. Our scores are reported as mean and standard deviation σ over ten random seeds. With training data randomly sampled from S2ORC, SciNCL outperforms SPECTER in terms of avg. score with 1.7 points. The scores with * are from Cohan et al. (2020). *Oracle SciDocs* † is the upper bound of the performance with triplets from SciDocs's data.

Task →	**Classification**		**User activity pred.**				**Citation prediction**				**Recomm.**		**Avg.**
Subtask →	**MAG**	**MeSH**	**Co-View**		**Co-Read**		**Cite**		**Co-Cite**				
Model ↓ / Metric →	**F1**	**F1**	**MAP**	**nDCG**	**MAP**	**nDCG**	**MAP**	**nDCG**	**MAP**	**nDCG**	**nDCG**	**P@1**	
Oracle SciDocs †	*87.1*	*94.8*	*87.2*	*93.5*	*88.7*	*94.6*	*92.3*	*96.8*	*91.4*	*96.4*	*53.8*	*19.4*	*83.0*
Random*	4.8	9.4	25.2	51.6	25.6	51.9	25.1	51.5	24.9	51.4	51.3	16.8	32.5
DBOW* (2014)	66.2	69.2	67.8	82.9	64.9	81.6	65.3	82.2	67.1	83.4	51.7	16.9	66.6
USE (2018)	80.0	83.9	77.2	88.1	76.5	88.1	76.6	89.0	78.3	89.8	53.7	19.6	75.1
SIF (2017)	78.4	81.4	79.4	89.4	78.2	88.9	79.4	90.5	80.8	90.9	53.4	19.5	75.9
Citeomatic* (2018)	67.1	75.7	81.1	90.2	80.5	90.2	86.3	94.1	84.4	92.8	52.5	17.3	76.0
SGC* (2019)	76.8	82.7	77.2	88.0	75.7	87.5	91.6	96.2	84.1	92.5	52.7	18.2	76.9
BERT (2019)	79.9	74.3	59.9	78.3	57.1	76.4	54.3	75.1	57.9	77.3	52.1	18.1	63.4
SciBERT* (2019)	79.7	80.7	50.7	73.1	47.7	71.1	48.3	71.7	49.7	72.6	52.1	17.9	59.6
BioBERT (2019)	77.2	73.0	53.3	74.0	50.6	72.2	45.5	69.0	49.4	71.8	52.0	17.9	58.8
CiteBERT (2021)	78.8	74.8	53.2	73.6	49.9	71.3	45.0	67.9	50.3	72.1	51.6	17.0	58.8
DeCLUTR (2021)	81.2	88.0	63.4	80.6	60.0	78.6	57.2	77.4	62.9	80.9	52.0	17.4	66.6
Sent.-BERT (2019)	80.5	69.1	68.2	83.3	64.8	81.3	63.5	81.6	66.4	82.8	51.6	17.1	67.5
SPECTER* (2020)	**82.0**	86.4	83.6	91.5	84.5	92.4	88.3	94.9	88.1	94.8	**53.9**	**20.0**	80.0
Replicated SPECTER training data (w/ leakage):													
SciNCL (ours)	81.4	**88.7**	**85.3**	**92.3**	**87.5**	**93.9**	**93.6**	**97.3**	**91.6**	**96.4**	**53.9**	19.3	**81.8**
$\pm$ σ w/ ten seeds	*.449*	*.422*	*.128*	*.08*	*.162*	*.118*	*.104*	*.054*	*.099*	*.066*	*.203*	*.356*	*.064*
Random S2ORC training data (w/o leakage):													
SPECTER	81.3	88.4	83.1	91.3	84.0	92.1	86.2	93.9	87.8	94.7	52.2	17.5	79.4
SciNCL (ours)	81.3	89.4	84.3	91.8	85.6	92.8	91.4	96.3	90.1	95.7	54.3	19.9	81.1

5.2.1 Overall Results

Table 5.1 shows the results, comparing SciNCL with the best validation performance against the baselines. With replicated SPECTER training data (w/ leakage), SciNCL achieves an average performance of 81.8 across all metrics, which is a 1.8 point absolute improvement over SPECTER (the next-best baseline). When trained without leakage, the improvement of SciNCL over SPECTER is consistent with 1.7 points but generally lower (79.4 avg. score). In the following, we refer to the results obtained through training on the replicated SPECTER data (w/ leakage) if not otherwise mentioned.

We find the best validation performance based on SPECTER's data when positives and hard negatives are sampled with k nearest neighbors, positives with k^+=25, and hard negatives with

k^{-}_{hard}=4000 (Section 5.2.2). Easy negatives are generated through filtered random sampling. Since random sampling accounts for a large fraction of the triplets (in the form of easy negatives), we report the mean scores and standard deviation based on ten random seeds (seed $\in [0,9]$).

For MAG classification, SPECTER achieves the best result with 82.0 F1 followed by SciNCL with 81.4 F1 (-0.6 points). For MeSH classification, SciNCL yields the highest score with 88.7 F1 (+2.3 compared to SPECTER). Both classification tasks have in common that the chosen training settings lead to over-fitting. Changing the training by using only 1% training data, SciNCL yields 82.2 F1@MAG (Table 5.2). In all user activity and citation tasks, SciNCL yields higher scores than all baselines. Moreover, SciNCL outperforms SGC on direct citation prediction, where SGC outperforms SPECTER in terms of nDCG. On the recommender task, SPECTER yields the best P@1 with 20.0, whereas SciNCL achieves 19.3 P@1 (in terms of nDCG SciNCL and SPECTER are on par). The recommendation task shows the strongest effect of random seeds (σ of 0.3 nDCG and 0.6 P@1). The performance difference between SciNCL and SPECTER is close to or within the standard deviation. Hence, it remains unclear whether the difference is significant since Cohan et al. (2019) did not report standard deviations. In contrast to the classification tasks, training for more than two epochs leads to further improvement on the recommendation task (currently under-fitting). As a result, one should adjust the training settings accordingly when aiming only for this particular task.

When training SPECTER and SciNCL without leakage, SciNCL outperforms SPECTER even in 11 of 12 metrics and is on par in the other metric. This suggests that SciNCL's hyperparameters have a low corpus dependency since they were only optimized on the corpus with leakage.

Regarding the language model baselines, we observe that the general-domain BERT, with a score of 63.4, outperforms the domain-specific BERT variants, namely SciBERT (59.6), BioBERT (58.8), and CiteBERT (58.8) Language models without citations or contrastive objectives yield generally poor results (even compared to Doc2Vec or fastText). This emphasizes the anisotropy problem of embeddings directly extracted from current language models and highlights the advantage of combining text and citation information.

In summary, we show that SciNCL's triplet selection leads on average to a performance improvement on SciDocs, with most gains being observed for user activity and citation tasks. The gain from 80.0 to 81.8 is particularly notable given that even *Oracle SciDocs* yields with 83.0 an only marginally higher avg. score despite using test and validation data from SciDocs for the triplet selection.

5.2.2 Impact of Sample Difficulty

In this section, we present the optimization of SciNCL's sampling strategy (Section 5.1.2.1). We optimize the sampling for positives and hard or easy negatives with a partial grid search on a random sample of 10% of the replicated SPECTER training data (sampling based on queries). Our experiments show that optimizations on this subset correlate with the entire dataset. The validation scores in Figure 5.2 and 5.3 are reported as the mean over three random seeds including standard deviation.

Positive samples. Figure 5.2 shows the avg. scores on the SciDocs validation set depending on the selection of positives with the k nearest neighbors strategy (error bars are the standard deviations over three random seeds). We only modify k^{+}, while negative sampling remains fixed

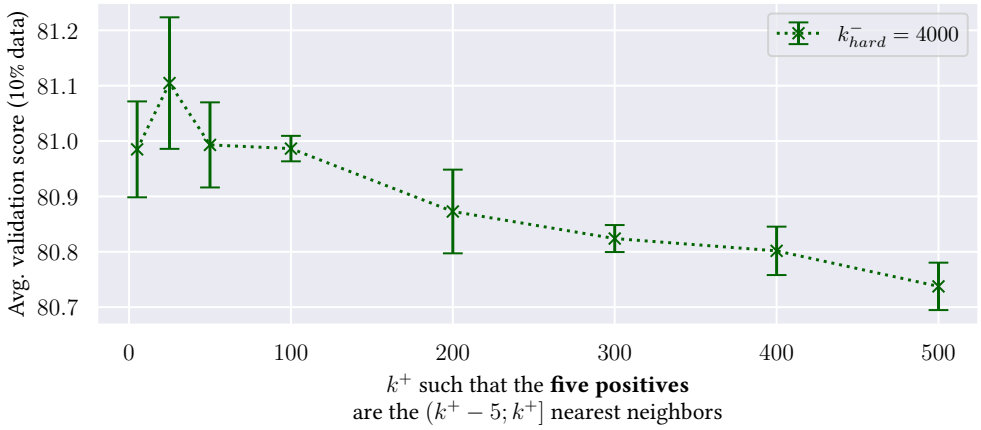


Figure 5.2: Results on the validation set with respect to positive sampling with k nearest neighbors when using 10% training data, hard negative samples are fixed to $k^-_{\text{hard}} = 4000$.

to its best setting (Section 5.2.2). The performance is relatively stable for k^+<100 with the peak at k^+=25, for k^+>100 the performance declines as k^+ increases. Wang and Isola (2020) state that positive samples should be semantically similar to each other, but not too similar to the query. For example, positives with k^+=5 might be "too easy" to learn, such that they produce less informative gradients than the optimal setting k^+=25. Similarly, making k^+ too large leads to the *sampling induced margin* being too small, such that *positives collide with negative samples*, which creates contrastive label noise that degrades performance (Saunshi et al., 2019).

Another observation is the standard deviation σ: One would expect σ to be independent of k^+ since random seeds affect only the negatives. However, positives and negatives interact with each other through the triplet margin loss. Therefore, σ is also affected by k^+. To account for the interaction of positives and negatives, one could sample simultaneously based on the distance to the query and the distance of positives and negatives to each other.

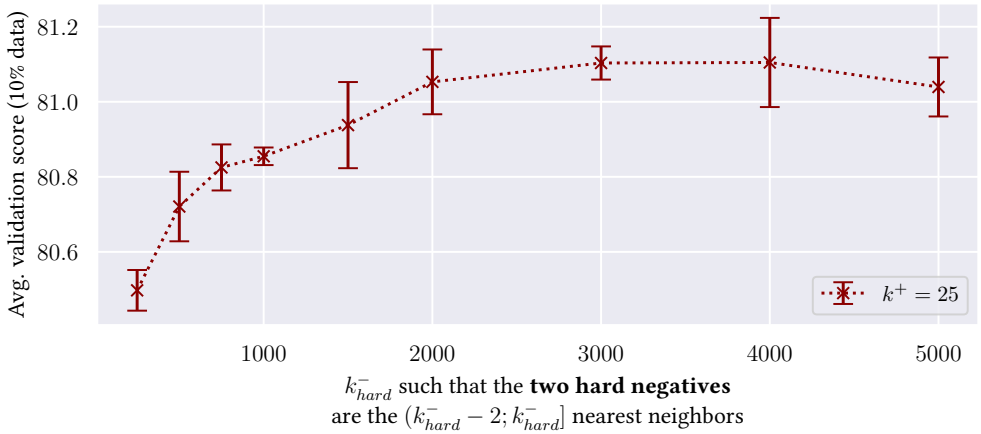


Figure 5.3: Results on the validation set with respect to hard negative sampling with k nearest neighbors using 10% training data, positive samples are fixed to $k^+ = 25$.

Hard negative samples. Figure 5.3 presents the validation results for different k^-_{hard} given the best setting for positives (k^+=25). The performance increases with increasing k^-_{hard} until a plateau between 2000<k^-_{hard}<4000 with a peak at k^-_{hard}=4000. This plateau can also be observed in the test set, where k^-_{hard}=3000 yields a marginally lower score of 81.7 (Table 5.2). For k^-_{hard}>4000, the performance starts to decline again. This suggests that for large k^-_{hard} the samples are not "hard enough" which confirms the findings of Cohan et al. (2020).

Intuitively, the k nearest neighbors strategy should suffer from a centrality or hubness problem. How many neighbors are semantically similar strongly depends on the query paper itself. A popular and frequently cited paper like BERT (Devlin et al., 2019) has many more similar neighbors than a paper about a niche topic like citation recommendation (Jeong et al., 2020). To test this assumption, we also evaluate a strategy with an absolute distance in the embedding space. The absolute distance should account for the hubness problem. However, this strategy underperforms with a score of 81.7 points.

Easy negative samples. The filtered random sampling of easy negatives yields the best validation performance compared to pure random sampling (Table 5.2). However, the performance difference is marginal. When rounded to one decimal, their average test scores are identical. The marginal difference is caused by the large corpus size and the resulting small probability of randomly sampling one paper from the k nearest neighbors results. But without filtering, the effect of random seeds increases, since we find a higher standard deviation compared to the one with filtering. As a potential way to decrease randomness, we experimented with other approaches like k means clustering but found that they decrease the performance.

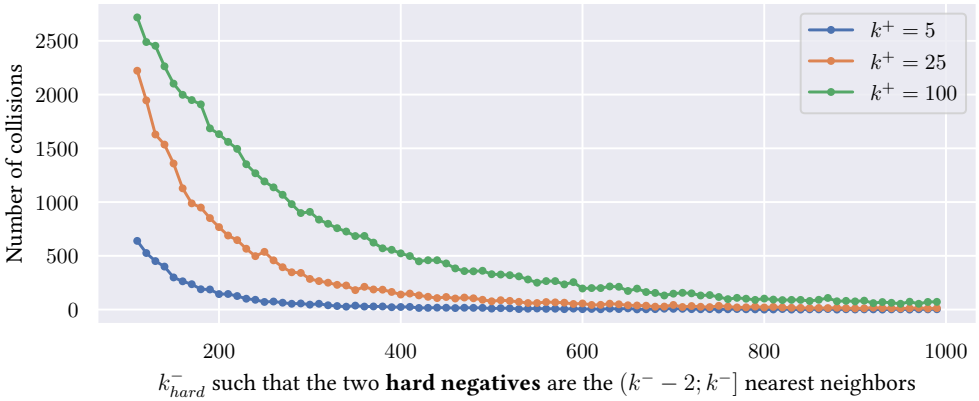
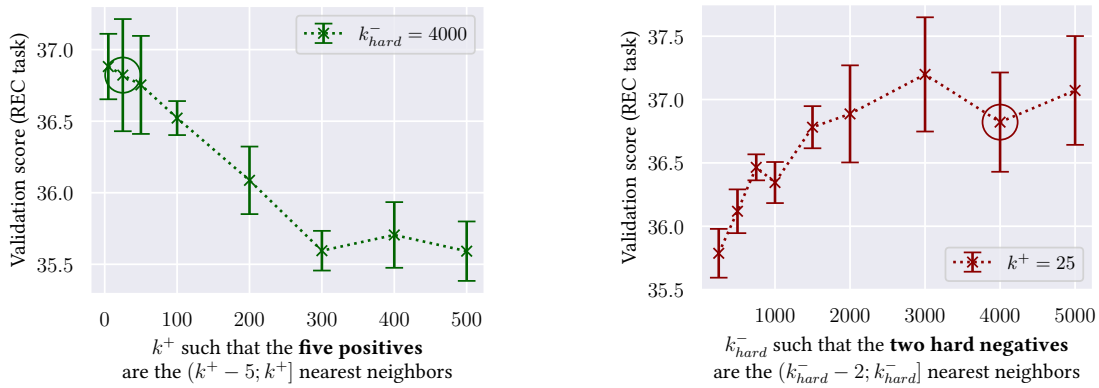


Figure 5.4: Number of collisions with respect to the size of the sample induced margin as defined through k^+ and k^-_{hard}. As the margin increases the collisions get less likely.

Collisions. Similar to SPECTER, SciNCL's sampling based on graph embeddings could cause collisions when selecting positives and negatives from regions close to each other. To avoid this, we rely on a sample-induced margin that is defined by the hyperparameters k^+ and k^-_{hard} (distance between the red and green band in Figure 5.1). When the margin gets too small, positives and negatives are more likely to collide. A collision occurs when the paper pair (d_q, d_s) is contained in the training data as a positive and as a negative sample at the same time.

Figure 5.4 demonstrates the relation between the number of collisions and the size of the sample-induced margin. The number of collisions increases when the sample-induced margin gets smaller. The opposite is the case when the margin is large enough ($k^-_{\text{hard}} > 1000$), i.e., then the number of collisions goes to zero. This relation also affects the evaluation performance as Figure 5.2 and 5.3 show. Namely, for large k^+ or small k^-_{hard} SciNCL's performance declines and approaches SPECTER's performance.



(a) Positives

(b) Hard negatives

Figure 5.5: Validation performance for the recommendation task w.r.t. k^+ and k^-_{hard} with k nearest neighbors strategy using 10% data. The hyperparameters are task-specific. The selected k values are suboptimal for the recommendation task (circled values).

Task-specific results. Figure 5.5 presents the validation performance for the recommendation task and not as an average over all tasks. The plots show that the optimal k^+ and k^-_{hard} values are task-specific. This means that one could optimize the SciNCL representations for a specific downstream task. For example, the recommendation task performance could be improved by selecting $k^+ = 5$ and $k^-_{\text{hard}} = 3000$ as the hyperparameters. The optimal hyperparameters for the recommendation task are different from the ones for the average over all SciDocs tasks.

5.2.3 Ablation Analysis

In addition to sample difficulty, we also evaluate the performance impact of data quantity, trainable parameters, and language model initialization.

Initial language models. Table 5.2 shows the effect of initializing the model weights not with SciBERT but with general-domain language models (BERT-Base and BERT-Large) or with BioBERT. The initialization with other pretrained language models decreases the performance. However, the decline is marginal (BERT-Base -0.6, BERT-Large -0.4, BioBERT -0.4) and all other initializations outperform the SPECTER baseline. For the recommendation task, in which SPECTER is superior over SciNCL, BioBERT outperforms SPECTER. This indicates that the improved triplet mining of SciNCL has a greater domain adaption effect than pretraining on domain-specific literature. Given that model pretraining requires a magnitude more resources than fine-tuning with SciNCL, our approach can be a solution for resource-limited use cases.

Table 5.2: Ablations. Numbers are averages over tasks of the SciDocs test set, average score over all metrics, and rounded absolute difference to SciNCL.

Ablations ↓	**CLS**	**USR**	**CITE**	**REC**	**Avg.**	**Δ**
SciNCL	85.0	88.8	**94.7**	36.6	**81.8**	–
SPECTER	84.2	88.4	91.5	36.9	80.0	-1.8
k^{-}_{hard}=2000	84.9	88.8	**94.7**	36.1	81.6	-0.2
k^{-}_{hard}=3000	84.5	88.7	94.6	36.9	81.7	-0.1
easy neg. w/ random	85.1	88.8	**94.7**	36.6	**81.8**	0.0
undirected citations	84.6	88.8	**94.7**	36.6	81.7	-0.1
Init. w/ BERT-Base	83.4	88.4	93.8	37.5	81.2	-0.6
Init. w/ BERT-Large	84.6	88.7	94.1	36.4	81.4	-0.4
Init. w/ BioBERT	83.7	88.6	93.8	**37.7**	81.4	-0.4
1% training data	85.2	88.3	92.7	36.1	80.8	-1.0
10% training data	85.1	88.7	93.5	36.2	81.1	-0.6
BitFit training	**85.8**	88.6	93.7	35.3	81.2	-0.5

Data and computing efficiency. The last three rows of Table 5.2 show the results regarding data and computing efficiency. When keeping the citation graph unchanged but training the language model with only 10% of the original triplets, SciNCL still yields a score of 81.1 (-0.6). Even with only 1% (6840 triplets), SciNCL achieves a score of 80.8 which is 1.0 points less than with 100% but still 0.8 points more than the SPECTER baseline. With this *textual* sample efficiency, one could manually create triplets or use existing supervised datasets as demonstrated in Gao et al. (2021).

Lastly, we evaluate BitFit training (Ben Zaken et al., 2022), which only trains the bias terms of the model while freezing all other parameters. This corresponds to training only 0.1% of the original model parameters. With BitFit, SciNCL yields a considerable score of 81.2 (-0.5 points). As a result, SciNCL could be trained on the same hardware with even larger (general-domain) language models.

5.3 Discussion

Our experiments show that SciNCL achieves new state-of-the-art results on the SciDocs benchmark. Specifically, SciNCL outperforms the next-best baseline SPECTER with a 1.8 point absolute improvement (+1.7 points when using the w/o leakage dataset).

The improvements can be exclusively attributed to SciNCL's contrastive sample generation since all other settings remain unchanged, i.e., SciNCL uses the same model architecture, pretrained weights, and training settings as SPECTER. SciNCL's sample generation is based on contrastive learning theory insights provided by Musgrave et al. (2020) and Rethmeier and Augenstein (2021a). Using the embedding space of a second modality, i.e., citations, introduces new infor-

mation that complements the text information. The aggregation of the whole citation graph into the citation embeddings makes the similarity signal less noisy compared to discrete citations.

As opposed to prior work, our neighborhood contrastive learning approach does not require handcrafted rules for the sample generation, like SPECTER's citations-of-citations strategy for hard negatives. Instead, the hyperparameters k^+ and k^-_{hard} allow the optimization of the model without any explicit domain knowledge. This will become an even greater advantage in domains or modalities where it is less trivial to derive a continuous similarity signal such as images or other high-dimensional data.

With the hyperparameters, the document encoder model can be optimized for arbitrary goal metrics. In this chapter, we optimized SciNCL for avg. validation score of the SciDocs benchmark that includes a diverse set of downstream tasks. So the resulting document representations are not tailored to any specific task but rather generic ones. On one hand, this optimization led to suboptimal results on the recommendation subtask due to the selected hyperparameters. On the other hand, this improved especially the results for user activity prediction and citation prediction tasks since they account for the major of task metrics from SciDocs (8 of 12).

Overall, SciNCL's approach invests additional (computational) resources to carefully select to most informative positive and negative samples for contrastive learning. While the investment is reasonable (approx. 20% of total training time), the training gets more sample-efficient. Already 1% of training triplets is sufficient to outperform SPECTER. This efficiency will be even more valuable as the language model sizes are growing. Thus, one could train even larger language models with SciNCL with a little increase in computational costs. Interestingly, even starting with a general-domain language model like BERT is sufficient to outperform SPECTER. This indicates that the improved triplet mining of SciNCL has a greater domain adaption effect than pretraining on domain-specific literature. The sample efficiency is achieved by selecting the positive and negative samples such that they are hard to learn without causing collisions.

5.4 Summary of the Chapter

Document similarity measures can be only as good as the methods that encode the underlying document semantics into vector representations. From the previous experiments in Chapter 3 and 4, we have learned that text and graph information complement each other and that they should be combined to achieve optimal results. From the review of related work, especially from classical methods like co-citations (Section 2.4.3), we also know that citations are not a discrete signal for semantic similarity making the generation of contrastive samples based on them suboptimal. Motivated by these findings, this chapter worked on Research Task **II** and improved text-based document representations by utilizing graph information.

This chapter presented the SciNCL approach for contrastive learning of scientific document embeddings with a focus on the challenge of selecting informative positive and negative samples. By leveraging citation graph embeddings for sample generation, SciNCL achieved a score of 81.8 on the SciDocs benchmark, a 1.8 point improvement over the previous best method SPECTER. This was purely achieved by introducing tunable sample difficulty and avoiding collisions between positive and negative samples while the existing language model and data setup can be reused. This improvement over SPECTER can be also observed when excluding the SciDocs papers during training (see w/o leakage in Table 5.1). It is remarkable that the improvement was

consistent even though SciNCL hyperparameters are not optimized for this training corpus indicating a low hyperparameter sensitivity of SciNCL. Furthermore, SciNCL's improvement from 80.0 to 81.8 was particularly notable given that even *oracle triplets*, which are generated with SciDocs's test and validation data, yielded with 83.0 only a marginally higher score (see *Oracle SciDocs* in Table 5.1).

This chapter emphasized the importance of sample generation in a contrastive learning setting. We showed that language model training with 1% of triplets was already sufficient to outperform SPECTER, whereas the remaining 99% provided only 1.0 additional points (80.8 to 81.8). This sample efficiency was achieved by adding reasonable effort for sample generation, i.e., graph embedding training and k nearest neighbors search. We also demonstrated that in-domain language model pretraining (like SciBERT) was beneficial, while general-domain language models could achieve comparable performance and even outperform SPECTER. This indicates that controlling sample difficulty and avoiding collisions is more effective than in-domain pretraining, especially in scenarios where training a language model from scratch is infeasible.

Lastly, this chapter concludes the *aspect-free* part of the thesis, in which we have investigated various document similarity measures that aim for a general similarity without accounting for the aspects that documents may share. As SciNCL is a general document representation method, it also does not directly account for aspect information. However, SciNCL can be used as a base model for aspect-based similarity methods as we investigate in Chapter 7 and 8.

Part III

Aspect-based Document Similarity

Chapter 6

Pairwise Classification for Wikipedia Articles

In Chapters 3-5, we have explored various content-based recommendation approaches that determine the semantic similarity of documents through the textual content, citation or link graphs and hybrid combinations of text and graph information. Our experiments have shown quantitatively and qualitatively that the aspect-free document similarity measures implicitly address different aspects of the document content. This chapter revisits the research subject of Wikipedia articles from Chapter 3 but incorporates aspect information into the similarity assessment as defined by Research Task **III**. The chapter proposes a pairwise multi-class document classification approach, which classifies the explicit aspects that make Wikipedia articles alike. The content of this chapter is based on Ostendorff et al. (2020c).

"Pairwise Multi-Class Document Classification for Semantic Relations between Wikipedia Articles" by **Malte Ostendorff**, Terry Ruas, Moritz Schubotz, Georg Rehm, and Bela Gipp. In: *Proceedings of the 2020 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2020.

The common approach of recommending semantically similar documents based on a similarity measure is a simplification that neglects the many aspects of extensive documents typically found in digital libraries. It remains unclear to which of the many aspects the similarity relates. In philosophy (Goodman, 1972) and in natural language processing (Bär et al., 2011), the similarity of A to B has been criticized as an ill-defined notion unless one can say to what the similarity relates. For content-based literature recommendations, one would rather know what aspects of the two documents are similar or how they relate to each other than just knowing that the documents are similar or dissimilar. Identifying the aspects connecting different documents would allow users to explore the document space by formulating complex queries in terms of documents and their aspect-based similarity (e.g., find a document similar in aspect a_1, but different in aspect a_2). These queries are generally referred to as analogical queries (Gick and Holyoak, 1983). Especially for complex information needs, formulating analogical queries is more intuitive (Lofi and Tintarev, 2017).

Nonetheless, today's document similarity measures do not consider the aspect information that would underpin such a system. While other NLP tasks, like aspect-based sentiment classification (Section 2.5.2), deal with aspects, they are not concerned with aspects in the context of document similarity. Likewise, the document classification task aims to categorize individual documents but fails to address the relationship that binds two or more documents.

In this chapter, we combine the ideas of aspects, document classification, and document similarity to classify the aspect-based similarity of document pairs. Given a seed document d_s, we are interested in finding a target document d_t that shares the aspect a_i with d_s. We use the term "aspect" to indicate a semantic connection between two documents above the syntax level (Khoo and Na, 2007). We formulate the task of determining the aspect a in which a document pair (d_s, d_t) is similar as a pairwise multi-class document classification problem. The classifier has a

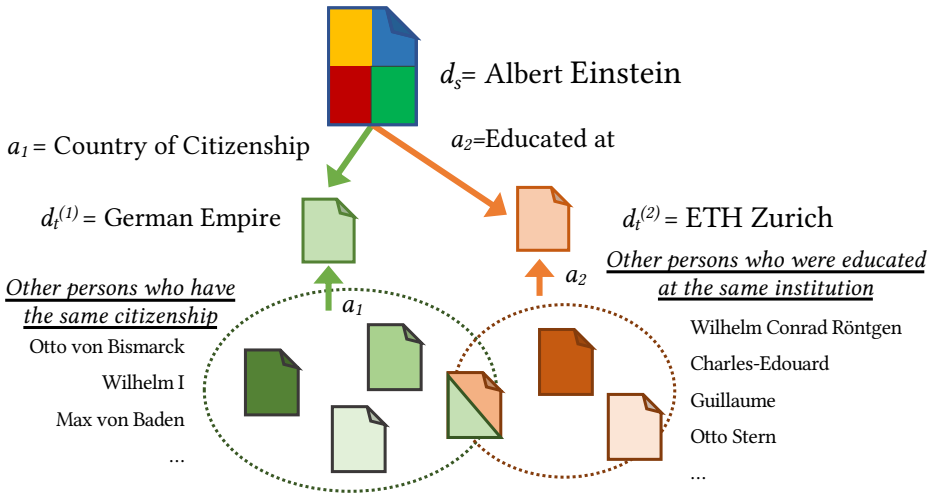


Figure 6.1: Shared aspects between Wikipedia articles. Seed article *Albert Einstein* is connected to other articles by the aspects a_1 and a_2 that are the two Wikidata property *educated at* and *citizenship*. Considering articles only a single edge apart leads to diverse recommendations, while two edges can be used for recommendations focused on a specific or an intersection of aspects.

document pair as input and predicts the corresponding aspect-based similarity. Accordingly, the research questions of this chapter are as follows:

Research questions

RQ1: What methods can measure the aspect-based similarity of Wikipedia articles?

RQ2: If one method performs significantly better than other methods, what methodological differences cause the performance difference?

RQ3: Is there a measurable difference between the aspect classes?

To answer these research questions, we evaluate a diverse set of methods for aspect-based similarity. For the experiments, we build a dataset using Wikipedia and Wikidata (Vrandecic and Krötzsch, 2014) that is suitable to compare the methods. Wikipedia articles are the seed and target documents, while Wikidata properties provide the aspect information that describes what a document pair has in common. Figure 6.1 shows one example from our dataset. The articles *Albert Einstein* and *German Empire* are the pair $(d_s, d_t^{(1)})$ and the aspect similarity is defined by a_1, which is the Wikidata property *country of citizenship*. The *country of citizenship* aspect could then be used to tailor Wikipedia article recommendations to users who are interest in persons with a similar citizenship.

This chapter makes the following three main contributions:

1. We formulate the problem of aspect-based document similarity as a pairwise multi-class document classification task.
2. We implement six different models using word-based document embeddings from GloVe (Pennington et al., 2014) and Paragraph Vectors (Le and Mikolov, 2014), and Transformer language models from BERT (Devlin et al., 2019) and XLNet (Yang et al.,

2019) in vanilla and Siamese architecture (Bromley et al., 1993). Each system is evaluated under specific configurations regarding its concatenation method and sequence length.

3. We introduce a novel dataset composed of 32,168 Wikipedia article pairs and Wikidata properties that define the aspect-based similarity of these articles.

All our datasets, trained models, and source code are publicly available to contribute to transparency and reproducibility.[1]

The remainder of this chapter is structured as follows: First, we introduce the general methodology, i.e., the datasets and the evaluated methods. Subsequently, we present the overall results in Section 6.2.1, the analysis of sequence length, concatenation, and aspects in Sections 6.2.2 to 6.2.4, and results of the manual sample analysis in Section 6.2.5. In Section 6.3, we discuss the results of all evaluations. Finally, we summarize the main findings of this chapter.

6.1 Methodology

This section describes the dataset and implementation to facilitate the reproduction of our results.

6.1.1 Dataset

Existing datasets provide either class annotations of single documents, e.g., topic (Ostendorff et al., 2019), relations between sentences or entities as in natural language inference (Wang et al., 2019), word analogies (Mikolov et al., 2013b), entity relation extraction (Yao et al., 2019), or similarity between text pairs formulated as binary classification (Dolan and Brockett, 2005). Our task is defined as a multi-class classification of document pairs consisting of multiple sentences.

6.1.1.1 Training Dataset

One example of a digital library that employs a content-based recommender system is Wikipedia. Recommendations for Wikipedia articles have been addressed in our experiments from Chapter 3 and the literature (Ollivier and Senellart, 2007). Wikipedia is closely connected with Wikidata. Wikidata is an open knowledge graph in which nodes represent items (e.g., Wikipedia articles) and edges represent properties of these items (e.g., aspects that connect two different articles). The fact that Wikipedia articles are linked to their corresponding Wikidata items allows the construction of a large dataset tailored to the problem of aspect-based similarity. The triple (d_s, d_t, a_i) of two documents d_s and d_t, and the aspect a_i describes an aspect-based document similarity. In the Resource Description Framework (RDF) terminology, d_s is the subject, d_t the object, and a_i the predicate, whereas in the Wikidata terminology, an aspect corresponds to a statement[2]. The aspect a_i (predicate) is a Wikidata property that semantically relates a pair of Wikipedia articles (d_s, d_t). For instance, the Wikipedia article of *Albert Einstein*[3] and its Wikidata item[4] is connected to the article[5] and item[6] of the *German Empire* through the property *country*

[1] `https://github.com/malteos/semantic-document-relations`, last accessed: 18/01/2023
[2] `https://www.wikidata.org/wiki/Help:Statements`, last accessed: 18/01/2023
[3] `https://en.wikipedia.org/wiki/Albert_Einstein`, last accessed: 18/01/2023
[4] `https://www.wikidata.org/wiki/Q937`, last accessed: 18/01/2023
[5] `https://en.wikipedia.org/wiki/German_Empire`, last accessed: 18/01/2023
[6] `https://www.wikidata.org/wiki/Q43287`, last accessed: 18/01/2023

of citizenship[7]. The Wikidata property acts as both the shared aspect of the Wikipedia article pair and the class label in the training data for this same pair of documents. Table 6.1 lists other examples to better illustrate this approach.

Given Wikipedia's nature as an encyclopedia, its use as a dataset has some shortcomings. Encyclopedic documents tend to describe a single entity, and their semantics can be seen as relatively homogeneous in comparison to other types of literature. Nonetheless, we consider Wikipedia and Wikidata to be suitable corpora to demonstrate our approach. Wikidata properties range from entity-specific aspects (e.g., *educated at*) to abstract ones (e.g., *facet of*). Wikipedia articles and their shared aspects are usually more comprehensible than those in the scientific literature (Chapter 7) , which contributes to the analysis of our results. Another fact that supports our choice of Wikipedia and Wikidata is their open license copyright.

6.1.1.2 Aspect Classes

At the time of writing, Wikidata contained 7,091 properties[8] of which we selected the following nine as aspect classes for this research:

- *country of citizenship* - seed is citizen of the target;
- *different from* - item that is different from another item, with which it is often confused;
- *educated at* - educational institution attended by seed;
- *employer* - seed works or worked for target;
- *facet of* - topic of which this item is an aspect, item that offers a broader perspective on the same topic;
- *has effect* - the seed causes the target;
- *has quality* - the entity has an inherent or distinguishing non-material characteristic;
- *opposite of* - item that is the opposite of this item;
- *symptoms* - possible symptoms of a medical condition.

Table 6.1 lists the corresponding Wikidata PIDs, their quantity, and examples for each property. Besides the number of available Wikipedia article pairs, diversity was also a criterion in our selection. Diversity refers to the different semantic meanings of properties (e.g., *country of citizenship*, *opposite of*). Similarly, the requirements to predict an aspect similarity between documents can also be diverse.

While some aspect classes are clearly expressed within the document text (e.g., for documents referencing people, their citizenship is often put in the first sentences), others will require a more comprehensive understanding of the article content. For instance, while *floor* being the opposite of *ceiling* is evident, this fact will most likely not be explicitly mentioned in the article text. Also, other aspects like *has effect* or *symptoms* can require unwritten domain knowledge.

The classification performance can also be affected by the type of the connected articles. For example, the aspect class *country of citizenship* exclusively connects persons and countries. No

[7] `https://www.wikidata.org/wiki/Property:P27`, last accessed: 18/01/2023
[8] `https://tools.wmflabs.org/hay/propbrowse/`, last accessed: 18/01/2023

other property uses such a combination. On the contrary, the aspect classes *educated at* and *employer*, connect a person with an organization. Additionally, all aspect classes are unidirectional, except for *opposite of*. Given the many semantic relations that the selected aspect classes represent, we expect significant differences in the classification performance.

Table 6.1: The aspect classes with their Wikidata PIDs, three examples, and the number of samples per aspect in our dataset. In total the dataset contains 16,084 samples.

Aspect class	PID	#	Example articles
country of citizenship	P27	3636	Torben Ulrich → Denmark, Neal Doughty → United States, Julian Kenny → Trinidad and Tobago
different from	P1889	4048	Computer file → File folder, Lee County, Alabama → Lee County, Illinois, Karo → Karo (name)
educated at	P69	1798	Hillar Eller → University of Tartu, Al Young → University of Michigan, Heinrich Finkelstein → Leipzig University
employer	P108	1557	Gary M. Mavko → Stanford University, Alexander Medvedev → Gazprom, John Reif → Duke University
facet of	P1269	1343	Reformation → Protestantism, 1974 in Portugal → Portugal, Sportsmanship → Sport
has effect	P1542	698	Language attrition → Extinct language, Arsenic poisoning → Lung cancer, Foul ball → Out (baseball)
has quality	P1552	1022	Antisemitism → Nazism, Employment → Access badge, Human → Gender
opposite of	P461	929	Floor → Ceiling, Person → Society, Exponentiation → Logarithm
symptoms	P780	1053	Myalgia → Influenza, Mercury poisoning → Cough, Death rattle → Sound

6.1.1.3 Data Preprocessing

We sample 10,000 article pairs in total with a balanced class distribution over the nine aspect classes. The aspect information was obtained through the Wikidata SPARQL interface in December 2019. For each Wikipedia article in the sample, we also check whether the article is connected to any other article but is not part of the initial sample and retrieve the missing documents-aspect triplets. We remove all duplicated article pairs and multi-label aspect classes.

The main goal of this chapter is to explore the multi-class classification problem, so we ensure that the same pair of documents did not share different labels. Wikidata provides data for multi-label aspects, especially for hierarchical properties. However, only less than 1% of our sample data contain multi-label aspects. For the sake of simplicity, we decided to remove them (Chapter 7 also considers multi-label aspects).

The preprocessing procedure generates 16,084 Wikipedia article pairs with an imbalanced class distribution (Table 6.1). The increase in samples is due to the retrieval of missing documents-aspect triplets. The corresponding articles are converted to plain text from the English Wikipedia dump of November 2019 using the Gensim API (Rehurek and Sojka, 2010).[9]

[9] `https://radimrehurek.com/gensim/scripts/segment_wiki.html`, last accessed: 18/01/2023

6.1.1.4 Negative Sampling

In addition to the nine positive aspect classes from Wikidata, we introduce a class named *None* that works as a negative class and generates negative samples in the same proportion as the positive samples. The articles in the *None* category are randomly selected and do not share any aspect with the positive ones, i.e., the articles are dissimilar. A more elaborated sampling strategy similar to the one from Chapter 5 was omitted for simplicity. The resulting final dataset contains 32,168 samples in total.

6.1.2 Evaluated Methods

This chapter evaluates six classifiers under different configurations, 30 methods in total. Each classifier takes two documents d_s and d_t as input and predicts the probability for d_s and d_t being similar in aspect a_i. The hyperparameters for the considered systems are described at the end of this section.

We distinguish between three document encoding strategies: (i) document embeddings from word embeddings using the full document text (GloVe and Paragraph Vectors), (ii) Vanilla Transformers, and (iii) Siamese Transformers (each Transformer as BERT and XLNet).

Word-based methods. We use two word vector based methods that utilize the full article text since they are not bound to any input token limit:

- Paragraph Vectors (Le and Mikolov, 2014): We obtained a document vector $\boldsymbol{d} \in \mathbb{R}^{200}$ for each Wikipedia article using the DBOW model and the default settings in Gensim (Section 2.3.5).[10]
- GloVe (Pennington et al., 2014): We use the $\boldsymbol{w} \in \mathbb{R}^{200}$ pretrained word embeddings[11] and compute a Wikipedia article embedding $\boldsymbol{d}$ as the weighted average over its word vectors $\boldsymbol{w_i}$, whereby the number of occurrences of the word i in d defines the weight c_i.

For both methods, we encode each document from our document pair (d_s, d_t) independent of the classification task and concatenate their resulting vectors. The different concatenation variants tested in our experiments are discussed below. The resulting document pair vector is then used as an input to a fully-connected multilayer perceptron (MLP), which predicts the aspect-based similarity for the document pair. The MLP consists of two layers with 512 units and ReLU activation. These hyperparameters are obtained through a grid search. The dimension of the output of the last layer of all classifiers corresponds to the nine Wikidata properties (Table 6.1) and one additional dimension for the *None* class of negative samples (Section 6.1.1.4). The logistic sigmoid function generates the probabilities for the multi-class classification.

Vanilla Transformers. We employ two language models based on the Transformer architecture (Vaswani et al., 2017), specifically BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019); see Section 2.3.7.1 for more details. The two Transformer models support two text segments as their input. BERT is even pretrained to solve a sequence pair classification (next sentence prediction). The content of the document pair (i.e., title and text of d_s and d_t) is tokenized, delimited with special tokens, i.e., `[CLS]` and `[SEP]` for BERT, `<cls>` and `<sep>` for XLNet,

[10] `https://radimrehurek.com/gensim/models/doc2vec.html`, last accessed: 18/01/2023
[11] `https://nlp.stanford.edu/projects/glove/`, last accessed: 18/01/2023

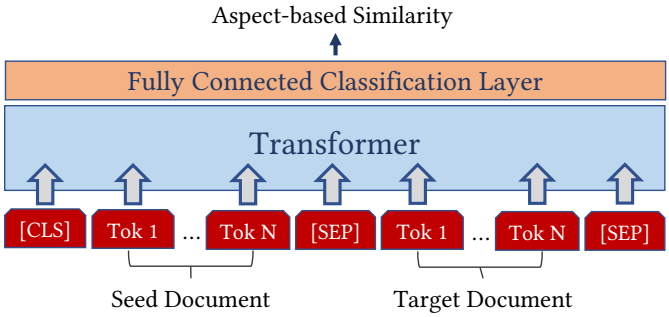


Figure 6.2: Vanilla Transformer for sequence pair classification. [SEP]-token separates seed and target document. The classification head predicts the aspect-based document similarity.

and then jointly fed trough the Transformer (Figure 6.2). The Transformer output is used as the input to a single fully-connected linear layer with 512 units for the classification (prediction head). Regarding terminology, we refer to the two models as vanilla Transformers since their original architecture remains unchanged.

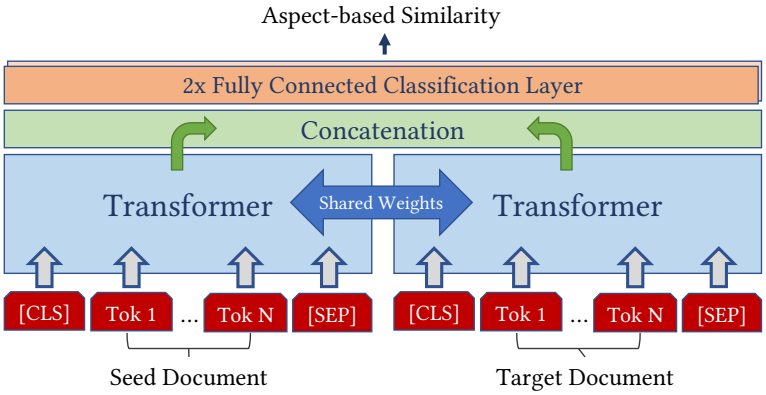


Figure 6.3: Siamese Transformer architecture. Both documents are fed separately through the Transformer, the concatenated document vectors are input to the classification layer.

Siamese Transformer. We combine the two Transformers (BERT and XLNet) in a Siamese network architecture (Bromley et al., 1993). In Siamese networks, two inputs are fed through identical sub-networks with shared weights (in this case, the Transformers), and then passed to a classifier or a similarity function. Reimers and Gurevych (2019) have shown that Siamese BERT networks are suitable for text similarity tasks.

For our experiment, the documents d_s and d_t are individually fed into the Transformer sub-networks to derive two independent contextual document vectors (Figure 6.3). Next, the document vectors are concatenated and classified with a 2-layer MLP (2x512 units with ReLU activation), which is the same method applied by GloVe and Paragraph Vectors. In contrast to Paragraph Vectors and GloVe, the document representations are neither fixed nor frozen but continually learned during the classifier's training. In contrast to Reimers and Gurevych (2019),

our implemented Siamese architecture is applied to a multi-class classification instead of a binary classification.

The architectures of the underlying BERT and XLNet models are the corresponding `BASE-CASED` versions of the pretrained models with 12 layers, 768 hidden size, 12 heads, and 110M parameters. Even though the architectures of BERT and XLNet are comparable, the associated language models are pretrained with different data. While BERT is trained on English Wikipedia and the BooksCorpus (Zhu et al., 2015) alone, XLNet uses additional Web corpora for pretraining (Yang et al., 2019).

Sequence length. The vanilla and Siamese Transformer models based on BERT have a maximum sequence length of 512 tokens due to absolute positional embeddings. However, XLNet integrates the relative positional encoding, as proposed in Transformer-XL (Dai et al., 2019). Therefore, XLNet's architecture is, in theory, not bound to a maximum sequence length. However, custom pretraining is out of scope for this research, and the publicly available pretrained models of XLNet have the same limit of 512 tokens as BERT. It remains unknown how the length of the processed sequence affects the classification task. From Chapter 3, we know that the performance of similarity measures peaks at 450 words since the introduction section in Wikipedia articles presumably contains all essential information. Other sections might add only noise and make it harder to encode relevant semantic information from the articles. Thus, we evaluate the Transformers using 128, 256, and 512 tokens (Section 6.2.2).

Concatenation. Paragraph Vectors, GloVe, and the Siamese models concatenate the separately encoded document vectors $\boldsymbol{d_s}$ and $\boldsymbol{d_t}$. In the literature, there is no widely accepted concatenation method. For instance, Conneau et al. (2017) use $[u; v; |u-v|; u*v]$ for sentence embedding, while Reimers and Gurevych (2019) find $[u; v; |u-v|]$ as the best method. In Section 6.2.3, we test the following variations:

- $[u; v]$ Concatenation of the two vectors u and v;
- $[u; v; |u-v|]$ and absolute value of element-wise difference;
- $[u; v; |u-v|; u*v]$ and element-wise product.

6.1.3 Implementation Details

All experiments with Paragraph Vectors and GloVe can be run on the CPU in less than 15 minutes using the Gensim (Rehurek and Sojka, 2010) and Scikit-learn (Pedregosa et al., 2011) framework. Before training the Paragraph Vectors model, Gensim preprocesses[12] the plain-text from the Wikipedia articles. For GloVe, the individual words occurring in the article text are extracted with Scikit-learn's CountVectorizer[13] including English stop word removal. The Transformer models require a GPU as hardware. We rely on HuggingFace's PyTorch implementation (Wolf et al., 2020) of BERT and XLNet. The training time for a single epoch on a GeForce GTX 1080 Ti (11 GB) ranged from less than 10 minutes for vanilla BERT-128 (simplest Transformer architecture) to 55 minutes for Siamese XLNet-512 (most complex Transformer architecture).

[12] `https://radimrehurek.com/gensim/utils.html#gensim.utils.simple_preprocess`, last accessed: 18/01/2023

[13] `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html`, last accessed: 18/01/2023

As suggested in Devlin et al. (2019), the Transformer training is performed with batch size $b = 4$, dropout probability $d = 0.1$, learning rate $\eta = 2^{-4}$ (Adam optimizer) and 4 training epochs. If not otherwise stated, the default settings of the frameworks were used.

The evaluation is conducted as stratified k-fold cross-validation with $k = 4$ and 24,126 training, and 8,041 test samples (the class distribution remains identical for each fold). The source code, dataset, and trained models are publicly available on Zenodo[14], GitHub[15] and as a demo on Google Colab[16].

6.2 Evaluation

Our results are divided into overall results (Section 6.2.1), sequence length (Section 6.2.2), concatenation (Section 6.2.3), aspect classes (Section 6.2.4), and manual sample examination (Section 6.2.5). These five subsections move from a high-level perspective to a detailed investigation of the results that most contributed to our findings.

Table 6.2: Results as micro avg. F1 score with standard deviation in 4-fold cross-validation for the best configurations, i.e., full-text embeddings from GloVe and Paragraph Vectors and vanilla and Siamese Transformers (BERT-base and XLNet-base). Vanilla BERT-512 performs best.

Model	Seq.	Concatenation	F1	Std.
GloVe	-	$u; v; \lvert u - v \rvert; u * v$	0.875	$\pm$ 0.0036
Paragraph Vectors	-	$u; v; \lvert u - v \rvert; u * v$	0.845	$\pm$ 0.0019
Siamese BERT	512	$u; v; \lvert u - v \rvert; u * v$	0.870	$\pm$ 0.0067
Siamese XLNet	256	$u; v; \lvert u - v \rvert; u * v$	0.870	$\pm$ 0.0078
Vanilla BERT	512	-	**0.933**	$\pm$ 0.0039
Vanilla XLNet	512	-	0.926	$\pm$ 0.0016

6.2.1 Overall Results

The empirical results of the tested methods and hyperparameters are presented in Table 6.2. Vanilla BERT-512 yields the best micro average F1-score with 0.933. The second-best model is the vanilla XLNet-512 with 0.926 F1 and a statistically significant lower score compared to vanilla BERT-512 (95% confidence interval). The vanilla Transformers generally outperform their Siamese counterparts. Siamese BERT (0.870 F1) and Siamese XLNet (0.870 F1) do not achieve the same performance as their vanilla architectures for the same 128 sequence length size, with scores of 0.920 (BERT-128) and 0.914 (XLNet-128) respectively. The shared contextual information during the encoding of document pairs most likely yields better performance of vanilla Transformers. GloVe (0.875 F1) outperforms Siamese BERT and Siamese XLNet, which makes GloVe preferable over Siamese Transformers since GloVe requires only a fraction of the computing resources and runs on commodity hardware. With an F1 score of 0.845 at its best configuration, Paragraph Vectors is the worst model.

[14] `https://doi.org/10.5281/zenodo.3713183`, last accessed: 18/01/2023
[15] `https://github.com/malteos/semantic-document-relations`, last accessed: 18/01/2023
[16] `https://ostendorff.org/r/jcdl2020-colab`, last accessed: 18/01/2023

In summary, we consider the results of GloVe and vanilla BERT as the most promising for future application scenarios. We hypothesize that an F1 score of above 0.90 is already suitable enough for generating relevant content-based recommendations. Expert users especially would tolerate some misclassifications in favor of otherwise undiscoverable information. This would be the case for target documents that are considered to be dissimilar to the seed with existing methods but are found to have a shared aspect with the help of our methods.

6.2.2 Impact of Sequence Length

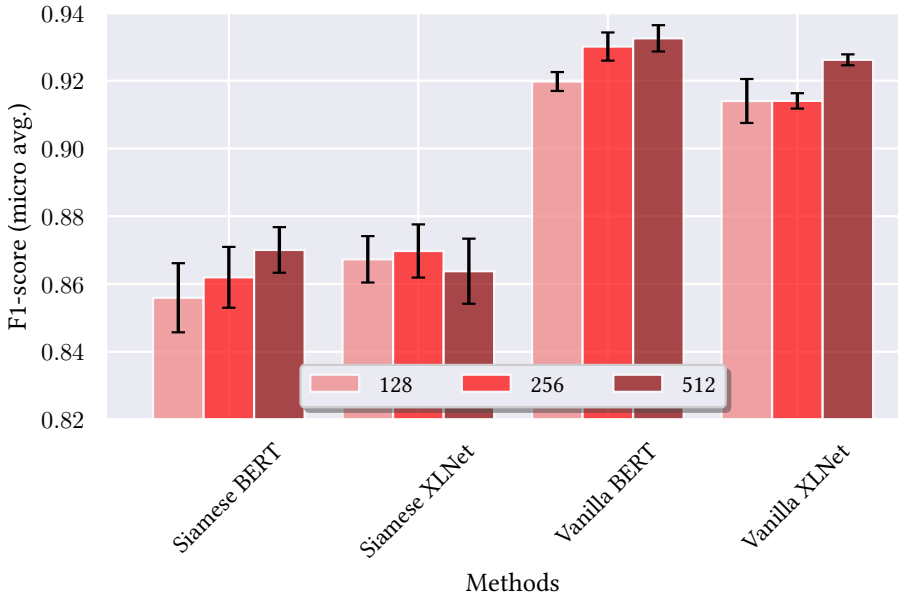


Figure 6.4: Results for vanilla and Siamese Transformers with respect to sequence length. Siamese models use $[u; v; |u - v|; u * v]$ as concatenation. Except for Siamese XLNet, 512 tokens achieve the best F1 scores for all models.

As explained in Section 6.1.2, we are particularly interested in the effect of the sequence length on the Transformer models. To illustrate this effect, Figure 6.4 shows the comparison of Siamese BERT, Siamese XLNet, vanilla BERT, and vanilla XLNet with respect to their sequence length (i.e., 128, 256, and 512). In this comparison, the Siamese models use the best-performing concatenation method, which is $[u; v; |u - v|; u * v]$.

Our findings reveal longer sequences correlate with better recommendation performance. For all models, except Siamese XLNet, the highest F1 score is achieved with 512 tokens and the second-highest with 256 tokens. One could think this outcome is to be expected. However, in Chapter 3, the performance of text- and graph-based document similarity measures declines for Wikipedia articles with more than 450 words. When comparing Siamese with vanilla Transformers, the vanilla models work with only half of the sequence length to encode one document of the pair. In vanilla Transformers, the document pairs share the sequence length, while in Siamese Transformers each document has its own Transformer sub-network (sequence length). For example, a vanilla 128-Transformer would use only 62 or 63 tokens of each document (three tokens are reserved for special tokens as Figure 6.2 shows). Thus, the small performance difference within vanilla BERT with 512 tokens (0.933 F1), 256 tokens (0.930 F1), and 128 tokens (0.920 F1) is

remarkable. Moreover, the performance differences should be considered relative to the higher computation expenses of longer sequences.

6.2.3 Impact of Concatenation

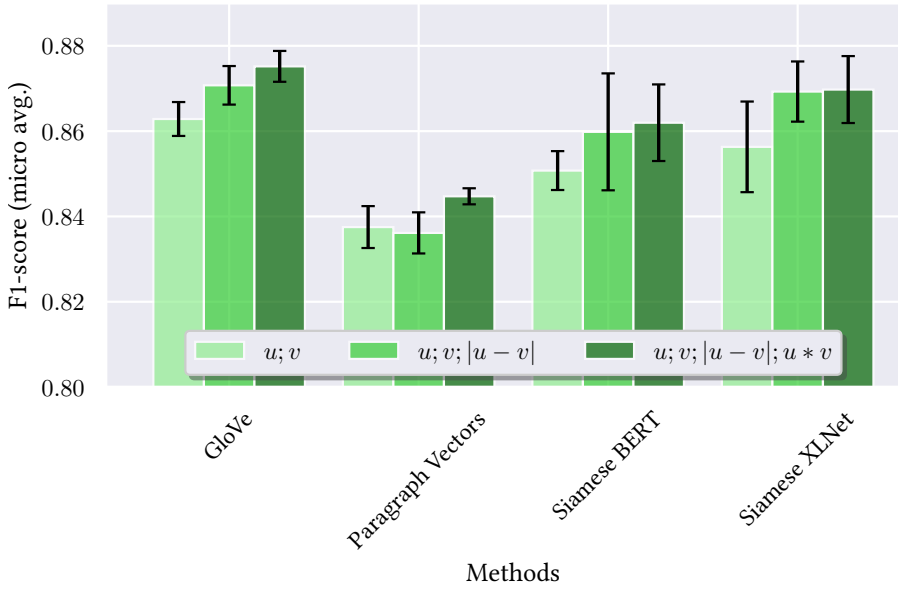


Figure 6.5: Results for the full-text document embeddings and Siamese Transformer-512 models with respect to concatenation.

Aside from the sequence length, we also analyze the different concatenation methods in GloVe, Paragraph Vectors, and the Siamese models (Figure 6.5). All models achieve the highest F1-score when the concatenation with an element-wise difference and product is used ($[u; v; |u - v|; u * v]$). Furthermore, we confirmed the results of Reimers and Gurevych (2019), i.e., the most crucial component is the element-wise difference $|u - v|$. Only for Paragraph Vectors the element-wise difference decreases the performance in comparison to the simple concatenation. However, this performance decrease is marginal and within the standard deviation. In general, the element-wise difference measures the distance between the dimensions of the two document vectors and, thus, ensures that similar pairs are closer to each other than dissimilar pairs. This effect is evident for Siamese BERT and Siamese XLNet, for which the element-wise difference yields the most substantial performance improvement. On the contrary, the element-wise product adds only a small improvement to our models.

6.2.4 Impact of Aspect Classes

We selected nine diverse Wikidata properties (aspect classes) to explore how the methods would respond to the individual challenges of each property. Table 6.3 presents precision, recall, and F1-score of the best four methods for the different model categories. Each score is the mean over the 4-fold cross-validation (cf. Table 6.2 for standard deviation). GloVe and Siamese BERT use $[u; v; |u - v|; u * v]$ as the concatenation method, and all Transformer models (Siamese BERT, vanilla BERT, and vanilla XLNet) use the 512 sequence length. The best aspect classes in terms of performance are *country of citizenship*, *none* (negative samples), and *different from*, whereas the

Table 6.3: Results for precision, recall, and F1-score with respect to aspect classes and as average over the aspect classes. The methods are GloVe, Siamese BERT, vanilla BERT, and vanilla XLNet. The results of the remaining models are published along with the code.

Methods →	**GloVe**			**Siamese BERT**			**Vanilla BERT**			**Vanilla XLNet**		
Aspect class ↓	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
country of citizenship	0.963	0.983	0.973	0.956	**0.996**	0.976	**0.993**	**0.996**	**0.994**	0.989	**0.996**	0.993
different from	0.856	0.843	0.849	0.872	0.899	0.885	**0.971**	0.931	**0.950**	0.969	**0.933**	**0.950**
educated at	0.683	0.729	0.703	0.730	0.740	0.734	0.759	**0.900**	**0.817**	**0.774**	0.759	0.763
employer	0.662	0.620	0.639	0.639	**0.769**	0.695	**0.892**	0.653	**0.740**	0.711	0.748	0.725
facet of	0.786	0.781	0.782	0.839	0.785	0.810	**0.916**	**0.908**	**0.911**	0.888	0.904	0.896
has effect	0.644	0.606	0.620	0.626	0.468	0.502	**0.783**	0.614	0.683	0.768	**0.658**	**0.704**
has quality	0.694	0.682	0.687	0.662	0.619	0.639	0.718	0.797	0.749	**0.763**	**0.799**	**0.774**
opposite of	0.672	0.666	0.667	0.540	0.791	0.640	0.761	0.763	0.756	**0.773**	**0.835**	**0.795**
symptoms	**0.887**	0.932	0.908	0.827	0.969	0.892	0.872	0.973	**0.920**	0.864	**0.984**	0.919
none	0.943	0.940	0.942	0.955	0.897	0.925	0.978	**0.981**	**0.979**	**0.979**	0.968	0.973
Micro Avg.	0.875	0.875	0.875	0.870	0.870	0.870	**0.933**	**0.933**	**0.933**	0.926	0.926	0.926
Macro Avg .	0.779	0.778	0.777	0.764	0.793	0.770	**0.864**	0.852	**0.850**	0.848	**0.858**	0.849

classes *employer*, *has quality*, *has effect* yield the lowest scores. Given that the best-performing aspects are also over-represented in terms of sample count, the outcome suggests that other classes could be improved by adding more training data. Still, the comparison of the *employer* class (389 test samples, vanilla BERT 0.740 F1) and *facet of* (336 test samples, vanilla BERT 0.911) reveals that the performance difference is also due to the diverse requirements of aspect classes themselves.

The superiority of vanilla BERT is also present in the aspect-specific evaluation scenario. However, vanilla BERT is outperformed by vanilla XLNet for three aspect classes with a small number of samples (*has effect*, *has quality*, and *opposite of*). In GloVe, *symptoms* has the highest precision score, which is probably caused by GloVe's ability to utilize the full text of articles in contrast to the Transformer models. Medical articles, like *Alcoholism* (Example 9 in Table 6.4), contain a section "Signs and symptoms" in which their symptoms are listed. However, such a section is not part of the first 512 tokens that fit into the Transformer input. When comparing precision and recall for all classes, both scores are mostly balanced. There is only one striking exception for vanilla BERT. For *employer*, the precision score of 0.829 is higher than the recall of 0.653, while for *educated at* the opposite occurs, with a precision of 0.759 and recall of 0.900, but in a smaller magnitude. A reason for this outcome is that *employer* is often confused with *educated at* as Figure 6.6 shows.

The confusion matrix in Figure 6.6 depicts which aspect classes are most often confused with each other. The predicted classes are taken from the vanilla BERT-512 method, whereby the number of true and predicted classifications is normalized to make the different classes comparable. With 27% of the test sample, *educated at* and *employer* are the most mistaken aspect classes in our

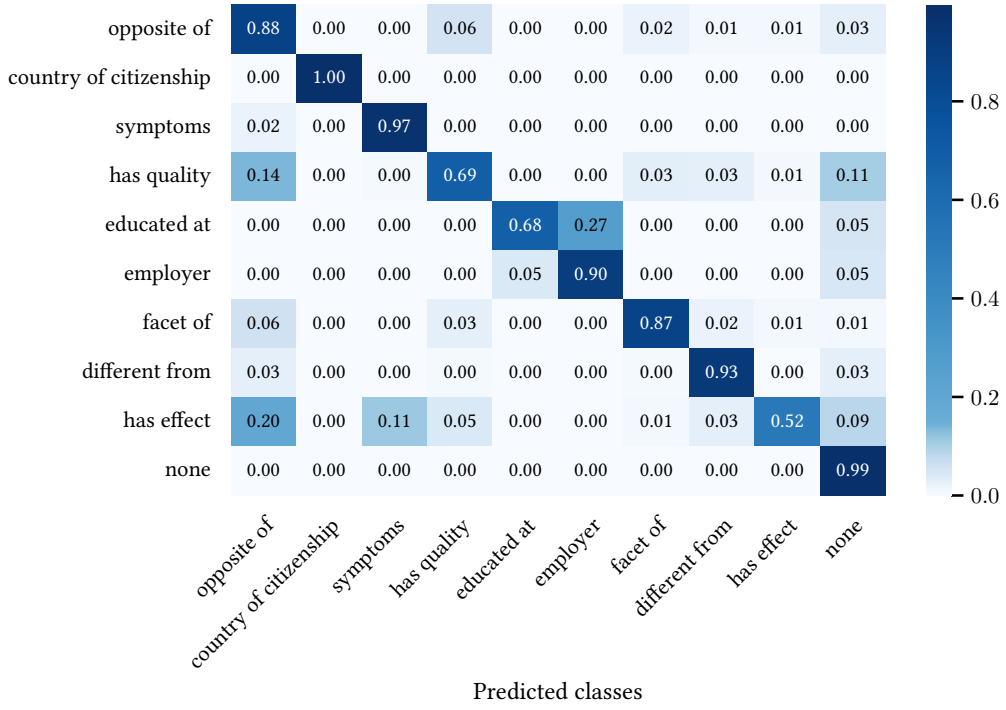


Figure 6.6: Confusion matrix for the predicted and Wikidata classes of vanilla BERT. The prediction count is normalized. The most frequent confusion is found with the *educated at* and *employer* aspect for 27% of the test samples.

experiments. We see this outcome because both aspect classes connect persons and organizations, and we assume it is harder for the classifier to tell the aspects apart. For instance, *Albert Einstein* could be employed or educated at *ETH Zurich* (Figure 6.1). The misclassification between different aspects is also found in *opposite of*, *has quality*, and *has effect*, which we conclude is because of similar reasons. In particular, the *opposite of* aspect class connects various types of articles.

6.2.5 Manual Sample Analysis

To validate our empirical findings, we manually examine the prediction from vanilla BERT-512 with a focus on errors (Table 6.4). According to the Wikidata properties, examples 1 and 2 show the desired classifier predictions apart from one misclassification. While *Armenia* is correctly identified as *Rudolf Muradyan's country of citizenship*, *Brazil* is not recognized. However, *Brazil* is also not mentioned in *Rudolf Muradyan's* Wikipedia article. The Wikidata statement is not reflected in the Wikipedia article, which states *Muradyan* as *Armenian* only. Consequently, both predictions would be correct when only considering the article text.

Two errors are exemplified in 3 and 4. Even though *Zaki Naguib Mahmoud's* article explicitly expresses the *educated at* aspect with the sentence "Mahmoud was educated at Cairo University", *Cairo University* is classified as his *employer*. Despite not being mentioned in *Mahmoud's* article, *King's College London* is also wrongly classified as his *employer*. In example 5, *Light* is

incorrectly classified as the quality of *Darkness*, not as the opposite of it. Still, *opposite of* is the class with the second-highest probability.

Table 6.4: Examples for the aspect-based similarity between Wikipedia articles (seed and target) as defined by Wikidata and as predicted by Vanilla BERT-512 with the first and second highest probability. Correct predictions are marked with ✓.

ID	Seed article	Target article	Wikidata class	Predictions (1st; 2nd)
1	Rudolf Muradyan	Brazil	country of citizenship	none; country of citizenship ✓
2	Rudolf Muradyan	Armenia	country of citizenship	country of citizenship ✓; none
3	Zaki Naguib Mahmoud	Cairo University	educated at	employer; educated at ✓
4	Zaki Naguib Mahmoud	King's College London	educated at	employer; educated at ✓
5	Light	Darkness	opposite of	has quality; opposite of ✓
6	Mexican Revolution	Mexican War of Independence	different from	has effect; symptoms
7	History of blogging	Blog	facet of	opposite of; different from
8	Iced tea	Ice-T	different from	none; different from ✓
9	Alcoholism	Cirrhosis	has effect	has effect ✓; none

Example 6 shows the *Mexican Revolution* as *different from* the *Mexican War of Independence*, which would be clear to a human user since the Wikipedia article contains a banner "Not to be confused with the Mexican War of Independence.". However, this banner is missing in the Wikipedia dump and, thus, is not available to the classifier. Many shared terms and vocabulary make their classification hard to predict for the *different from* aspect. Examples 7-9 illustrate a similar classifier's performance.

Our manual examination confirms the overall results. Most aspect classes are correctly identified, while some aspects are missing even if they are explicitly mentioned in the text.

6.3 Discussion

Given the results in Table 6.2, we can state that vanilla Transformers outperform all other methods. Rather unexpected is that BERT generally achieves slightly better results than XLNet. According to Yang et al. (2019), XLNet surpasses BERT on the related GLUE benchmark (Wang et al., 2019), so we were expecting a similar outcome. We hypothesize that this difference may be attributed to two reasons, pretraining on different corpora and smaller models compared to Yang et al. (2019). We use the `BASE`, not the `LARGE` versions of the pretrained models used by Yang et al. Furthermore, the published XLNet `BASE` model we considered is pretrained on different data than the one in Yang et al. (2019)[17]. In contrast to BERT, XLNet is pretrained on Web

[17] See `https://github.com/zihangdai/xlnet#released-models` "*This model (XLNet-Base) is trained on full data (different from the one in the paper)*", last accessed: 18/01/2023.

corpora in addition to Wikipedia and the BooksCorpus (Zhu et al., 2015). The almost exclusive pretraining on Wikipedia most likely causes BERT to surpass XLNet. The effect of domain-specific pretraining on the performance of the language model has already been shown (Beltagy et al., 2019).

Our results also shows that the evaluated Siamese networks cannot capture the aspect-based similarity, unlike vanilla Transformers. In Siamese models, the encoding of the seed document does not affect the target, and vice-versa. Only the MLP is exposed to the documents as a pair in the form of concatenated document vectors. During the encoding phase, the shared aspects between the documents play no role. On the contrary, the multi-head attention mechanism in the vanilla Transformers allows attending to the two documents simultaneously. As the results suggest, this ability is crucial for pairwise document classification.

The Siamese models are also outperformed by the computationally less expensive GloVe. Generally, the Siamese models are very similar to GloVe (and Paragraph Vectors) since they derive two document vectors and classify their concatenation. So the method's performance ultimately depends on its ability to encode the document's content. Arora et al. (2017) have shown that the weighted average of word vectors can outperform more sophisticated methods. GloVe benefits from the fact that it utilizes the full-text article in contrast to the Transformers, which use only the 512 first tokens of the article text. As a result, GloVe is a reasonable method for practical scenarios in which computational resources are critical concerns. In such scenarios, one would avoid classifying all possible n^2 document pairs. Instead, evidently unrelated pairs must be filtered out with traditional similarity measures at first, as done with the pairwise baseline in Chapter 8.

Regarding the different aspect classes, almost all results present reasonable performance. Moreover, complex aspect classes like *facet of* or *has effect* yield promising results since they are attractive for the recommender use case. As examples 1 and 2 show in Table 6.4, current systems already reveal wrong or contradicting information between Wikidata and Wikipedia. The results suggest that increasing the sequence length beyond the 512 tokens could further improve the Transformer models. Higher sequence length is already possible with XLNet's architecture, but it would require a pretraining step with longer sequences.

From aspect-based similarity to recommendations. Classifying the aspect classes is not a purpose on its own. We envision content-based recommendations as an example of a downstream task. The obtained aspect information can be used for diverse or focused recommendations. As the aspect classes describe different facets of the seed document, one could diversify the recommendations. Choosing the recommendations from documents connected with different aspect classes to the seed document would ensure diversity. In Figure 6.1, the *German Empire* and *ETH Zurich* can be considered as diverse recommendations since they present different aspects of *Albert Einstein*, i.e., his citizenship and education. When considering documents that are connected to the seed document (i.e., one common document) over two edges (i.e., different aspects), recommendations focusing on specific aspects are more feasible. Diverse and focused recommendations could be especially suitable for scenarios in which different perspectives are required for the same seed article. In contrast to user-based recommender systems, content-based approaches usually struggle to account for the specific preferences of their users. One way to respect different information needs would be to suggest alternative recommendation sets that are focused on specific aspects. In the example of *Albert Einstein*, shown in Figure 6.1, focused

recommendation sets could include articles about people with a similar citizenship or a similar educational backgrounds. Additionally, the intersection of aspect classes would allow finding people with a similar citizenship but different educational backgrounds. The classification of the aspect-based document similarity, as done in our experiments, is the foundation for such recommendations.

Generalization. Given the goal of applying the tested methods to other literature domains, the question arises whether our findings are generalizable. We acknowledge that Wikipedia is presumable a simpler corpus compared to other literature domains like research papers. Wikipedia articles represent distinct entities, while most aspect classes are explicitly expressed in the article text. However, even research papers express aspects in their abstracts, e.g., "we used X" or "we found Y". Accordingly, we hypothesize that our systems would yield worse but still satisfactory results under comparable conditions (size of training data, pretraining on an in-domain corpus, etc.). A reference value would be the F1-score of 0.65, which was achieved by SciBERT on the related task of citation intent classification (Beltagy et al., 2019). While the effort for the unsupervised pretraining of a language model is reasonable, we recognize that annotating sufficient training data for other corpora is one of the most challenging tasks. In Chapter 7, we demonstrate that our results are transferable to the domain of research papers.

6.4 Summary of the Chapter

This chapter investigated Research Task **III** and introduced pairwise document classification to determine the aspect-based similarity between documents as an underlying task to advance content-based recommender systems and other information retrieval applications. We elaborated on why document similarity measures do not account for the heterogeneous semantics of extensive documents and argued that similarity needs aspect information that defines what it relates to.

The task of measuring the aspect-based similarity was implemented as a multi-class classification of document pairs. We demonstrated the viability of this approach with a new proposed dataset of 32,168 Wikipedia article pairs and Wikidata properties that define shared aspects among these articles. In an empirical study, we implemented six different methods (GloVe, Paragraph Vectors, Siamese BERT, Siamese XLNet, vanilla BERT, and vanilla XLNet) and evaluated them under different settings regarding the concatenation method and sequence length (Table 6.2). Our evaluation revealed a sequence length of 512 tokens as the best-performing sequence limit for the Siamese and vanilla Transformer models. In addition, we identified $[u; v; |u - v|; u * v]$ as the best concatenation method for GloVe, Paragraph Vectors, and the Siamese Transformer models. With the manual sample examination and our evaluation for different aspect classes, we showed the behavior of the classifiers when exposed to different input data and provide analysis from different perspectives. Moreover, the manual analysis confirmed our empirical results.

Our findings demonstrated that pairwise document classification is a solvable task using the evaluated methods. Even abstract aspect classes, like *facet of*, yielded considerably high F1 scores. This outcome motivates us to investigate the aspect-based similarity between documents of other literature domains. Therefore, the subsequent chapter will evaluate the pairwise classification approach in the context of research papers.

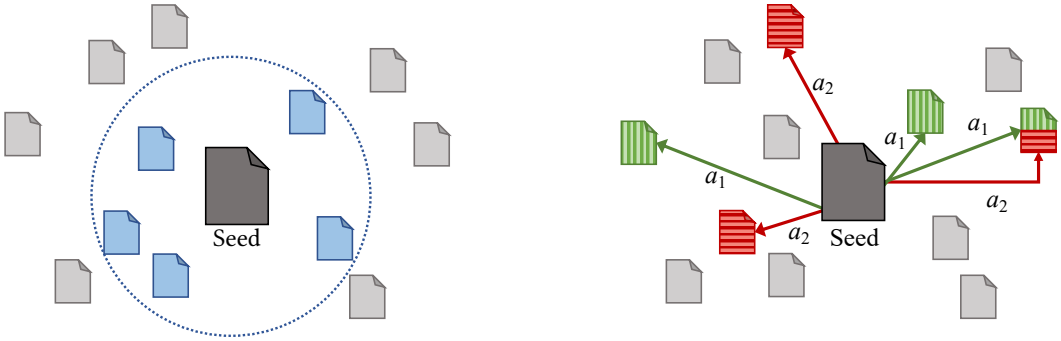
Chapter 7

Pairwise Classification for Research Papers

The last chapter presented the first steps towards integrating aspect information into document similarity measures. The experiments with Wikipedia articles and Wikidata properties were successful. This chapter continues the work on Research Task **III** and on the pairwise document classification approach with two major extensions. We move from a single-label to a multi-label classification problem, i.e., documents can simultaneously be similar in multiple aspects. Furthermore, we extend the approach to research papers that express shared aspects presumably less explicit, making the classification task more challenging. The chapter's content is based on Ostendorff et al. (2020b).

"Aspect-based Document Similarity for Research Papers" by **Malte Ostendorff**, Terry Ruas, Till Blume, Bela Gipp, and Georg Rehm. In: *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, 2020.

Content-based recommender systems assist researchers in finding relevant papers from the ever-increasing amount of scientific literature. At the same time, research papers can be semantically similar in many different ways. For instance, Huang et al. (2020) consider *method* or *findings* as the aspects that make two papers alike. Differentiating between these aspects could facilitate innovations and scientific discoveries (Chan et al., 2018). This makes research papers a particularly interesting domain for aspect-based document similarity.



(a) Aspect-free similarity

(b) Aspect-based similarity

Figure 7.1: Most research paper recommender systems rely on similarity measures between a seed and the k most similar target paper (a). This neglects the aspects which make two or more papers similar. In aspect-based similarity (b), papers are related according to the inner aspects connecting them (a_1 or a_2), which we measure with a pairwise comparison.

In classical research paper recommender systems (Section 2.1.1.1), papers are recommended as the k nearest neighbors based on a single similarity measure (Figure 7.1a). As a result, a researcher using such a recommender system does not know whether a paper is recommended because it uses a similar *method* or reports similar *findings*. With aspect-based similarity (Figure 7.1b), relevant

papers can be recommended specifically when they are similar in a given aspect. Figure 7.1b shows an example with two aspects, a_1 (green) and a_2 (red), which would correspond to *method* and *findings* respectively.

Following the line of work from Chapter 6, we infer aspect information for document similarity by formulating the problem as a multi-class classification of research paper pairs. We extend the approach of Chapter 6 from a single-label to a multi-label scenario and focus on scientific literature instead of Wikipedia articles. More specifically, we aim to answer two research questions.

Research questions

RQ1: Does the multi-class pairwise document classification approach achieve comparable results in scientific literature as in Wikipedia articles?

RQ2: How do model architecture and pretraining objectives of the Transformer language model affect their ability to measure the aspect-based similarity?

Similar to the work of Jiang et al. (2019) and Cohan et al. (2020), we use citations as training signals. Instead of using citations for binary classification (i. e., similar and dissimilar), we include the section's title where a citation occurs as a label for a document pair. The section titles of citations describe the aspect-based similarity of citing and cited papers. Our two datasets originate from the ACL Anthology (Bird et al., 2008) and CORD-19 (Wang et al., 2020a).

In summary, this chapter's main contributions are:

1. We extend aspect-free document similarity to aspect-based in a multi-label multi-class document classification task.
2. We apply aspect-based document similarity successfully on research papers.
3. We evaluate six Transformer-based models and a baseline for the pairwise document classification task.

The chapter's source code, trained models, and two datasets from the computational linguistics and biomedical domain are publicly available.[1]

The remainder of this chapter is structured as follows: First, we introduce the general methodology, i.e., the datasets and the evaluated methods. Subsequently, we present the overall results in Section 7.2.1, the impact of aspects in Section 7.2.2, and the manual sample analysis in Section 7.2.3. In Section 7.3, we discuss the results of all evaluations. Finally, we summarize the main findings of this chapter.

7.1 Methodology

This section presents our methodology for the aspect-based similarity of research papers.

7.1.1 Datasets

The generation of human-annotated data for research paper recommendations is costly and usually limited to small quantities (Beel et al., 2016b). The small dataset size hampers the application

[1] `https://github.com/malteos/aspect-document-similarity`, last accessed: 18/01/2023

of learning algorithms. To mitigate the data scarcity problem, researchers rely on citations as ground truth, i.e., when a citation exists between two papers, the two papers are considered similar (Cohan et al., 2020; Jiang et al., 2019). Whether one or no citation exists corresponds to a label for a binary classification that corresponds to aspect-free similarity.

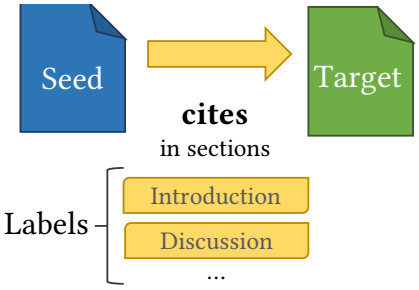


Figure 7.2: We use the citations' section titles as labels for the pair of citing and cited papers (seed and target). The sections define the aspects of the similarity.

To refine the similarity from *aspect-free* to *aspect-based*, we transfer this idea to the problem of multi-label multi-class classification. As ground truth, we adopt the title of the section in which the citation from paper *A* (seed) to *B* (target) occurs as label class (Figure 7.2). The classification is multi-class because of multiple section titles and multi-label because paper A can cite B in multiple sections. For example, paper *A* citing *B* in the *Introduction* and *Discussion* section would correspond to one sample of the dataset.

ACL Anthology. We use the ACL Anthology Reference Corpus version 2.0 (Bird et al., 2008) as a dataset. The corpus comprises 22,878 research papers about computational linguistics. Aside from full texts, the ACL Anthology dataset provides additional citation data. The citations are annotated with the title of the section in which the citation markers are located. This information is required for our experiments.

CORD-19. The COVID-19 Open Research Dataset (CORD-19) is a collection of papers on COVID-19 and related coronavirus research from several biomedical digital libraries (Wang et al., 2020a).[2] CORD-19 contains approx. 1M papers with nearly 370K papers having full-text content. The citation and metadata of all CORD-19 papers are standardized according to the processing pipeline of Lo et al. (2020). Citations in CORD-19 are also annotated with section titles.

Data preprocessing. Considering the ACL Anthology and CORD-19, we derive two datasets for pairwise multi-label multi-class document classification. The section titles of the citations, i. e., the label classes, are presented in Table 7.1. We normalize sections titles (lowercase, letters-only) and resolve combined sections into multiple ones (*Conclusion and Future Work* to *Conclusion*; *Future Work*). We query the API of DBLP (Ley, 2009) and Semantic Scholar (Lo et al., 2020) to match citations and retrieve missing information from the papers such as abstracts. Invalid papers without any text or duplicated ones are removed. We divide both datasets, ACL Anthology and CORD-19, into ten classes according to their number of samples, so that the first nine compose the most popular section titles and the tenth (*Other*) groups the remaining ones.

[2] `https://www.semanticscholar.org/cord19/download`, last accessed: 18/01/2023

Table 7.1: Label class distribution as extracted from the citations' section titles in the two datasets. We report the top nine section classes in decreasing order and group the remaining as *Other*.

(a) ACL Anthology

Label class	Count
Introduction	16,279
Related Work	12,600
Experiment	4,025
Background	1,365
Results	1,181
Conclusion	1,158
Discussion	1,132
Evaluation	971
Methods	719
Other	22,249

(b) CORD-19

Label class	Count
Introduction	15,108
Discussion	13,258
Conclusion	1,003
Results	910
Methods	523
Background	454
Materials	420
Virus	218
Future work	171
Other	43,154

Even though the selection of our ten classes might neglect section title variations in the literature, our approach still doubles the number of research aspects from existing datasets (Chan et al., 2018; Huang et al., 2020). The resulting class distribution is unbalanced, but it reflects the true nature of the corpora as Table 7.5 shows. Scripts for reproducing the datasets are available with our source code.

Negative sampling. In addition to the ten positive classes (Table 7.1), we introduce a class named *None* that works as a negative counterpart for our positive samples in the same proportion (Mikolov et al., 2013a). The *None* document pairs are randomly selected and dissimilar from each other. A random pair of papers is a negative sample when the papers do not exist as a positive pair, are not co-cited together, do not share any authors, and are not published in the same venue. A more elaborated sampling strategy similar to the one from Chapter 5 was omitted for simplicity. We generate 24,275 negative samples for ACL Anthology and 33,083 for CORD-19. These samples let the models distinguish between similar and dissimilar documents.

7.1.2 Evaluated Methods

We focus on sequence pair classification with models based on the Transformer architecture (Vaswani et al., 2017). Transformer-based models are often used in text similarity tasks (Jiang et al., 2019; Reimers and Gurevych, 2019). Moreover, we found in Chapter 6 that vanilla Transformers, e.g., BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), outperform Siamese networks (Bromley et al., 1993) and traditional word embeddings, e.g., GloVe (Pennington et al., 2014), Paragraph Vectors (Le and Mikolov, 2014), in the pairwise document classification task. Hence, we exclude Siamese networks and pretrained word embedding models in this chapter's experiments.

Instead, we investigate six Transformer variations and an additional baseline for comparison. The titles and abstracts of research paper pairs are used as input for the model so that the `[SEP]`

token separates seed and target paper (Figure 7.3). This procedure is based on our prior findings (Chapter 6). We use only the paper abstracts in our experiments since many full texts are not freely available. Another reason is that the selected Transformers are limited to 512 tokens.

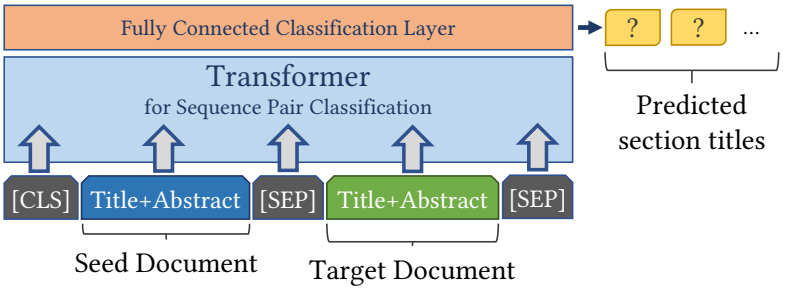


Figure 7.3: A Transformer model with titles and abstracts as input is used for classification.

Baseline LSTM. The baseline is a bidirectional LSTM (Hochreiter and Schmidhuber, 1997). To derive representations for document pairs, we feed the title and abstract of two papers through the LSTM, where the papers are separated with a special separator token. We use the SpaCy tokenizer (Honnibal et al., 2020) and word vectors from fastText (Bojanowski et al., 2017). The word vectors are trained on the abstracts of ACL Anthology or CORD-19.

BERT, Covid-BERT & SciBERT. BERT is a neural language model based on the Transformer architecture (Devlin et al., 2019). Commonly, BERT models are pretrained on large text corpora in an unsupervised fashion. The two pretraining objectives are the recovery of masked tokens (i. e., mask language modeling) and next sentence prediction (NSP). After pretraining, BERT models are fine-tuned for specific tasks like sentence similarity (Reimers and Gurevych, 2019) or document classification (Ostendorff et al., 2019). Several BERT models pretrained on different corpora are publicly available.

For our experiments, we evaluate three BERT variations. (1) The BERT model from Devlin et al. (2019), trained on English Wikipedia and the BooksCorpus (Zhu et al., 2015). (2) SciBERT (Beltagy et al., 2019), a variation of BERT tailored for scientific literature, which is pretrained on computer science and biomedical research papers. (3) Covid-BERT (Chan, 2020) is the original BERT model from Devlin et al. (2019) but fine-tuned on the CORD-19 corpus. BioBERT (Lee et al., 2019) is another BERT model specialized in the biomedical domain. Nonetheless, we exclude BioBERT as SciBERT even outperforms it on biomedical tasks (Beltagy et al., 2019). All three models, i. e., BERT, SciBERT, and Covid-BERT, are similar in their structure, except for the corpus used during the language model training.

SciNCL. As presented in Chapter 5, SciNCL is a citation-informed SciBERT language model. The key difference to SciBERT is that SciNCL is explicitly trained for document representations and not for a typical language modeling task.

RoBERTa. Liu et al. (2019) propose RoBERTa, which is a BERT model trained on larger batches, longer training time, and drops the NSP task from its objective. Moreover, RoBERTa uses additional corpora for pretraining, namely Common Crawl News (Nagel, 2016), OpenWebText (Gokaslan and Cohen, 2019), and STORIES (Trinh and Le, 2018).

XLNet. Unlike BERT, XLNet (Yang et al., 2019) is not an autoencoder but an autoregressive language model. XLNet does not employ NSP. We use the XLNet model published by its authors, which is pretrained on Wikipedia, BooksCorpus (Zhu et al., 2015), Giga5 (Parker et al., 2011), ClueWeb 2012-B (Callan et al., 2009), and Common Crawl (Elbaz, 2007).

ELECTRA. ELECTRA (Clark et al., 2020) has the pretraining objective of detecting replaced tokens in the input sequence as an addition to mask language modeling. For this objective, Clark et al. use a generator that replaces tokens and a discriminator network that detects the replacements. The generator and discriminator are both Transformer models. ELECTRA does not use the NSP objective. For our experiments, we use the discriminator model of ELECTRA. The pretrained ELECTRA discriminator model is pretrained on the same data as BERT.

7.1.3 Implementation Details

We choose the LSTM hyperparameters according to the findings of Reimers and Gurevych (2017) as follows: 10 epochs for training, batch size $b = 8$, learning rate $\eta = 1^{-5}$, two LSTM layers with 100 hidden size, attention, and dropout with probability $d = 0.1$. While the LSTM baseline uses vanilla PyTorch, all Transformer-based techniques are implemented using the Huggingface API (Wolf et al., 2020). Each Transformer model is used in its `BASE` version. The hyperparameters for Tranformer fine-tuning are aligned with Devlin et al. (2019): four training epochs, learning rate $\eta = 2^{-5}$, batch size $b = 8$, and Adam optimizer with $\varepsilon = 1^{-8}$.

We conduct the evaluation in a stratified k-fold cross-validation with $k = 4$ (i.e., the class distribution remains identical for each fold). On average, this produces 54,618/18,206 train/test samples for ACL Anthology, and 74,436/24,812 train/test samples for CORD-19.

The source code, datasets, and trained models are publicly available on GitHub[3] and Zenodo[4]. We provide a Google Colab to try out the trained models on any papers from Semantic Scholar.[5]

7.2 Evaluation

Our results are divided into three parts: overall, label classes, and qualitative evaluation.

7.2.1 Overall Results

The overall results of the quantitative evaluation are presented in Table 7.2. We conduct the evaluation as 4-fold cross-validation based on our datasets. We report micro and macro averages for precision, recall, and F1-score to account for the unbalanced label class distribution (see Section 7.1.1).

Given the overall scores, SciBERT is the best method with 0.326 macro-F1 and 0.678 micro-F1 on ACL Anthology, and with 0.439 macro-F1 and 0.833 micro-F1 on CORD-19. All Transformer models outperform, in all metrics, the $\text{LSTM}_{baseline}$ except for the micro-precision on ACL Anthology. The gap between macro and micro average results is due to discrepancies between the label classes (see Section 7.2.2). BERT, SciBERT, SciNCL, and Covid-BERT perform better,

[3] `https://github.com/malteos/aspect-document-similarity`, last accessed: 18/01/2023
[4] `https://doi.org/10.5281/zenodo.4087898`, last accessed: 18/01/2023
[5] `https://ostendorff.org/r/coling2020-colab`, last accessed: 18/01/2023

Table 7.2: Overall F1 score, precision, and recall for the macro and micro averages of eight methods for ACL Anthology and CORD-19. SciBERT yields the best results in both datasets.

Dataset	ACL Anthology						CORD-19					
	macro avg			micro avg			macro avg			micro avg		
	F1	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**
$\text{LSTM}_{baseline}$	0.063	0.069	0.058	0.290	**0.761**	0.179	0.128	0.137	0.121	0.579	0.758	0.469
BERT	0.256	0.317	0.238	0.641	0.719	0.578	0.387	**0.619**	0.357	0.822	0.840	0.806
Covid-BERT	0.270	0.404	0.253	0.648	0.715	0.592	0.394	0.578	0.364	0.818	0.836	0.802
SciBERT	**0.326**	**0.458**	**0.303**	**0.678**	0.725	**0.637**	**0.439**	0.560	**0.401**	**0.833**	**0.846**	**0.820**
SciNCL	0.319	0.379	0.296	0.671	0.724	0.624	0.428	0.573	0.393	0.830	0.845	0.816
RoBERTa	0.250	0.285	0.232	0.626	0.703	0.564	0.332	0.473	0.316	0.820	0.840	0.801
XLNet	0.263	0.372	0.250	0.645	0.705	0.595	0.362	0.523	0.345	0.817	0.832	0.804
ELECTRA	0.245	0.287	0.228	0.616	0.693	0.554	0.280	0.306	0.276	0.820	0.840	0.801

on average, for ACL Anthology and CORD-19 when compared to the baseline and the other Transformer-based models. For ACL Anthology, the methods produce the same rankings for both macro and micro averages.

SciBERT achieves the highest scores with a large margin, followed by SciNCL, Covid-BERT, XLNet, and BERT. The lowest scores are from RoBERTa (0.626 micro-F1) and ELECTRA (0.616 micro-F1). In terms of macro average, the methods present the same ranking for CORD-19 and ACL Anthology except for BERT, which outperforms XLNet. Only for micro average on CORD-19 the outcome is different, i. e., ELECTRA and RoBERTa achieve higher F1 scores than Covid-BERT and XLNet. Even though Covid-BERT is fine-tuned on CORD-19 its performance yields only a 0.818 micro-F1. SciBERT outperforming SciNCL can be attributed to SciNCL's training being tailored towards document representations from the `[CLS]` token, which is fundamentally different from the sequence pair classification evaluated in this experiment. As already shown in Chapter 6, using the next-sentence-prediction objective for pretraining seems especially important for the performance of sequence pair classification due to the similarity of the task and the pretraining objective.

7.2.2 Impact of Aspect Classes

We divide both datasets (ACL Anthology and CORD-19) into 11 aspect classes between positive and negative examples (Section 7.1.1). Each class represents a different section in which a paper is cited. The section indicates in what aspects the two papers are similar. The aspects can also be ambiguous making their classification a hard task. The following section investigates the classification performance with respect to the different aspect classes. Table 7.3 presents F1 score, precision, and recall of SciBERT for all 11 labels. Additionally, we include the overall results for single and multi-label samples (i. e., 2, and ≥ 3). The remaining methods from Table 7.2 present lower but proportionally similar scores.[6]

[6]The detailed data on the remaining methods is available together with the trained models in our GitHub repository.

Table 7.3: Results of SciBERT on ACL Anthology and CORD-19 datasets per aspect class or aggregated by label count with number of test samples, F1 score, precision, and recall.

ACL Anthology					CORD-19				
Aspect	**Samples**	**F1**	**Prec.**	**Rec.**	**Aspect**	**Samples**	**F1**	**Prec.**	**Rec.**
Background	341	0.436	0.651	0.329	Background	113	0.617	0.655	0.588
Conclusion	289	0.000	0.000	0.000	Conclusion	250	0.274	0.563	0.182
Discussion	283	0.000	0.000	0.000	Discussion	3314	0.636	0.641	0.631
Evaluation	242	0.008	0.396	0.004	Future work	42	0.032	0.150	0.018
Experiment	1006	0.360	0.491	0.284	Introduction	3777	0.644	0.669	0.620
Introduction	4069	0.527	0.576	0.486	Materials	105	0.241	0.552	0.157
Methods	179	0.014	0.208	0.007	Methods	130	0.205	0.519	0.130
Related work	3150	0.638	0.660	0.617	Results	227	0.322	0.558	0.227
Results	295	0.015	0.475	0.008	Virus	54	0.000	0.000	0.000
Other	5562	0.645	0.646	0.645	Other	10788	0.876	0.872	0.879
None	6068	0.942	0.934	0.951	*None*	8270	0.979	0.980	0.977
Number of labels					**Number of labels**				
1 label	15652	0.721	0.717	0.726	1 label	22885	0.860	0.844	0.876
2 labels	1968	0.540	0.738	0.425	2 labels	1632	0.656	0.849	0.535
$\geq$ 3 labels	585	0.492	0.857	0.345	$\geq$ 3 labels	295	0.590	0.925	0.433

The *None* class has the highest F1 score by a large margin (0.942 for ACL Anthology, 0.980 for CORD-19). The *Other* class shows the second-best F1 score, which in a similar-dissimilar classification scenario can be interpreted as an opposite class to the *None* label. The remaining positive aspect classes yield lower scores but also a lower number of samples. Since we conduct a 4-fold cross-validation the ratio of train and test samples is 75/25. In CORD-19, 10,788 *Other* test samples exist compared to 3,777 *Introduction* samples, which is the most common section title (Table 7.1). Still, the lower number of samples does not necessarily correlate with low accuracy. In ACL Anthology, the aspect class *Related work* (3,150 samples) yields higher scores when compared to *Introduction* (4,069 samples) with an F1 score of 0.638 and 0.527 respectively. The aspect class *Background* in CORD-19 has an F1 score of 0.617 despite having only 113 samples. The results in Table 7.3 show an impact from the aspect classes on the overall performance. Six aspect classes (ACL Anthology - *Conclusion*, *Discussion*, *Evaluation*, and *Methods*; CORD-19 - *Future work* and *Virus*) have F1 scores between zero and 0.05. The discrepancy in the number of samples and difficulty in uncovering latent information from aspects contribute to the decrease in some classes' accuracy. Even for domain experts, the location of whether one paper cites another, e. g., in *Introduction* or *Experiment*, is not trivial to predict.

The bottom rows in Table 7.3 illustrate the effect of multi-labels (similarity in more than one aspect class). F1 scores decrease on both datasets as the number of labels increases. This is due to decreasing recall. The precision increases with more labels. Table 7.4 shows a portion of the distribution of multi-label samples in CORD-19 and corresponding SciBERT predictions. When two or more aspect labels are present, SciBERT often correctly predicts one of the aspects but

not the others. For example, the label pair of *Discussion* and *Introduction* (D,I) has only 22% test samples correct. Still, SciBERT correctly predicts for the remaining samples one of the two aspects, i. e., either *Discussion* (35%) or *Introduction* (31%). We see comparable results for other multi-labels such as *Discussion*, *Introduction*, and *Other* (D,I,O).

Table 7.4: Confusion matrix of selected multi-labels for SciBERT on CORD-19 (N=None, C=Conclusion, O=Other, D=Discussion, I=Introduction, R=Results). For example **(in bold)**, 459 test samples are assigned to *Discussion* and *Introduction* (D,I), of which 103 are correctly classified. The remaining samples are mostly classified as single-label, i. e., either *Discussion* (163) or *Introduction* (146).

Ground Truth		**Predictions**														
Sections	**Sample**	***N***	**B**	**C**	**D**	**I**	**O**	**R**	**C,O**	**D,I**	**D,O**	**D,R**	**I,O**	**O,R**	**D,I,O**	**D,O,R**
C,D	21	-	-	-	1	6	7	-	-	1	-	-	1	-	-	-
C,O	79	-	-	2	1	2	58	-	13	-	-	-	3	-	-	-
D,I	**459**	**1**	-	-	**163**	**146**	**17**	-	-	**103**	**7**	**2**	**9**	-	**10**	-
D,O	351	1	2	-	102	30	120	1	-	15	59	1	4	1	4	-
D,R	65	1	-	-	6	10	10	-	-	1	3	28	-	-	-	1
I,O	453	2	1	-	15	114	215	1	-	12	16	1	62	-	9	-
D,I,O	142	1	1	-	28	31	11	-	-	33	8	-	12	-	14	-
D,O,R	23	-	-	-	5	-	7	-	-	-	5	2	-	1	-	1

7.2.3 Manual Sample Analysis

To validate the quantitative findings, we qualitatively evaluate the prediction from SciBERT on ACL Anthology. Table 7.5 presents example papers including SciBERT's predictions of whether the seed cites the target paper and in which section the citation should occur. We manually examine the predictions for their correctness.

The first example of Bär et al. (2012) and Agirre et al. (2012) is a correct prediction. Given the ground truth, the aspect is *Other* (the citation occurs in a section called "Results on Test Data"). We assess *Introduction* as a potentially valid prediction since Bär et al. (2012) is a submission to the shared task described in Agirre et al. (2012). Therefore, one could have cited it in the introduction. All predictions in the example 2 are correct. Compared to the other examples, we consider example 2 a simple case as both papers mention their topic (i. e., query segmentation) in the title and in the first sentence of the abstract (hint for *Introduction* label). Both abstracts of example 2 also refer to "mutual information and EM optimization" as their methods. In example 3, Zhang and Clark (2009) and Xi et al. (2012) do not share any citation. Hence, the paper pair is assigned with the *None* aspect according to the ground truth data even though they are topically related. Zhang and Clark (2009) and Xi et al. (2012) are both about Chinese machine translation. Still, we disagree with the model's prediction of *Experiment* since the two papers conduct different experiments making *Experiment* an invalid prediction. Example 4's predictions are correct. Polifroni et al. (1992) is published before Winterboer and Moore (2007) and, therefore, a citation cannot exist. Nonetheless, the two papers cover a related topic. Thus, one could expect a citation of Polifroni et al. (1992) in Winterboer and Moore (2007) in the introduction section as SciBERT predicted. The model finds this semantic similarity given their

Table 7.5: Example aspect-based similarity of research paper pairs (seed and target) as defined by citing section title and as predicted by SciBERT. Based on the test set, correct predictions are marked with ✓, invalid ones with ×.

	Seed Paper	Target Paper	Citation	Prediction
1	UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures (Bär et al., 2012)	SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity (Agirre et al., 2012)	Other	Introduction×
2	Query segmentation based on eigenspace similarity (Zhang et al., 2009)	Unsupervised query segmentation using generative language models and Wikipedia (Tan and Peng, 2008)	Introduction, Experiment	Introduction✓, Experiment✓
3	Transition-Based Parsing of the Chinese Treebank using a Global Discriminative Model (Zhang and Clark, 2009)	Enhancing Statistical Machine Translation with Character Alignment (Xi et al., 2012)	None	Experiment×
4	Experiments in evaluating interactive spoken language systems (Polifroni et al., 1992)	Evaluating information presentation strategies for spoken recommendations (Winterboer and Moore, 2007)	None	Introduction×, Other×
5	Similarity-based Word Sense Disambiguation (Karov and Edelman, 1998)	Targeted disambiguation of ad-hoc, homogeneous sets of named entities (Wang et al., 2012)	None	None✓
6	SciSumm: A Multi-Document Summarization System for Scientific Articles (Agarwal et al., 2011)	Improving question-answering with linking dialogues (Gandhe et al., 2006)	None	None✓

latent information on the topic. Examples 5-6 present two pairs for which *None* was correctly predicted according to the ground truth. Agarwal et al. (2011) and Gandhe et al. (2006) from Example 6 are topically unrelated as their titles already suggest. However, Karov and Edelman (1998) and Wang et al. (2012) on Example 5 share the topic of *disambiguation*. Thus, we would agree with the prediction of a positive aspect class, i.e., the papers are similar.

In summary, the qualitative evaluation does not contradict the quantitative findings. SciBERT distinguishes documents at a higher level and classifies which aspects make them similar. In addition to traditional document similarity, the aspect-based predictions allow us to assess how two papers relate to each other at a semantic level. For instance, whether two papers are similar in the aspects of *Introduction* or *Experiment* is valuable information, especially in literature reviews.

7.3 Discussion

In the experiments, SciBERT outperforms all other methods in pairwise document classification. We observe that in-domain pretraining and next-sentence-prediction objectives often lead to higher F1 scores. Transferring generic language models to a specific domain usually decreases the performance in our experiments. A possible explanation for this is the narrowly defined vocabulary in ACL Anthology or CORD-19. Beltagy et al. (2019) and Lee et al. (2019) have also explored the transfer learning between domains with similar findings. Covid-BERT seems

to be an exception as it yields lower results (micro-F1) than BERT on CORD-19 even though Covid-BERT was fine-tuned on CORD-19. We observe the language model fine-tuning in Covid-BERT does not guarantee a higher performance compared to pretraining from scratch in SciBERT. However, Covid-BERT's authors provide too little information to give a proper explanation for its performance.

Apart from in-domain pretraining, the next-sentence-prediction objective has a positive effect on the models. All BERT-based systems, which use next-sentence-prediction, outperform the models that excluded next-sentence-prediction (SciNCL, XLNet, RoBERTa, and ELECTRA). We attribute the positive effect of next-sentence-prediction to its similarity to our task since both are sequence pair classification tasks. Table 7.2 and 7.3 show variance among labels and both datasets. The larger number of training samples in CORD-19 (36%) may have contributed to higher performance in comparison to ACL Anthology. An unbalanced class distribution and different challenges of the aspects cause the performance to differ between the aspect classes. The high F1 scores of above 0.9 for negative samples are expected since the *None* aspect class is essentially an aspect-free similarity or citation prediction problem. Transformer models have been shown to perform well in these two problems (Cohan et al., 2020; Reimers and Gurevych, 2019). Besides the unbalanced distribution of training samples, we attribute the differences among positive aspect classes to their ambiguity and to the different challenges posed by the aspect classes. Authors often diverge when naming their section titles (e. g., *Results*, *Evaluation*), thus, increasing the challenge of classifying the different aspects of a paper. This also contributes to the high number of *Other* samples. Some sections are also content-wise more unique than others. An *Introduction* section usually contains different content than a *Results* section. The content difference makes some sections and the corresponding aspect classes easier to distinguish and predict than others. We suspect the poor performance for *Future work* is due to little or no information about them in the titles or abstracts.

Our main research objective in this chapter is to explore methods that are capable of incorporating aspect information into the traditional similar-dissimilar classification. In this regard, the results are successful. In particular, the micro-F1 score of 0.86 of SciBERT for the CORD-19 dataset is suitable for a recommender system. Our qualitative evaluation indicates that SciBERT's predictions can correctly identify similar aspects of the selected research papers.

Furthermore, we observe that aspect classes with little training data performed poorly. For example, *Conclusion* and *Discussion* have a zero F1-score for ACL Anthology whereas for the larger CORD-19 dataset *Discussion* yields 0.636 F1. We anticipate that more training data will lead to more correct predictions.

7.4 Summary of the Chapter

In this chapter, we continued the work on Research Task **III** and applied pairwise multi-label multi-class document classification on scientific papers to compute aspect-based document similarity scores. We used section titles as aspects of papers and labeled citations occurring in these sections accordingly. The investigated models were trained to predict citations and the aspect-based similarity based on the paper's title and abstract. We evaluated the Transformer models BERT, Covid-BERT, SciBERT, SciNCL, ELECTRA, RoBERTa, and XLNet and an LSTM baseline over two scientific corpora, i.e., ACL Anthology and CORD-19. Overall, SciBERT performed best in our experiments. Despite the challenging task, SciBERT predicted the aspect-

based document similarity with F1 scores of up to 0.83. SciBERT's successful results provide already a value for a research paper recommender system.

Before integrating the aspect-based document similarity into a recommender system, there are technical challenges to be solved. The pairwise classification as performed in this experiment is computationally expensive. We have used Transformer language models that require specific hardware such as GPUs. The computational less-expensive method, the LSTM baseline, yielded poor results making it not a valid alternative to Transformers. Also, pairwise classification is limited to a small corpus size since all possible document pairs would be need to classified. In the next chapter, we will address these issues and propose a method for aspect-based document similarity that scales to large document corpora.

Chapter 8

Specialized Research Paper Representations

The last two chapters introduced the pairwise document classification approach for Wikipedia articles (Chapter 6) and research papers (Chapter 7). Despite its high accuracy, the pairwise classification approach has a quadratic complexity making it computationally expensive even for small document corpora. However, aspect-based similarity measures should scale to large document corpora, as defined in Research Task **IV**. To improve the efficiency, this chapter revisits the use case of research papers from the previous chapter but formulates aspect-based similarity as a representation learning problem of aspect-specific document embeddings. This makes aspect-based similarity scale linearly with respect to the corpus size. The chapter's content is based on Ostendorff et al. (2022a).

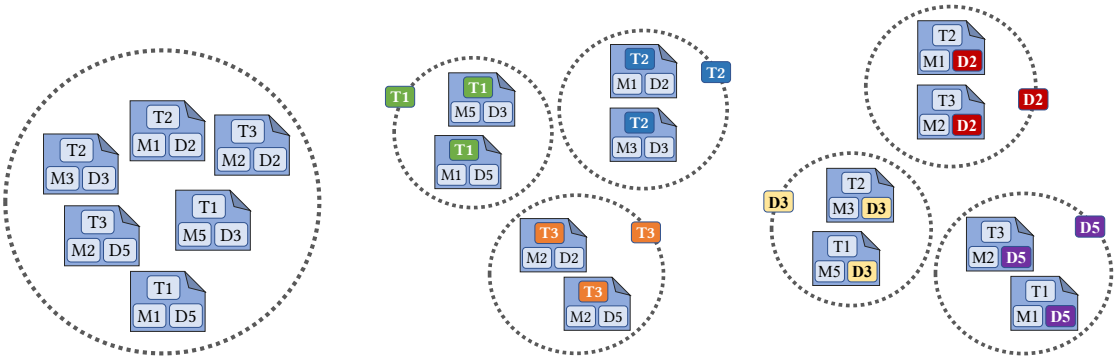
"Specialized Document Embeddings for Aspect-based Similarity of Research Papers" by **Malte Ostendorff**, Till Blume, Terry Ruas, Bela Gipp, and Georg Rehm. In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2022.

In content-based recommender systems and other information retrieval applications, the retrieval of semantically similar documents is often performed based on document embeddings that can be derived from the text (Devlin et al., 2019; Le and Mikolov, 2014), citations or links (Han et al., 2018; Tang et al., 2015), and combinations of text and citations (Cohan et al., 2020; Ostendorff et al., 2022b). The similarity between documents is then calculated based on the similarity of their vector representations, e. g., with cosine similarity (Ellis et al., 1993; Salton, 1963). Existing approaches represent a document with a single monolithic vector in the embedding space.

The single point representation follows the geometric understanding of similarity (Blough, 2001). As a result, it entangles the many aspects or meanings that a document can represent in a single measurement and makes the aspects indiscriminative (Camacho-Collados and Pilehvar, 2018). Consequently, it also leads to a single and generic notion of document similarity, which neglects the many aspects represented within a document. In the context of word embeddings, Camacho-Collados and Pilehvar (2018) coined "the inability to discriminate among different meanings of a word" as the meaning conflation deficiency. While the appearance of contextualized word embeddings has solved the meaning conflation for words (Peters et al., 2018; Vaswani et al., 2017), document embeddings still suffer from this issue. Given the length and different aspects covered by documents such as research papers, the meaning conflation deficiency is even more prevalent at the document level. Since single generic representations conflate individual aspects, similarity measures derived from them are *aspect-free*.

As discussed in Chapter 7, the similarity of research papers is often concerned with multiple aspects of the presented research, e. g., methods or findings (Chan et al., 2018). Addressing these aspects individually enables recommendations tailored for specific information needs and increases their diversity (Ge et al., 2010; Kunaver and Požrl, 2017). Especially in the scientific domain, this can help burst filter bubbles or facilitate new discoveries (Narechania et al., 2022; Portenoy et al., 2021).

In Chapter 6 and 7, we have demonstrated how aspect-based document similarity can be achieved through a pairwise multi-class document classification approach. However, with $\mathcal{O}(n^2)$ comparisons for a corpus of n documents, the pairwise multi-class classification approach scales poorly to large-scale corpora. A quadratic complexity requires extensive computational resources, in particular in combination with other computationally expensive methods, e. g., large Transformer language models (Section 2.3.7.1).



(a) Generic embeddings **(b)** *Task*-specific embeddings **(c)** *Dataset*-specific embeddings

Figure 8.1: Papers are associated with tasks (T), methods (M), and datasets (D). With generic embeddings (a), the k-nearest neighbors are papers similar in any aspect. Specializing the embeddings for the *task* aspect (b) lets papers with the same task be close to each other in the embedding space. In the *dataset*-specific embedding paper (c), the papers with the same dataset are close to each other.

In this chapter, we present specialized representations as an alternative approach to aspect-based document similarity. We propose to represent a document using multiple specialized embeddings – one embedding for each aspect. We learn an aspect-specific embedding space for each aspect. Thus, we can capture the similarity of documents regarding different aspects. We build upon the idea of disentangled representation learning (Section 2.5.4) and specialization (sometimes referred to as retrofitting) of word embeddings (Faruqui et al., 2015; Glavaš and Vulić, 2018). According to Higgins et al. (2018), disentangled representations are characterized by "the decomposition of a vector space into independent subspaces". The decomposition is typically done in an unsupervised setting and without making the semantic meaning of the subspaces explicit. These subspaces are closely related to the conceptual spaces from Gardenfors (2004).

In the context of word embeddings, specialization models leverage external lexical knowledge and other constraints, e. g., vectors of synonyms should be close to each other. The use of multi-sense embeddings to better represent the different meanings of words is known to improve natural language understanding related tasks (Li and Jurafsky, 2015; Pilehvar and Collier, 2016; Ruas et al., 2020; Ruas et al., 2019; Wahle et al., 2021). We apply the idea of disentanglement and specialization to documents and for each aspect-specific embedding space. Our goal is to leverage aspect information such that documents similar in a particular aspect are close to each other in the embedding space for that aspect (Figure 8.1). Thus, we refer to these embeddings as *specialized* for a specific aspect in contrast to *generic* embeddings that only reflect one unspecified aspect or view of a document.

The specialization approach keeps the documents intact as opposed to segmentation approaches (Chan et al., 2018; Huang et al., 2020; Kobayashi et al., 2018). More importantly, the approach

addresses the scalability issues of pairwise document classification (Chapter 6 and 7). The computationally expensive encoding of aspect information is only performed once per document and aspect. Retrieving similar documents can be done through a k nearest neighbor search in each aspect-specific embedding space. As a result, the approach has linear complexity, i. e., $\mathcal{O}(n)$ w.r.t. to n documents in the corpus. But it is unclear how the improved scalability affects the recommendation performance.

Thus, this chapter seeks to answer the following two research questions:

Research questions

RQ1: Can specialized document representations measure the aspect-based similarity more efficiently than the pairwise classification approach but without loss in quality? If so, what specialization method yields the best results?

RQ2: What do the specialized document representations reveal about generic unspecialized representations?

In the experiments, we evaluate our approach of specialized document representations on a content-based recommendation task using the Papers with Code[1] corpus. Research papers in Papers with Code are labeled with three aspects: the papers' *task*, the applied *method*, and the used *dataset*. We use these labels as aspects to specialize the embeddings of the research papers. In contrast to the citation dataset from Chapter 7, Papers with Code can be considered as a true gold standard since the aspect annotations are manually created. As specialization methods, we rely on existing methods but apply them in a way diverging from their original purpose. Namely, we evaluate retrofitting (Glavaš and Vulić, 2018) and jointly learned embeddings from Transformer fine-tuning (Beltagy et al., 2019; Cohan et al., 2020) and Siamese Transformers (Reimers and Gurevych, 2019). The specialized embeddings are compared against a pairwise multi-class document classification baseline and generic (non-specialized) embeddings from FastText word vectors (Bojanowski et al., 2017), SciBERT (Beltagy et al., 2019), SPECTER (Cohan et al., 2020), and SciNCL (Chapter 5).

In summary, this chapter's main contributions are:

1. We propose a representation learning approach to aspect-based document similarity using specialization methods. As opposed to pairwise document classification, we treat aspect-based similarity as a classical vector similarity problem in aspect-specific embedding spaces, which improves the scalability.
2. We empirically evaluate three specialization methods for three aspects on a newly constructed dataset based on Papers with Code for the use case of research paper recommendations. In our experiment, specialized embeddings improved the results in all three aspects, i. e., *task*, *method*, and *dataset*.
3. We find that recommendations solely based on generic embeddings had an implicit bias towards the *dataset* and against the *method* aspect.
4. We demonstrate the practical use of our approach in a prototypical recommender system.[2]

[1] `https://paperswithcode.com/`, last accessed: 18/01/2023
[2] `https://recsys.ostendorff.org/`, last accessed: 18/01/2023

The chapter's code, dataset, and models are publicly available.[3]

The remainder of this chapter is structured as follows: First, we introduce the general methodology, i.e., the datasets and the evaluated methods. Subsequently, we present the overall results in Section 8.2.2, the overlap of recommendation sets in Section 8.2.3, and the manual sample analysis in Section 8.2.4. In Section 8.3, we discuss the results of all evaluations. Finally, we summarize the main findings of this chapter.

8.1 Methodology

Figure 8.1 illustrates the specialization approach. It consists of two major components: (1) aspect information for a defined set of aspects $A = \{a_0, \ldots, a_n\}$, and (2) a specialization method that derives for any document d_i in the corpus D a set of n specialized embeddings $\boldsymbol{d}_i^{(a_j)}$ for each specific aspect a_j with $0 \leq j \leq n$. The aspect information is given in the form of triples $(d_a, d_b, y^{(a_j)})$ where the label $y^{(a_j)} = \{0, 1\}$ holds the binary information whether d_a and d_b are similar or dissimilar in aspect a_j. The training objective of the specialization method is to maximize the similarity of the embeddings of those document pairs (d_a, d_b) with $y^{(a_j)} = 1$, which are the ones that are similar in aspect a_j.

We distinguish between *specialized* embeddings and *generic* embeddings. Generic embeddings can be considered aspect-free, i. e., $\boldsymbol{d}_i^{(a_1)} = \boldsymbol{d}_i^{(a_2)} = \boldsymbol{d}_i^{(a_n)}$. *Specialized* or *generic* similar documents are retrieved through a k nearest neighbors search using the cosine similarity of the document embeddings. We evaluate our approach in the context of content-based recommender systems. Therefore, we refer to the results of the nearest neighbor search as *specialized* or *generic* recommendations.

With this approach, we treat aspect-based similarity as a classical vector similarity problem in aspect-specific embedding spaces. As a result, similar documents can be more efficiently retrieved as in the pairwise classification approach (Chapter 6 and 7). Pairwise classification requires the classification of all document pairs, i. e., a corpus with $|D|$ documents is equivalent to $\frac{|D|*(|D|-1)}{2}$ classifications. Thus, the pairwise classification approach has a quadratic complexity, i. e., $\mathcal{O}(|D|^2)$ w.r.t. the number of documents $|D|$. This quadratic complexity makes the computation infeasible even for a medium-sized corpus, in particular when Transformers are used for each classification. Our approach computes for each document $d \in D$ and each aspect $a \in A$ one specialized document embedding $\boldsymbol{d}^{(a)}$. Consequently, only $|D| * |A|$ Transformer forward passes are sufficient during inference. Thus, our approach scales linearly w.r.t. the number of documents $|D|$. Retrieving the k most similar documents can be done efficiently in the vector space using cosine similarity (Section 2.2.4). For larger corpora, an approximate nearest neighbor search could be used (Aumüller et al., 2017).

8.1.1 Dataset

Our approach requires information about aspects that make a document pair similar. To the best of our knowledge, no appropriate dataset for the problem of aspect-based similarity is publicly available as they lack either quantity or quality. For instance, the dataset provided by Chan et al. (2018) is too small in size for a machine learning approach. In Chapter 7, we rely on citations

[3] `https://github.com/malteos/aspect-document-embeddings`, last accessed: 18/01/2023

and section titles as a training signal. The citations have the advantage of being available for all fields of science. However, section titles are inconsistently used and, therefore, prevent a clear distinction among aspects.

Papers with Code provides a hand-curated collection of research papers in the machine learning domain (Kardas et al., 2020). In addition to metadata on authors or bibliography, each research paper is labeled with the *task* a paper is focusing on, the papers' *method*, and the *dataset* used. We exploit these labels as aspects, $A = \{task, method, dataset\}$. These aspects address different information needs that are beneficial for research paper recommender systems and are comparable to the aspects used in related work (Chan et al., 2018). For example, *Beltagy et al. (2019)* and *Cohan et al. (2020)* are labeled with *BERT (Devlin et al., 2019)* as their *method*. Thus, we consider the pair of *Beltagy et al. (2019)* and *Cohan et al. (2020)* as similar regarding the *method* aspect. Other aspect labels are for example:

- **Tasks:** Low-Rank Matrix Completion, Q-Learning, Quantization, Speaker Recognition, Object Detection
- **Methods:** Residual Connection, Tanh Activation, Multi-Head Attention, LSTM, Transformer
- **Datasets:** Atari 2600 Atlantis, Cityscapes, SOP, MS MARCO, Labeled Faces in the Wild

We use the Papers with Code dataset as our gold standard that contains 157,606 papers in total.[4]

Table 8.1: Dataset statistics for each aspect

Aspect	Papers	Labels	Avg. papers per label
Task	154,350	1,421	17.9
Method	108,687	788	12.4
Dataset	37,604	1,743	5.6

For each aspect, we construct separate datasets containing positive and negative samples. Positive samples are unique unordered paper pairs with the same label, i. e., $y = 1$. For each label, the number of pairs is $\frac{L}{2}$ where L is the number of papers per label. Negative samples are randomly sampled paper pairs without the same label, i. e., $y = 0$. A more elaborated sampling strategy similar to the one from Chapter 5 was omitted for simplicity. The number of negative samples is 50% of the number of positive samples.

Some labels are too frequent in the corpus, e. g., the *method* label *Softmax* is assigned to 5,324 papers. To ensure the specificity of aspect information, we discard all labels which are assigned to more than 100 papers. The removal of too frequent labels increases the task's difficulty and ensures an appropriate dataset size. The dataset would become too large otherwise, e. g., *Softmax* alone would account for 1.2M paper pairs.

The experiments are conducted as a 4-fold cross-validation, i.e., we split the data into 75% training and 25% test papers. The resulting ground truth consists of 1,227,058 *task*, 284,193 *method*, and 58,984 *dataset* paper pairs.

[4]Downloaded on Oct 27th, 2020

In summary, Papers with Code enables us to evaluate aspect-based similarity for research papers based on a curated dataset. The aspect labels from Papers with Code have a higher quality compared to the citation dataset from Chapter 7. At the same time, Papers with Code contains only machine learning papers while the citation dataset provides a broader coverage across scientific domains. Therefore, both datasets complement each other.

8.1.2 Evaluated Methods

We evaluate the document embeddings from three base models and three specialization methods. Besides the aspect information (Section 8.1.1), each method utilizes the title and abstract to generate the embeddings. We distinguish between generic and specialization methods, where the latter is divided into two categories: retrofitted and jointly learned embeddings. Source codes, trained models, and instructions to reproduce our work are publicly available[3].

8.1.2.1 Generic Embeddings

We use *generic* document embeddings that do not leverage any aspect information. As base models, we rely on averaged FastText word vectors as document embeddings (Bojanowski et al., 2017), document embeddings from SciBERT (Beltagy et al., 2019)[5], SPECTER (Cohan et al., 2020), and SciNCL (Chapter 5). The latter three are BERT-inspired models (Devlin et al., 2019) pretrained on scientific literature. In contrast to SciBERT, SPECTER and SciNCL apply additional contrastive fine-tuning based on citations. SciBERT, SPECTER, and SciNCL are used as published by their authors without any fine-tuning on our corpus and in their `BASE`-version.

8.1.2.2 Retrofitted Embeddings

Retrofitting refers to the postprocessing of existing embeddings such that they fit predefined constraints (Faruqui et al., 2015). In the context of word embeddings, synonyms or antonyms are typically used as constraints and define which vectors should be close or apart. For our experiments, we use the aspect labels from Papers with Code as constraints. We retrofit all generic embeddings with Explicit Retrofitting (ER) as proposed by Glavaš and Vulić (2018). As opposed to other retrofitting methods such as the one from Faruqui et al. (2015), ER generalizes to unseen vectors for which no predefined constraints exist. An ER model can be learned on a subset for which constraints exist (training set) and, then, be applied to all remaining embeddings (test set). The training constraints are the positive samples in the same fashion, in which the synonyms are used in the retrofitting of words.

8.1.2.3 Jointly Learned Embeddings

We refer to this category as jointly learned embeddings since the aspect information is integrated into the representation learning process. Aspect-based embeddings are directly generated from textual input (title and abstract of a paper). We fine-tune SPECTER, SciNCL, and SciBERT in a sequence-pair setup on positive and negative samples from our training set. The input is a pair of two papers separated with a `[SEP]`-token. The sequence pair is subject to a binary classification

[5]For SciBERT, we apply mean-pooling, i. e., a document vector is the mean of the hidden-states of the last layer of the SciBERT model. Documents embeddings from the `[CLS]`-token yielded significantly lower results, e. g., 0.001 MAP for the *task* aspect).

(similar in the current aspect or not). To derive embeddings for the test set, we use only a single paper as input to the fine-tuned version of SPECTER, SciNCL, and SciBERT.

Aside from the sequence pair fine-tuning, we also test a Siamese network based on three Transformer models; see Sentence-BERT from Reimers and Gurevych (2019). Siamese-SciBERT, Siamese-SPECTER, and Siamese-SciNCL use a Siamese architecture (Bromley et al., 1993), in which the paper pair is separately fed as an input and then used in the loss function.[6]

In summary, our experiments use the four generic embeddings from FastText, SciBERT, SPECTER, and SciNCL (see Section 8.1.2.1). As specialization methods, we retrofit the four generic embeddings, and also jointly learn specialized embeddings with Transformer fine-tuning and Siamese Transformers (see Section 8.1.2.2 and 8.1.2.3). Furthermore, we use the pairwise classification approach as a baseline:

8.1.2.4 Baseline

We train a pairwise classification model based on the SciNCL model according to the methodology from Chapter 7. We selected SciNCL over SciBERT and SPECTER since SciNCL's generic version outperformed the other two models. With a document pair as input, the model predicts the probability distribution over the aspect labels.

The pairwise approach is not directly applicable to our dataset as its quadratic complexity would require the classification of 1.3 billion document pairs. To reduce the number of candidate pairs, we first retrieve the $n = 300$ nearest neighbors d_n for any seed document d_s based on the generic SciNCL embeddings. The pairs of seed and neighbor documents (d_s, d_n) are selected as candidates for the classifier. This candidate filtering reduces the number of classifications to 11.3 million document pairs.

8.1.3 Evaluation Methodology

Each of the n aspects is evaluated separately (n train, n test sets). All documents from the test set are used as seeds. For a given aspect a_j and the vector $\boldsymbol{d}_s^{(a_j)}$ of seed d_s, we retrieve k candidate documents, with a k nearest neighbor search (Cover and Hart, 1967). The similarity of documents is computed as the cosine similarity of their vectors. The only exception is the pairwise baseline approach, for which the predicted class probabilities are used instead of cosine similarity. A candidate document d_c is relevant for the seed d_s if they are associated with the same label for aspect a_j, i. e., $(d_s, d_c, y^{(a_j)} = 1)$ is part of the ground truth. We compute precision, recall, mean average precision, and mean reciprocal rank based on this relevance definition (Section 2.1.3).

8.2 Evaluation

This section presents the experimental results, starting with the pairwise baseline. Subsequently, aspect-based similarity methods, generic and specialized embeddings are compared.

[6]For Siamese-Transformers, we experimented with different losses and found the Multiple Negative Ranking Loss (Henderson et al., 2017), with only positive samples from the train set, yielding the best results.

Table 8.2: Classification results for Pairwise SciNCL.

Aspect ↓ / Metric →	Precision	Recall	F1-Score
Task	0.88	0.81	0.84
Method	0.56	0.45	0.50
Dataset	0.10	0.33	0.16
Micro Avg.	0.79	0.74	0.76
Macro Avg.	0.51	0.53	0.50

8.2.1 Pairwise Results

For the pairwise approach, we first need to train a classification model that can be separately evaluated on the test set. Table 8.2 shows the classification performance of Pairwise SciNCL in terms of precision, recall, and F1-score. With a micro F1 score of 0.76, the performance is comparable to the experiments from Chapter 7. A performance discrepancy can be seen between the aspect classes. For *task* the F1-scores are the highest with 0.84, followed by *method* with 0.50. The worst performance yields the *dataset* aspect with only 0.16 F1.

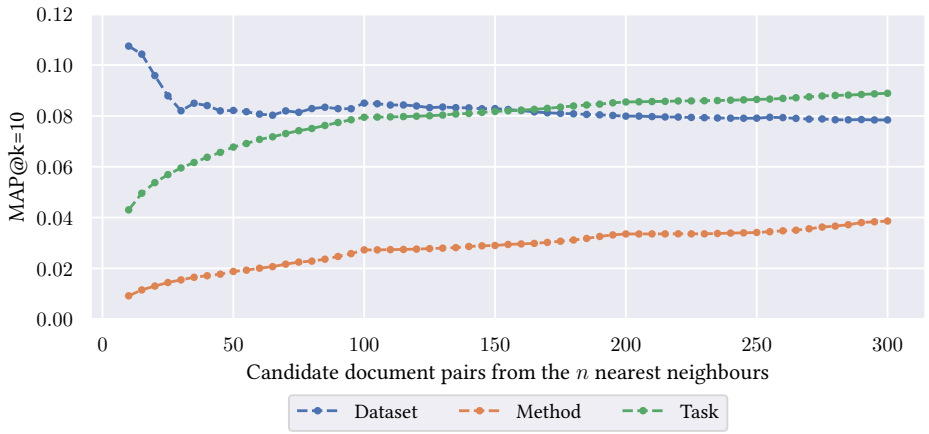


Figure 8.2: Results of the pairwise SciNCL baseline in terms of MAP@k=10 depending on the candidate filtering for different n nearest neighbors.

To make the pairwise approach applicable to our dataset, we introduced an artificial constraint since the prediction for all document pairs is not possible due to the quadratic complexity and limited resources. We retrieve the $n = 300$ nearest neighbors based on generic SciNCL to filter for candidate pairs for that we predict the aspect labels. As this constraint potentially harms the performance, we plot Pairwise SciNCL's performance as MAP@k=10 depending on the size of the n nearest neighbor filter in Figure 8.2. The performance generally increases as n increases. However, performance gains are getting smaller for larger n. The high MAP for the *dataset* aspect and small n is due to the good performance of generic SciNCL for this aspect.

Table 8.3: Overall results for generic and aspect-specific representations and the pairwise SciNCL baseline. Precision, recall, and mean average precision (MAP) are reported as average over a 4-cross-validation and for each aspect and as average over the aspects. The highest score among aspects in each metric is underlined for the individual method, and **bold** shows the highest score among methods for a single metric. Siamese SciNCL yields the best result.

Aspects →		*Task*			*Method*			*Dataset*			*Avg.*
	Methods ↓	**Prec.**	**Rec.**	**MAP**	**Prec.**	**Rec.**	**MAP**	**Prec.**	**Rec.**	**MAP**	**MAP**
Pairwise SciNCL		0.289	0.110	0.089	0.152	0.048	0.039	0.133	0.126	0.078	0.069
Generic emb.	FastText	0.208	0.071	0.046	0.096	0.029	0.016	0.170	0.260	0.152	0.071
	SciBERT	0.083	0.027	0.015	0.044	0.012	0.006	0.079	0.112	0.059	0.026
	SPECTER	0.241	0.084	0.056	0.077	0.023	0.011	0.214	0.325	0.195	0.087
	SciNCL	0.268	0.093	0.065	0.081	0.023	0.012	0.227	0.355	0.219	0.099
Specialized embeddings	Retrofitted FastText	0.233	0.081	0.054	0.133	0.040	0.024	0.202	0.290	0.174	0.084
	Retrofitted SciBERT	0.106	0.035	0.019	0.067	0.018	0.009	0.103	0.140	0.073	0.034
	Retrofitted SPECTER	0.263	0.093	0.064	0.099	0.029	0.015	0.225	0.337	0.205	0.095
	Retrofitted SciNCL	0.285	0.101	0.071	0.101	0.029	0.016	0.240	0.367	0.227	0.105
	Fine-tuned SciBERT	0.091	0.031	0.020	0.052	0.013	0.007	0.070	0.088	0.045	0.024
	Fine-tuned SPECTER	0.098	0.032	0.021	0.088	0.028	0.018	0.077	0.096	0.052	0.030
	Fine-tuned SciNCL	0.084	0.027	0.017	0.089	0.027	0.017	0.107	0.138	0.083	0.039
	Siamese SciBERT	0.569	0.242	0.224	**0.407**	**0.168**	**0.137**	0.270	0.374	0.235	0.199
	Siamese SPECTER	**0.571**	**0.244**	**0.227**	0.402	0.162	0.134	0.262	0.371	0.229	0.196
	Siamese SciNCL	0.567	0.241	0.222	0.402	0.166	0.135	**0.280**	**0.390**	**0.244**	**0.200**

8.2.2 Overall Results

Table 8.3 presents the overall results based on the most $k = 10$ similar documents from each method. Results for other k values are depicted in Figure 8.3. In the following, unless stated otherwise, we refer to the MAP scores since we consider MAP as our primary evaluation metric since it takes the rank of multiple relevant candidates into account.

Siamese SciNCL is the best method in terms of average MAP scores over all three aspects. In general, all three Siamese methods (also including Siamese SciBERT and Siamese SPECTER) outperform the other methods for all metrics and aspects by a large margin. Between the Siamese methods, the performance differences are insignificant. Among the generic embeddings, SciNCL is the best method closely followed by SPECTER and FastText. For *task* and *dataset*, the generic SciNCL and SPECTER outperform FastText, while for *method* the opposite is the case. SciBERT yields the lowest scores in the generic category. As our experiments in Chapter 4 and 5 showed, BERT-based embeddings perform poorly without task-specific fine-tuning. Even the computationally less complex FastText outperforms SciBERT. Despite requiring the largest computational effort, the Pairwise SciNCL baseline yields generally poor results especially compared to the Siamese Transformers.

The ER retrofitting approach from Glavaš and Vulić (2018) has a small but positive effect on the performance. For FastText and SciBERT, the retrofitting increases all scores (on average +26%

MAP for FastText, +34% MAP for SciBERT), while for SciNCL and SPECTER retrofitting has an even smaller effect on the performance. The fine-tuning of SciNCL, SPECTER, and SciBERT has a different effect depending on the aspects. Compared to its generic counterpart, fine-tuned SPECTER's MAP score is 25% higher for the *task* aspect but 57% lower for the *dataset* aspect. For SciBERT, the fine-tuning also decreases its MAP score by 23% for the *dataset* aspect.

Furthermore, we do not only see performance differences between the methods but also between the aspects. All methods yield the highest precision for *task*, whereas recall and MAP are the highest for *dataset*. The poor *method* results can be partially attributed to the unbalanced distribution of the aspects (Section 8.1.1). Most samples are available for *task*, explaining its good performance compared to *method*. However, *dataset* has the least number of samples but still outperforms *method*. When we specialize the document embeddings, we also notice a decrease in performance differences between the aspects. While SciNCL has a high MAP difference from *dataset* to *method* (94%) and from *dataset* to *task* (70%) , the same difference is lower for Siamese SciNCL (44% and 9% respectively). The better the specialization effect the lower the performance gap between aspects.

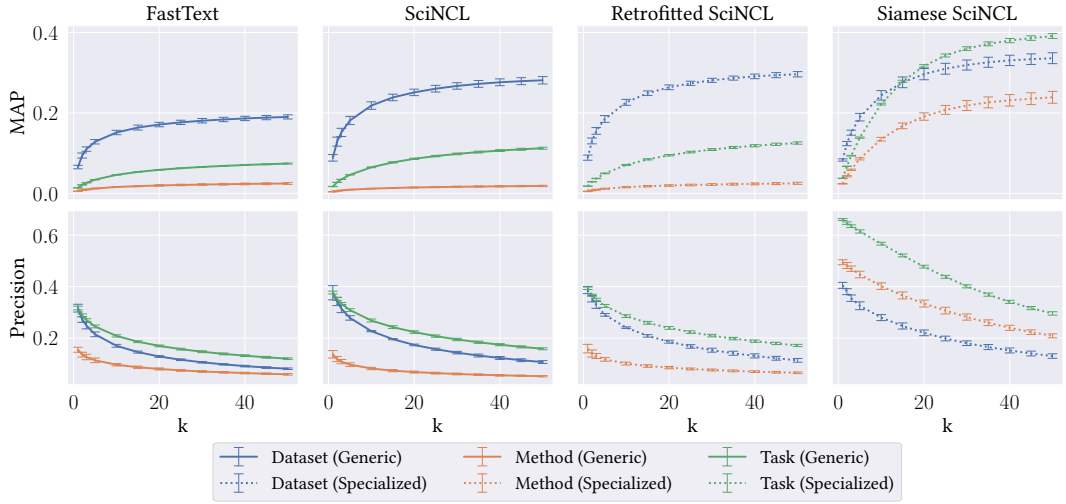


Figure 8.3: Precision and MAP@k for two generic (FastText and SciNCL) and two specialized embeddings (Retrofitted and Siamese SciNCL). For generic embeddings, each line presents the scores of the generic method evaluated on different aspect datasets. For specialized embeddings, a line presents a separately trained model. Generic embeddings and retrofitted SciNCL yield similar results on different k and aspects, while for Siamese SciNCL, the *task* aspect yields a higher MAP compared to *dataset* for $k > 15$.

To analyze the aspect-specific performance, Figure 8.3 depicts the performance ranking as MAP and precision for different k values for FastText, SciNCL, Retrofitted SciNCL, and Siamese SciNCL. The performance among the aspect remains stable independent of k for all methods, except Siamese SciNCL. With Siamese SciNCL, the *task* aspect yields a higher MAP than *dataset* for $k > 15$. In terms of precision, Siamese SciNCL is another exception since the precision of *method* is higher than in *dataset*. For all other methods, *method* has the lowest precision.

In summary, Siamese SciNCL achieves on average the best results. Thus, we consider SciNCL in combination with the Siamese specialization as the best method to handle specialized embeddings even outperforming the Pairwise SciNCL baseline.

8.2.3 Overlap of Recommendation Sets

The performance discrepancy among the aspects could indicate a systematic difference between the documents recommended through the similarity of generic embeddings and the specialized ones. Therefore, we conduct an additional experiment with overlapping recommendation sets.

We use the trained models from Table 8.3 but infer vectors for all documents in the whole corpus. Then, we retrieve $k = 50$ recommendations and measure the overlap between each method's nearest neighbors on a seed level. The large k value is selected to increase the chance of overlapping recommended documents. Table 8.4 presents the intersection ratio between the generic recommended documents from FastText and SciNCL and the specialized ones from Siamese SciNCL. For the remaining methods, we report the intersection in the supplemental materials[3]. The lower the overlap, the more distinct the recommendations are from each other.

Table 8.4: Intersection of $k = 50$ recommendations from A and B. Most overlap between generic methods (FastText and SciNCL) and between generic SciNCL and Siamese SciNCL's *dataset* recommendations. Only 7% of Siamese SciNCL's *method* recommendations are also retrieved by generic SciNCL.

Recommendations A	Recommendations B	A∩B
FastText	SciNCL	0.20
	Siamese SciNCL$^{\text{Dataset}}$	0.15
	Siamese SciNCL$^{\text{Method}}$	0.06
	Siamese SciNCL$^{\text{Task}}$	0.10
SciNCL	FastText	0.20
	Siamese SciNCL$^{\text{Dataset}}$	0.21
	Siamese SciNCL$^{\text{Method}}$	0.07
	Siamese SciNCL$^{\text{Task}}$	0.15

On the one hand, a substantial overlap can be found between the two generic recommendations from FastText and SciNCL. This suggests little difference within the generic recommended documents. An even larger overlap can be found between generic SciNCL and Siamese SciNCL's *dataset* recommendations. On the other hand, Siamese SciNCL's *method*-specific recommendations overlap the least with the generic ones. The discrepancy among the aspects is significant. Compared to SciNCL, Siamese SciNCL has an overlap of 15%, 7%, and 21% for *task*, *method*, and *dataset* respectively. Thus, indicating *dataset*-specific recommendations are overrepresented in generic recommendations, while *method*-specific ones are underrepresented.

Furthermore, the overlap between the aspect-specific recommendations is also low. The *method*-specific recommendations have only an overlap of 5% with *dataset* and 4% with *task*. This low overlap between the recommendations of individual aspects can be used to diversify the

recommendations by mixing recommendations across aspects. For example, we observed that selecting the top $k = 1$ recommendations from the three aspects instead of the top $k = 3$ generic recommendations increased the coverage by up to 5.8% (from 81.9 to 86.7).

8.2.4 Manual Sample Analysis

To verify the quantitative findings, we also qualitatively analyze the generic and specialized embedding spaces (Section 8.2.4.1) and the recommendations generated from the respective embeddings (Section 8.2.4.2).

8.2.4.1 Embedding Space Analysis

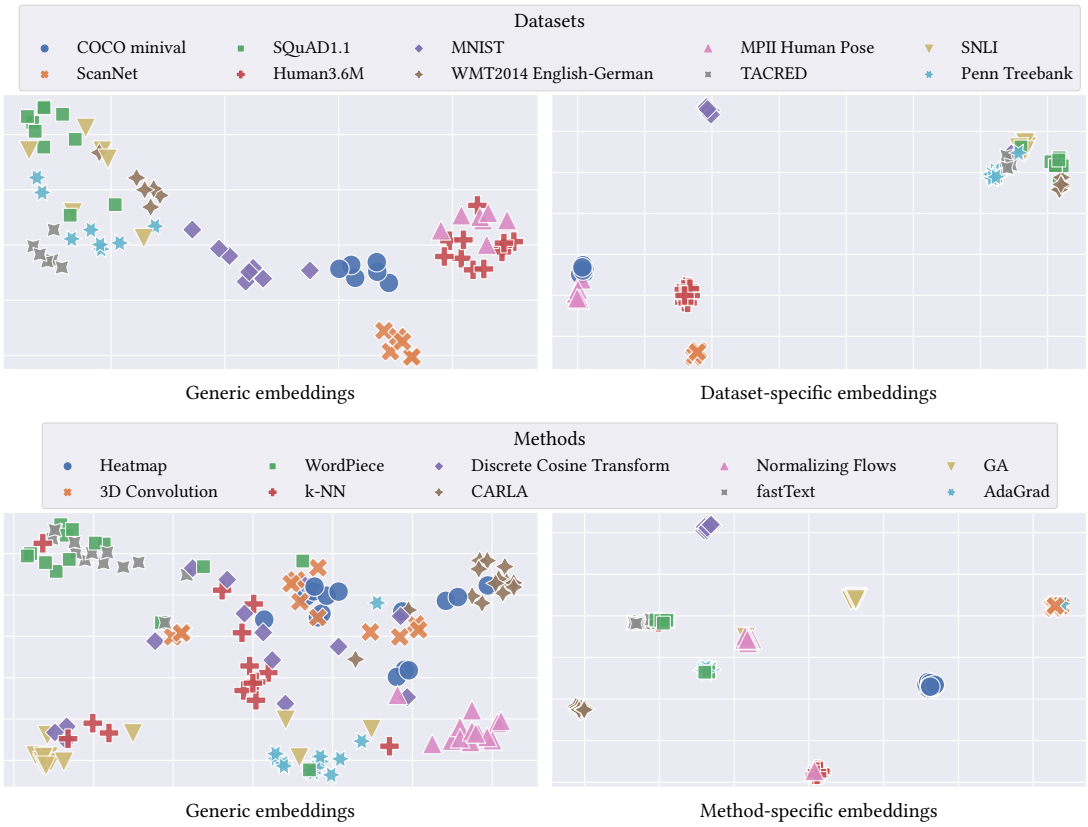


Figure 8.4: Visualization of embedding spaces reduced to two dimensions using UMAP. The separation between aspect labels is much clearer with aspect-specific embeddings (right column) compared to the generic ones (left column). The generic embeddings plotted according to their *dataset* aspect (top-left plot) have a better separation than the generic embeddings with *method* labels (bottom-left plot).

In addition to the quantitative evaluation, we visually inspect the embedding spaces from generic SciNCL and specialized Siamese SciNCL. For this purpose, we reduce the dimensionality of the paper embeddings from 768 to two dimensions using UMAP (McInnes et al., 2018). Figure 8.4 visualizes the paper embeddings of a subset of our corpus. Most papers in our corpus are associated with multiple aspect labels, e.g., more than one *dataset* or *method*. For better readability, the papers with multiple aspect labels are excluded from the plot, and only papers

with a single label are shown. This reduces the number of available papers but makes patterns in their embedding spaces more evident. We color all data points (e.g., the papers) according to their aspect labels.

While the two visualizations at the top of Figure 8.4 illustrate paper embeddings according to their *dataset*, the two bottom visualizations are about their *method*. In each row, the generic embeddings are in the left column and aspect-specific embeddings are in the right column, *dataset*-specific and *method*-specific respectively.

The comparison of generic and specialized embedding spaces in both rows demonstrates the specialization effect that we already found in the quantitative evaluation (see Table 8.3). The separation between aspect labels is much clearer with aspect-specific embeddings (right column) compared to the generic ones (left column). In contrast to aspect-specific embeddings, which produce distinct clusters, generic embeddings have many aspect labels scattered throughout the whole embedding space. This leads to papers being neighbors but having semantically dissimilar aspect labels, e.g., the dataset of MNIST and PennTreebank (top-left plot).

The visualizations of the embedding spaces also confirm the results from Table 8.4, as the aspects are differently reflected in the generic embeddings. The generic embeddings plotted according to their *dataset* aspect (top-left plot) have a better separation than the generic embeddings with *method* labels (bottom-left plot). Even though the separation is not as distinct as in the aspect-specific embeddings, the generic embeddings still produce separate regions for NLP datasets (e.g., SQuAD, SNLI, or Penn Treebank) and computer vision datasets (e.g., MNIST, ScanNet, or COCO). Also, datasets with related tasks are in close proximity (e.g., MPII Human Pose and Human3.6M). The *method* aspect is less reflected in the generic embeddings. For instance, the methods of k-NN or heatmap are scattered throughout the whole embedding space. We attribute this to the fact that these methods are general techniques applied in various contexts.

8.2.4.2 Recommendation Analysis

We also qualitatively analyze randomly sampled seed papers and their most similar documents in the context of research paper recommendations. Table 8.5 presents one of these samples with its top-$k = 3$ recommendations. Generic recommendations are taken from SciNCL and *task*-, *method*-, and *dataset*-specific ones from Siamese SciNCL. For other examples, we provide a Web-based demo to browse the recommendations for all papers from the dataset[2].

Gupta (2019) is the seed paper to which Papers with Code associates three *task* labels (*data augmentation*, *sentiment analysis*, *text generation*), two *method* labels (*convolution* and *generative models (GAN)*), and none *dataset* label. As the labels and the title suggest, *Gupta (2019)* uses generative adversarial networks as a data augmentation method to generate textual training data for the sentiment classification task. The four different recommendation sets illustrate the many aspects in that papers can be similar.

The first generic recommendation *(Zhu et al., 2017)* uses as the seed also GANs as an augmentation method and evaluates the related task of emotion classification. The second *(Monti et al., 2019)* and third *(Han et al., 2020)* generic recommendations do not have any obvious semantic connection to the seed paper but both use graph neural networks for fake news detection.

While the first *task*-specific recommendation *(Amram et al., 2018)* shares the task of sentiment analysis with the seed, the second *(Horne and Adali, 2017)* and third *(Farajtabar et al., 2017)*

Table 8.5: Example recommendations from SciNCL (generic) and Siamese SciNCL (aspect-specific) for the seed *"Data augmentation for low resource sentiment analysis using generative adversarial networks"* by Gupta (2019)

	Generic	Task	Method	Dataset
1	Data Augmentation in Emotion Classification Using Generative Adversarial Networks (Zhu et al., 2017)	Representations and Architectures in Neural Sentiment Analysis for Morphologically Rich Languages: A Case Study from Modern Hebrew (Amram et al., 2018)	Company classification using machine learning (Husmann et al., 2022)	From Image to Text in Sentiment Analysis via Regression and Deep Learning (Onita et al., 2019)
2	Fake News Detection on Social Media using Geometric Deep Learning (Monti et al., 2019)	This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News (Horne and Adali, 2017)	Fake News Mitigation via Point Process Based Intervention (Farajtabar et al., 2017)	Homogeneity-Based Transmissive Process to Model True and False News in Social Networks (Kim et al., 2019)
3	Graph Neural Networks with Continual Learning for Fake News Detection from Social Media (Han et al., 2020)	Fake News Mitigation via Point Process Based Intervention (Farajtabar et al., 2017)	Multi-agent Policy Optimization with Approximatively Synchronous Advantage Estimation (Wan et al., 2020)	Leveraging the Crowd to Detect and Reduce the Spread of Fake News and Misinformation (Kim et al., 2018)

task-specific recommendations are about the fake news detection task without any direct semantic similarity to the seed paper.

The *method*-specific recommendations *(Husmann et al., 2022)*, *(Farajtabar et al., 2017)*, and *(Wan et al., 2020)* are at first sight unrelated to the seed since they focus on unrelated topics such as company classification or fake news. Nonetheless, the seed and the first *method*-specific recommendation *(Husmann et al., 2022)* use t-distributed Stochastic Neighbor Embedding (t-SNE) for visualization. Despite of being different in central themes, the paper pairs have similar methodologies.

The similarity between the seed and the first *dataset*-specific recommendation *(Onita et al., 2019)* can be attributed to both being about sentiment analysis. The second *(Kim et al., 2019)* and third *(Kim et al., 2018) dataset*-specific recommendations have little similarity with the seed paper.

In summary, we consider all recommendations at the first rank as generally relevant since they share one or more aspects with the seed, whereas most of the recommendations at the second and third rank are only partially related. Due to the subjectiveness of relevance, a recommender system would need to relate the recommendations to its users' individual information needs. However, when new user data is unavailable, this is not feasible. This is a general problem of purely content-based recommendations.

Our example illustrates how different aspects can approximate similar research papers in a granular and more detailed perspective. The specialization from Siamese SciNCL also leads to diverse recommendations between aspect-specific recommendations and generic ones. SciNCL's generic recommendations have a relatively narrow focus. The *method*-specific recommendations even reveal the implicit shared use of the t-SNE visualization. Notably, many recommendations

are about fake news detection, which we would consider irrelevant to the seed. The *fake news* recommendations can be explained by the abstract of the seed paper in which the authors refer to "real data points". A similar wording can be also found in the abstracts of the *fake news* recommendations.

8.3 Discussion

Our experimental results reveal the effect of specialized document representations. The performance gains between the best generic and the best aspect-specific embeddings, i. e., generic SciNCL and Siamese SciNCL, are substantial. We anticipated this outcome as the generic embeddings are not optimized for this task compared to the specialized ones. Still, our findings do not mean generic embeddings lead to unrelated recommendations, but only that they are not similar concerning *task*, *aspects*, or *dataset*.

Pairwise baseline. Siamese SciNCL also outperforms the Pairwise SciNCL baseline. Pairwise SciNCL with an unbounded n would potentially yield better results than our restricted version. For instance, Reimers and Gurevych (2019) have showed that unrestricted Pairwise Transformers achieve better performances than Siamese Transformers. However, due to the quadratic complexity, we would have to perform 1.3 billion comparisons, which would take approximately 46 days on the hardware used in our experiments (GeForce RTX 2080 Ti with 11GB memory). Thus, the potential performance gains would not justify the increase in computational effort for most recommender system deployments.

Specialization performance. In terms of specialization, the Siamese Transformers (Siamese Network with SciBERT, SPECTER, or SciNCL) outperform retrofitting and non-Siamese Transformer fine-tuning. This outcome can be explained for several reasons. The ER retrofitting method from Glavaš and Vulić (2018) has been originally developed for words and optimized for the properties of a word embedding space. We see retrofitting having a larger effect on FastText compared to the Transformer models. The integration of citation information as done in SPECTER and SciNCL generally improves the performance of their generic and fine-tuned version compared to SciBERT. The poor performance of SciBERT is aligned with the results of our previous experiments (Chapter 4 and 5), which show that document embeddings from BERT-based models are suboptimal for the similarity search. Since we perform the similarity search based on static embeddings, each document needs to be independently encoded. While this is the case in the Siamese Transformers, the sequence pair fine-tuning uses a joint encoding of document pairs. As the results from Chapter 3 suggest, the joint encoding is superior for the pairwise document classification approach. However, our results show the opposite in the k nearest neighbor search setting. The independent encoding, as in the Siamese model, produces semantically similar document embeddings with higher precision and recall.

Given the overall results, we consider Siamese SciNCL as the best method to specialize the embeddings of research papers. Nevertheless, we ask ourselves if the specialization effect depends on individual aspects. The most positive specialization effect can be observed for the *method* aspect, while the effect is less significant for *dataset*. We partially attribute the discrepancy in the specialization effect to training data availability, e. g., more samples for *method* than *dataset*. However, the effect is also due to the aspects being differently inherent in generic embeddings' similarity.

Bias in generic embeddings. The similarity of generic embeddings does not explicitly contain aspect information, i. e., we cannot attribute the document similarity to a specific aspect in which documents are similar. However, we can assume the aspects are implicitly part of the similarity. Thus, the similarity of generic embeddings would be denoted as a weighted sum $\sum_{a \in A} w_a * s_a$, where $A = \{task, method, dataset, \dots a_n\}$ is a set of aspects consisting of our three and an arbitrary number of other aspects. If the similarity of generic embeddings would evenly incorporate all aspects, all weights w_a should be equal. Still, our experiments suggest the aspects are not equally weighted. Table 8.4 reports an uneven intersection ratio among the recommendations. The *method*-specific recommendations have less overlap with the generic recommendation than the *dataset* or *task*-specific recommendations. Given that *task* has the most samples in the ground truth, we would have expected a different outcome, e. g., more specialization concerning *task*. Therefore, $w_{method} < w_{task} < w_{dataset}$ likely holds true. Accordingly, the results indicate an implicit bias in the similarity of generic embeddings towards *dataset* and against *method*. Our qualitative analysis of the embedding spaces and sample recommendations does not reject this finding. We hypothesize the bias is more likely to be caused by the corpus' characteristics than by the embedding methods themselves. Title and abstract of papers prominently mention tasks and datasets, whereas methodological details are of marginal importance, e. g., the t-SNE visualization in our example from Table 8.5.

Implications for content-based recommender systems. Having this bias toward a single aspect indicates the generic embeddings present only a single view of the content of a document. Therefore, the conflation of meaning, which has been shown for word embeddings (Camacho-Collados and Pilehvar, 2018; Pilehvar and Collier, 2016), also exists for document embeddings. Consequently, a recommender system based on generic embeddings is limited in the information needs that the system can address. Namely, those information needs that match with the single aspect, which is the *dataset* aspect in our case. Such a narrow focus on one information need hurts the diversity of the recommendations. In the literature (Ge et al., 2010; Nguyen et al., 2014b), the lack of diversity has been identified as a major issue of today's recommender systems. By changing the approach of representing documents, from generic to specialized embeddings, diverse information needs can be addressed even when user data is sparse. In the context of recommendations, our results do not allow a decisive statement on the relevancy of the generic or aspect-based recommendations since we primarily evaluate the similarity of research papers. We use similarity only as an approximation of relevance for specific information needs, i. e., interest in the task, method, or dataset of the presented research. To the best of our knowledge, a dataset that would allow a relevance-based evaluation of the Papers with Code corpus is not publicly available. Thus, further experiments involving user feedback are required to investigate the relevancy of aspect-based recommendations. Nonetheless, the recommendations from specialized embeddings can expose the implicit bias within the generic recommendations. Integrating the aspect information can improve research paper recommender systems as users would decide in which particular aspect they are interested. As a result, tailored content-based recommendations are feasible even without user feedback. The aspect-based recommendations increase the transparency of a recommender system since the system can provide explicit explanations of the aspects to which documents are related. Such explanations can also strengthen the trust in the recommendations, as demonstrated by Kunkel et al. (2019). Furthermore, recommendation diversity and coverage can be improved through selection from multiple aspects, as our results showed. In a user interface, recommendations will not only be selected from a single aspect but

rather across multiple aspects, e. g., the top recommendation for *task*, *method*, and *dataset* (the items in the first row of Table 8.5; illustrated in Figure 9.1).

Scalability. Diversity and explainability are also covered by the pairwise multi-class classification approach. However, the pairwise approach bears scalability constraints that would prevent recommender systems to be deployed in practice. Pairwise document classification requires large computational resources even for medium-sized corpora since aspect information needs to be separately derived for all document pairs. To use the pairwise approach as a baseline, we introduced the candidate filtering but it still needs to perform 11.3M Transformer forward passes while achieving only a lower performance compared to Siamese SciNCL. Instead, our approach derives the aspect information during the encoding phase, which results in a linear time complexity (118,146 forward passes in our experiments). During the indexing of a new document, the system would only need to create n specialized embeddings instead of a single generic embedding. Thus, the complexity of this chapter's approach is mainly bound to the number of aspects and not to the size of the document corpus as in pairwise classification. As a result, our approach can be used in practice and is not limited to academic experiments. Our Web-based demo is one example of a prototypical recommender system based on specialized document embeddings[2].

Interpretability. Aside from scalability, the specialized embeddings have additional advantages such as explainability and interpretability. Each individual aspect-specific vector $\boldsymbol{d}_i^{(a_j)}$ could also be combined through concatenation into a single document vector $\boldsymbol{d}_i = [\boldsymbol{d}_i^{(a_1)}; \ldots; \boldsymbol{d}_i^{(a_n)}]$ for other downstream tasks. The aspect's dimensions could then facilitate the interpretability of the document vectors in a similar fashion as Liao et al. (2020a) already demonstrated with sparse vectors. In the context of words, related approaches already exist. For example, Schwarzenberg et al. (2019) project word vectors into a concept space in which the dimensions correspond to predefined concepts.

Alternative approaches. Lastly, the question is whether comparable recommendations are also possible with alternative approaches such as query-sensitive similarity (Tombros and Van Rijsbergen, 2001). One could filter papers by a query, i. e., their respective aspect labels, and then perform a nearest neighbor search on the filtered papers' generic embeddings. However, the filtering depends on hard label assignments, e. g., papers need to have an identical task, method, or dataset to be considered. Papers without an exact match that are only similar in a particular aspect would be excluded. In our example (Table 8.5), the papers about *emotion classification* would have been excluded even though the task is very related to *sentiment classification*. Moreover, the specialized embedding space allows dissimilarity search, e. g., considering papers with similarity above a certain threshold. This allows retrieving papers similar in their task but different in their method. The formulation of such queries could furthermore facilitate the discovery of analogies between research papers (Chan et al., 2018).

8.4 Summary of the Chapter

This chapter investigated Research Task **IV** and proposed specialized document representations for aspect-based similarity of research papers. Instead of considering each research paper as a single entity for document similarity, we incorporated multiple aspects in our approach, i. e.,

task, *method*, and *dataset*. Therefore, we moved from a single generic representation to three specialized ones. We treated aspect-based similarity as a classical vector similarity problem in aspect-specific embedding spaces.

This chapter contributed two major improvements to aspect-based document similarity: In contrast to segment-level similarity (Chan et al., 2018; Huang et al., 2020; Kobayashi et al., 2018), documents were not divided into segments, potentially harming the coherence of the document. Instead, we preserved the semantics of the whole document that is needed for a meaningful representation. Additionally, our approach was less resource intensive and achieves higher precision and recall compared to the pairwise document classification baseline (Chapter 7). The improved scalability allowed the development of a recommender system, which we demonstrated with our demo[2]. Having such a working prototype with recommendations for a large corpus based on aspect-based similarity fulfills the goal formulated in Research Task **IV**.

In our experiments, we compared and analyzed four generic document embeddings, ten specialized document embeddings, and a pairwise classification baseline in the context of research paper recommendations. To the best of our knowledge, all applied specialization methods were, so far, used only to derive generic embeddings. Our evaluation is conducted on the newly constructed Papers with Code corpus containing more than $150,000$ research papers. The Papers with Code dataset is unique for research on aspect-based document similarity as it contains curated annotations regarding different aspects of research papers in the machine learning domain. Thus, the dataset from this chapter complements the citation dataset from Chapter 7. In our experiments, Siamese SciNCL outperformed all other methods with a 0.20 avg. MAP score.

Our comparison between recommendations using generic and specialized embeddings indicated a tendency of generic recommendations to be more similar regarding *dataset* than *method*. Thus, papers with a similar method were less likely to be recommended with these generic embeddings. This outcome confirms the findings from Research Task **I**, which already identified that the lack of aspect information implicitly impacts recommendations. The aspect-specific document embeddings mitigate potential risks arising from implicit biases by making them explicit. This can, for example, be used for diverse recommendations with higher coverage, e. g., by recommending documents for every aspect.

Most importantly, we have presented a practical approach that allows aspect-based document similarity to be integrated into a recommender system since the approach scales to large document corpora. The approach even can be combined with existing approximate nearest neighborhood frameworks that have been shown to scale to trillions of documents, e.g., SCANN (Guo et al., 2020) or FAISS (Johnson et al., 2021).

Part IV

Final Considerations

Chapter 9

Conclusion and Future Work

This chapter concludes the thesis by summarizing the presented research in Section 9.1, providing an overview of the research contributions in Section 9.2, contrasting aspect-free and aspect-based similarity in Section 9.3, and highlighting areas for future research in Section 9.4.

9.1 Summary

This thesis introduced a new approach to literature recommendation systems through aspect-based document similarity. Aspect-based document similarity addresses limitations of existing aspect-free similarity measures, which fail to differentiate between the many aspects in which documents can be similar. By incorporating aspect information into document similarity measures, this thesis presented an approach that provides a more nuanced view of the document content. The differentiation between aspects gives literature recommender systems more control over the generated recommendations. This control can both tailor recommendations to specific aspects or diversify them across different aspects.

Similarity is a subjective and context-sensitive measure. What counts as "similar" or "not similar" can vary depending on the aspects a person considers when making the assessment. However, the unspecified use of similarity has been criticized in the psychological and philosophical literature (Chapter 2). In his famous critique, Goodman (1972) describes "similarity as a slippery and both philosophically and scientifically useless notion" unless one can say in what aspect two things are similar. As a consequence, the similarity should be measured with respect to a given aspect that provides the context and specifies to what the similarity relates. NLP tasks such as sentiment analysis are context-sensitive and address this by incorporating aspect information. For instance, attributing sentiment to specific aspects has been shown to be beneficial for analyzing customer feedback (Pontiki et al., 2016). However, to the best of our knowledge aspect-based document similarity has been an underexplored research area. Thus, it remained an open question about which impact the lack of aspect information has on document similarity.

This thesis evaluated document similarity measures in the context of literature recommendations and showed the impact of the lack of aspect information. We compared two classical similarity measures for Wikipedia article recommendations using qualitative and quantitative experiments (Chapter 3). The two methods, MLT and CPA, rely on different sources of information to determine whether documents are similar or not – text and graph information, respectively. The evaluations revealed that although the overall user satisfaction was comparable between the two information sources, the users perceived the recommendations from these sources as different. Therefore, the choice of similarity measures affects the recommendations, i.e., they implicitly address different aspects. MLT's recommendation had a narrow topical focus, whereas CPA was perceived as more diverse, as our user study showed. In other words, MLT and CPA implicitly convey a different notion of similarity and address different information needs. While the discrepancy between MLT and CPA can be attributed to the difference in the information they rely on, it demonstrates that Goodman's critique applies to MLT and CPA. Both methods measure how similar two documents are but do not specify in what aspects they are similar.

Subsequently, we conducted experiments with legal literature and state-of-the-art document representations (Chapter 4). The experiments revealed the same pattern we already found with MLT and CPA. Despite seemingly close recommendation scores, the overlap of the legal recommendation sets between the individual methods was very low. This suggests that the evaluated methods also implement a different notion of similarity. We found that the overlap is especially low when comparing methods based on different information sources, i.e., text or graph information. This shows the lack of aspect information also affects state-of-the-art document representations and legal literature recommendations.

The findings suggest the hybrid combination of text-based and graph-based methods since both information sources implicitly address different aspects. Based on this finding, the thesis introduced SciNCL, a representation learning approach for scientific documents (Chapter 5). SciNCL combines text and graph information and achieves state-of-the-art results on the SciDocs benchmark (Cohan et al., 2020). It is a general representation learning approach for which we show performance improvements on diverse tasks, ranging from topic classification to citation or user activity prediction. Due to its generality, SciNCL does not incorporate explicit aspect information, but it can be used as the foundation for both aspect-free and aspect-based document similarity.

Furthermore, the findings of the evaluation of existing methods motivate the development of document similarity measures that account for Goodman's critique and incorporate explicit aspect information. The similarity of documents should be measured with respect to a specific aspect that defines the perspective from which the document content is looked at when assessing the similarity. In other words, solving the identified problem corresponds to getting from an *aspect-free* to an *aspect-based* document similarity.

To develop an aspect-based document similarity, we followed the concepts of the feature similarity model from Tversky (1977) and conceptual spaces from Gardenfors (2004). Specifically, we first designed a pairwise multi-class document classification approach for measuring the similarity of a document pair for a given aspect. In experiments with Wikipedia articles and aspect information from Wikidata properties (Chapter 6), we demonstrated this approach's validity and evaluated its different implementations. Based on these findings, we extended this approach in the subsequent experiments. We changed the application domain from Wikipedia articles to research papers (Chapter 7), which can be seen as a more challenging literature domain for recommender systems. Additionally, we extended the classification task from a single-label to a multi-label problem such that a document pair can be similar not just in a single but in multiple aspects. Even with these two modifications, which increased the task's difficulty, pairwise document classification achieves high accuracy for aspect-based document similarity.

The pairwise classification approach has a quadratic time complexity with respect to the corpus size since it requires the classification of all possible document pairs in the document corpus. To achieve a linear complexity, we proposed specialized document representations for research papers (Chapter 8). Specialized document representations formulate aspect-based document similarity as a classical vector similarity problem in aspect-specific embedding spaces. For each aspect, one specialized embedding space is learned such that documents are located in close proximity when they are similar in this particular aspect. The computationally expensive encoding of aspect information is performed only once per document and aspect. After the initial encoding, similar documents can be retrieved through a nearest neighbor search in the aspect-specific embedding space. This makes the approach scalable to large document corpora.

To obtain aspect-specific document representations, we combined SciNCL with the Siamese specialization method and found this combination as the best-performing method in our experiments (Chapter 8). Further analysis of the resulting embedding spaces confirmed the findings concerning aspect-free methods. Specifically, we found that aspect-free representations of research papers had an implicit bias towards papers being similar in their dataset and against the similarity with respect to the methods utilized in the papers. With the classical approach of aspect-free similarity, this bias remains hidden despite affecting the recommendations. In contrast, aspect-based similarity mitigates potential risks arising from implicit biases by making them explicit and controllable.

The aspect-based similarity gives literature recommender systems more control over the generated recommendations. This can, for example, be used for more diverse and specifically tailored recommendations. As the aspect describes different semantics of the document content, the aspect information allows diversifying recommendations, e.g., by choosing the recommendations from documents that are similar in different aspects to the seed document. Generating recommendations from diverse aspects increases the recommendation coverage, as our experiments showed. Likewise, recommendations can be tailored to specific aspects that are most relevant to the users of the recommender system. These features are enabled by the specialized document representations, which efficiently incorporate aspect information making the aspect-based similarity scale linearly with respect to the corpus size.

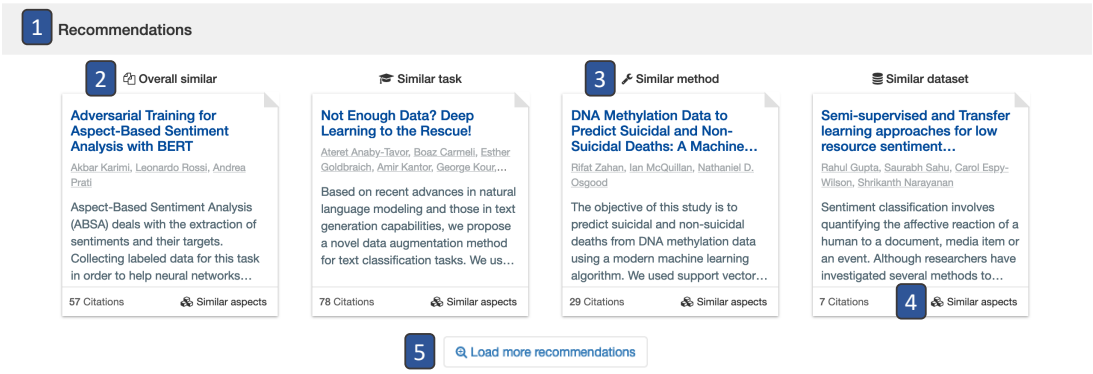


Figure 9.1: Example of aspect information integrated into a research paper recommendation system. 1) Similar papers are listed in the "Recommendations" section. 2) Overall similar papers are recommended using aspect-free similarity. 3) Aspect-specific recommendations are provided for *task*, *method*, or *dataset*. Users can click on the icon to browse more papers focused on a particular aspect. 4) Users can navigate to a detailed paper comparison. 5) More recommendations are available upon request.

We showcase our approach by creating a prototypical recommender system for research papers using aspect-based similarity.[2] Figure 9.1 shows a screenshot of the prototype and how aspects are integrated into a graphical user interface. The four presented recommendations are diverse since they reflect the aspect-free similarity (2) and aspect-based similarity (3) concerning the three aspects of *task*, *method*, or *dataset*. If users are interested in one particular aspect, they can click on the aspect icon (3) to browse more focused recommendations. A detailed comparison between the two papers is also available (4). The aspect-based document similarity developed in this thesis is the foundation for such a recommender system.

In summary, this thesis evaluated existing methods for content-based literature recommendations and for the underlying document similarity measures, identified the lack of aspect information as a major limitation of the existing methods, developed approaches to incorporate aspect information into document similarity, and iteratively improved the resulting approach.

9.2 Contributions

This thesis made three main contributions:

1. The thesis showed that the lack of aspect information in document similarity measures is not only a theoretical problem but has a notable impact on literature recommendations, which is demonstrated with quantitative and qualitative experiments.
2. The thesis presented two approaches for incorporating aspect information into document similarity measures that can both tailor literature recommendations to specific aspects or diversify recommendations across different aspects.
3. The thesis introduced a novel approach for state-of-the-art scientific document representations that combines text and graph information and improves aspect-free and aspect-based similarity measures and other downstream tasks.

The following section summarizes the individual contributions for each of the four research tasks, as defined in Section 1.3. The work on these research tasks resulted in nine core publications (Ostendorff, 2020; Ostendorff et al., 2021a; Ostendorff et al., 2022a; Ostendorff et al., 2021b; Ostendorff et al., 2022b; Ostendorff et al., 2020b; Ostendorff et al., 2020c; Schwarzer et al., 2017; Schwarzer et al., 2016b).

Furthermore, the thesis contributed to 14 additional publications partially related to the research tasks (Calizzano et al., 2021; Calizzano et al., 2022; Dehio et al., 2022; Garcia et al., 2023; Ostendorff et al., 2020a; Ostendorff et al., 2019; Ostendorff and Rehm, 2023; Raring et al., 2022; Rehm et al., 2022; Rehm et al., 2021; Rehm et al., 2020b; Ruan et al., 2022; Schulz et al., 2020; Schwarzer et al., 2016a). A detailed overview of these publications can be found in Section 1.5.

Research Task I

Evaluate state-of-the-art document similarity measures and underlying document representations that use text or graph information.

Contribution: Our evaluation showed that the lack of aspect information in document similarity measures is not only a theoretical problem but has a notable impact on literature recommendations (Ostendorff et al., 2021a; Ostendorff et al., 2021b; Schwarzer et al., 2017; Schwarzer et al., 2016b).

Research Task **I** was about the evaluation of the state-of-the-art in document similarity measures and underlying document representations for content-based literature recommendations. The analysis of the existing methods revealed the limitations, which are addressed in the subsequent research tasks, and determined the most promising technical directions.

As a first contribution to this research task, we reviewed the existing literature and laid the theoretical foundation for the empirical experiments (Chapter 2). Subsequently, we compared three document similarity measures in the context of Wikipedia articles (Chapter 3). Specifically, we evaluated MLT (an implementation of TF-IDF), Co-Citations, and CPA. We conducted a large-scale offline evaluation and a qualitative user study. With MLT, Co-Citations, and CPA, we selected three methods using either text or graph information. In particular, the combination of offline evaluation and user study provided insights into the evaluated methods. Most significantly, we identified that the lack of aspect information is not only a theoretical problem but has a notable impact on the recommendations. We found that these text-based and graph-based methods yield different kinds of document similarity, each implicitly addressing different aspects and information needs, and that this difference is also perceived by the users.

To verify this finding, we complemented our work on Research Task **I** with an evaluation of a large number of state-of-the-art document representation methods in the context of the legal literature (Chapter 4). Our offline evaluation compared 25 methods ranging from word vectors over language models to graph embeddings and their hybrid combinations. We found that graph-based and text-based methods yield comparable accuracy scores but produce different recommendation sets with low overlap. These findings align with the ones from Chapter 3. Moreover, graph-based and text-based methods were vulnerable to certain dataset characteristics like text length or citation count. Combining text and graph in a hybrid method reduced the weaknesses of a single information source and increased recommendation diversity. Overall, the hybrid methods also achieved the best results for legal recommendations.

Both experiments have shown that different methods also implicitly address different aspects. In other words, the methods implicitly convey a different notion of document similarity. These findings highlight the need for aspect-based document similarity and the hybrid combination of individual methods.

Research Task II

Design one document representation method that improves upon the state-of-the-art while using both text and graph information.

Contribution: We designed a novel approach for state-of-the-art scientific document representations that combines text and graph information and improves aspect-free and aspect-based similarity measures and other downstream tasks (Ostendorff et al., 2022b).

This thesis investigated the domains of Wikipedia, court decisions, and research papers. In all three domains, the textual content of a document is complemented by links or citations that provide additional semantic information. To overcome the limitations of individual methods that either rely on text or graph information, we combined both sources of information.

At first, we combined text and graph information through concatenation or score summation (Chapter 4). We either concatenated the document vectors of a text-based method with the vectors of a graph-based method to obtain hybrid vectors, or we added up the scores of two individual methods, i.e., we added up cosine similarities. Both approaches showed a positive effect on the recommendation performance, but they are also affected by the weakness of the individual methods. For example, score summation yielded only the best results when both information

sources were available. However, document representations are also needed for many applications where no or only little graph data in the form of citations is available.

To address these limitations, we proposed SciNCL as a text-based document encoder trained with citation information (Chapter 5). The foundation of SciNCL is the contrastive fine-tuning of a SciBERT language model that is built upon informative positive and negative samples derived from citation embeddings. SciNCL achieved state-of-the-art results for scientific document representations in the SciDocs benchmark (Cohan et al., 2020). The benchmark covers seven tasks, including topic classification, citation prediction, user activity prediction, and recommendations. In the subsequent experiments, we also showed how SciNCL could be used as the base model for aspect-based document similarity.

Research Task III

Design an aspect-based document similarity measure to address the limitations of existing aspect-free similarity measures.

Contribution: We designed the pairwise multi-class document classification approach, which addresses the limitation of aspect-free similarity by incorporating aspect information into document similarity, and evaluated this approach for Wikipedia articles and scientific literature (Ostendorff et al., 2020b; Ostendorff et al., 2020c).

Our findings from Research Task **I** emphasized the need to extend document similarity measures with aspect information. Given the novelty of aspect-based document similarity and a limited amount of related work (Section 2.1.1 and 2.5), we explored its general feasibility in Chapter 6.

Aspect-free similarity can be seen as a single-class classification problem for document pairs. Given a pair of documents, a classifier predicts if the two documents are similar or not similar. In line with this, we formalized aspect-based document similarity as a pairwise multi-class document classification problem. The classifier predicts the aspect-based similarity for a given aspect and a document pair. We evaluated this approach using Wikipedia articles as the application domain and Wikidata properties as the ground truth for aspect information (Chapter 6). Wikipedia and Wikidata provided sufficient data such that we were enabled to compare state-of-the-art deep learning approaches without suffering from data scarcity. Moreover, Wikipedia properties represented diverse aspect classes with varying classification difficulty. Our experiments showed that vanilla Transformer models outperformed all other methods, which we attribute to the pretraining with the next sentence prediction objective.

In a second experiment, we continued the same line of research but with two modifications (Chapter 7). We investigated research papers as a literature domain and relied on the section titles in that citations occur as a source for aspect information. Also, we increased the classification task's difficulty by modeling aspect-based similarity as a multi-label classification instead of a single-label classification, as done in the Wikipedia experiment. Despite these modifications, pairwise multi-class classification was able to achieve high accuracy for measuring the aspect-based similarity of research papers.

In summary, these two studies have shown the validity of the pairwise multi-class classification approach for aspect-based document similarity. The validity is shown for two application domains, namely Wikipedia articles and research papers, and for diverse aspect classes.

Research Task IV

Implement aspect-based document similarity such that it scales to large document corpora.

Contribution: We implemented aspect-based document similarity using specialized document representations that formulate aspect-based similarity as a classical vector similarity problem in aspect-specific embedding spaces. This approach scales linearly with respect to the corpus size and allows tailoring recommendations for specific aspects or diversifying recommendations across aspects (Ostendorff et al., 2022a).

Recommender systems are applications typically deployed into a live system where users interact with the recommendations. In particular, recommender systems should be capable of generating recommendations for corpora with a large number of documents. The recommendations must also reflect changes in the underlying corpus, e.g., when new documents are added. Thus, recommender systems should be scalable in terms of corpus size and corpus changes. The pairwise document classification approach has a quadratic complexity since it requires $\mathcal{O}(n^2)$ document pair classification for a corpus of n documents. Such a quadratic complexity is computationally expensive. Furthermore, the classification must also be repeated every time a new document gets added to the corpus leading to additional computational costs.

To reduce the computational costs, we presented the specialized document representations as an approach to aspect-based document similarity that scales efficiently to large document corpora (Chapter 8). This approach treats aspect-based similarity as a vector similarity problem in aspect-specific embedding spaces. The computationally expensive encoding of aspect information is only performed once per document and aspect. Retrieving similar documents can then be done through a nearest neighbor search in each aspect-specific embedding space. As a result, this approach has linear time complexity, i.e., $\mathcal{O}(n)$ with respect to n documents in the corpus. To showcase the functionality of this approach, we implemented a prototypical recommender system based on the proposed method. Our prototype displays the top-k recommendations selected from different aspects and allows users to browse recommendations for specific aspects (Figure 9.1).

Other contributions. We created several datasets, pretrained models and open source implementations and made them publicly available. Table 9.1 summarizes these contributions.

Table 9.1: Our other contributions, including datasets and open source implementations.

Contribution	Publication	Link
Evaluation benchmark for Wikipedia recommendation ("See also" links)	Schwarzer et al. (2016b)	github.com/wikimedia/citolytics
CPA-based Wikipedia recommender system, including backend and Android app integration	Schwarzer et al. (2017)	github.com/malteos/apps-android-wikipedia
Evaluation benchmark for legal document similarity (Wikisource and Open Case Book)	Ostendorff et al. (2021a)	github.com/malteos/legal-document-similarity
Pretrained SciNCL language model	Ostendorff et al. (2022b)	hf.co/malteos/scincl
Citation section title dataset	Ostendorff et al. (2020b)	github.com/malteos/aspect-document-similarity
Aspect-based similarity labels for research papers (Papers With Code)	Ostendorff et al. (2022a)	github.com/malteos/aspect-document-embeddings

9.3 Lessons Learned

The research presented in this thesis is centered around aspect-based similarity and its aspect-free counterpart. This section presents the lessons learned from our research by contrasting both types of similarity, summarizing their strengths and weaknesses, and recommending under what circumstances one of the similarities is superior over the other one.

An aspect determines the perspective of how we look at the content of a document (or an item in general) when assessing similarity. More formally, aspect-based similarity is a function of two items and an aspect. The aspect is assumed to be a part of a given set of aspects. Opposed to this, the classical aspect-free approach to similarity would correspond to a function of only two items without any given aspect. But our experiments showed that aspects are implicitly contained in the aspect-free similarity measures. Therefore, we conclude that aspect-free similarity is a special case of aspect-based similarity. The aspect-free similarity function is also parameterized with an aspect, but this parameter is an unspecified single aspect or an arbitrarily large set of aspects.

Our findings have implications for using similarity measures in recommender systems and other applications. The aspect-free similarity has a smaller implementation effort compared to aspect-based similarity. With aspect-free similarity, the set of aspects can remain undefined and supervised training data is not required. Aspect-free similarity can be considered a one-fits-all approach. Analog to user-based recommender systems, aspect-free similarity is the recommendation of the most popular items, whereas aspect-based similarity is more comparable to collaborative filtering. Such a one-fits-all approach yields good but not optimal recommendations while keeping the effort at a minimum. Recommending the most popular items is often a strong baseline compared to collaborative filtering approaches, as shown by Ji et al. (2020). Thus, the aspect-free similarity is a suitable approach when resources are scarce and recommendations are not crucial for the application. Wikipedia is such an example. Wikipedia's audience is mostly casual users that look up specific information instead of conducting literature surveys, as our user study showed.

The majority of our participants were already satisfied with recommendations from aspect-free similarity measures.

Implementing aspect-based similarity, on the other hand, requires more effort. The set of aspects needs to be defined and aligned to the application. Concerning the implementation, this means significant effort for creating supervised datasets when using the methods developed in this thesis. The increased effort yields a much more specialized implementation of similarity compared to the one-fits-all approach of aspect-free similarity. Hence, aspect-based similarity targets an expert audience, e.g., researchers or legal professionals. An expert audience benefits from aspect-based similarity since experts tend to have complex information needs and are less interested in just the most popular items.

Ultimately, whether the effort for aspect-based similarity is justified depends on its importance for the overall application. In law, there are high stakes for finding the most relevant information as a case can be won or lost depending on whether or not supporting information can be found. These high stakes justify a high implementation effort. Similarly, scientific literature recommendations that address a complex information need can foster innovations and discoveries, as shown by Chan et al. (2018). Generally speaking, any improvement to an application can make it more competitive and attract new users. In particular, aspect-based similarity can be the edge that sets an application apart from competitors. In a commercial setting, this could decide the success of a business. Finally, even tiny improvements in the recommender systems of applications like YouTube or Netflix can have significant effects due to the scale of these applications.

In conclusion, aspect-based similarity is an improvement over aspect-free similarity, but it requires additional effort. If the effort is reasonable depends on the application in that the similarity measure is used. However, it is vital that the application developers are aware of the role of aspects in similarity measures and their opportunities to enhance applications.

9.4 Future Work

The research presented in this thesis yielded various ideas to improve document similarity measures and other NLP tasks. We briefly discuss these ideas in the following.

Joint graph and text representation learning. Representation learning underpins all subsequent applications, whether it is document similarity or any other NLP task. Therefore, future work needs to focus on improving document representation methods. With SciNCL, we have shown the advantage of combing citation graphs with text information for scientific document representation learning. For other domains, related approaches like LinkBERT (Yasunaga et al., 2022) have shown link or citation prediction as a beneficial pretraining objective for Transformer language models. Despite achieving state-of-the-art results, these approaches still treat text and graph information as separate entities. The language and graph models are separately trained, and then a joint model is constructed based on the two sub-models. The separate learning of text and graph information is probably suboptimal compared to true joint learning. However, the exact approach of integrating two different modalities into a joint learning framework remains challenging. A potential approach could be to treat citation markers as regular tokens in a language modeling task that induces document-to-document semantics with text generation pretraining objectives. In recent works, Tay et al. (2022b) and Bevilacqua et al. (2022) have shown that text generation models can be used for information retrieval by generating document identifiers as

answers to a given query. We envision a similar approach capable of generating citation markers for a document similarity task.

Zero-shot pairwise document classification. The experiments on aspect-based document similarity conducted in this thesis assumed a predefined set of aspects. Being constrained on a set of aspects is sufficient when the potential information needs are limited. However, one can imagine other use cases in that you do not have a predefined set of aspects but rather need to determine the document similarity with respect to an arbitrary aspect. Large language models could provide aspect-based document similarity for arbitrary aspects. Brown et al. (2020) and Wang et al. (2021) have demonstrated the zero-shot or few-shot capabilities of large language models for the entailment task, which is essentially a pairwise document classification task. This approach could be adopted for aspect-based document similarity without requiring a predefined set of aspects and reducing the need for the expensive collection of supervised datasets.

Explainable content-based recommendations. As described in Chapter 3, the user study participants trusted the recommender system without understanding how the system generated the recommendation. Recommender systems should provide explanations that help users get an intuition of why a particular item is recommended to respect the user's trust. Also, explanations would help users to comprehend the connections between the seed item and the recommendations. Explainable recommendations are a subject of active research (Kunkel et al., 2019; Zhang and Chen, 2020). However, most research focuses on only user-based approaches. The content-based approaches would benefit as well from explanations. One promising line of work is the task of citation text generation (Luu et al., 2020; Xing et al., 2020). The aspect-based document similarity could also provide such explanations. However, further research in the form of user studies is needed in this direction.

Broader impact. The research presented in this thesis focussed on the literature recommendation use case. Given that similarity measures underpin many other use cases, we expect our work to be as well relevant outside of literature recommender systems. The most obvious impact would be on other types of recommendations. For instance, aspect-based similarity could be extended to product recommendations, where the aspects could be concerned with colors or other product attributes. More generally, aspect-based similarity allows tailoring recommendations to specific information needs and, therefore, it addresses one of the major weaknesses of content-based recommendations compared to user-based ones. User-based recommender systems dominate, especially in commercial settings, despite having known issues. In particular, user-based systems require the collection of large amounts of user information that might be unavailable (cold start problem) or cause privacy issues. In other use cases, information about the historical user interest may lead to irrelevant recommendations when the user's information need is changing too frequently. Given that aspect-based similarity enables tailored recommendations based on the content alone, recommender systems can get less dependent on user information, i.e., it would decrease the need for collecting user information but still allow tailored recommendations.

Another application for which our research is relevant is semantic storytelling, i.e., the semi-automatic story generation based on existing pieces of content. In Rehm et al. (2022) and Raring et al. (2022), we have already demonstrated that the multi-class pairwise classification approach can be used to arrange text segments into new stories. Similarly, multi-document summarization can benefit from aspect information. Aspect-based document similarity could be used to reduce

redundancy and to compose richer summaries by identifying the fine-grained differences and similarities of documents. Our experiments in Ruan et al. (2022) already showed that additional information about the hierarchical document structure is beneficial for summarization. Overall, we expect a broader impact on applications dealing with complex documents such as laws, patents, and other industry use cases.

Appendix

List of Figures

1.1 Content-based recommender systems rely on similarity measures 4
1.2 Recommendations from aspect-based document similarity 6

2.1 Geometric similarity model 24
2.2 Feature similarity model 24
2.3 Vector space model 28
2.4 One-hot representation 30
2.5 Dense representation 30
2.6 Illustration of a graph of documents 38
2.7 Direction citations 39
2.8 Co-citations 40
2.9 Co-citation proximity analysis 41

3.1 Distribution of words, in-links, and out-links in the Wikipedia dataset 55
3.2 Number of links per "See also" section in Wikipedia articles. 56
3.3 Overview of study design. 60
3.4 Optimization of the α parameter of CPA's CPI model 63
3.5 CPA and MLT results for the number of words per article 65
3.6 CPA and MLT results for the number of in-links per article 66
3.7 User study responses for MLT and CPA 72
3.8 Responses about diversity or similarity 73
3.9 Responses about familiarity with article topic 74
3.10 Responses about user's satisfaction 75

4.1 MAP scores with respect to words in the seed document 87
4.2 MAP scores with respect to citation count of seed documents 89
4.3 Jaccard index for similarity or diversity of two recommendation sets 90

5.1 Positive and negatives samples from the citation graph neighborhood 99
5.2 Results for positive sampling settings 108
5.3 Results for negative sampling settings 108
5.4 Number of collisions w.r.t. the sample induced margin 109
5.5 Validation performance for the recommendation task 110

6.1 Shared aspects between Wikipedia articles 118
6.2 Vanilla Transformer for sequence pair classification 123
6.3 Siamese Transformer architecture 123
6.4 Results for vanilla and Siamese Transformers w.r.t. sequence length 126
6.5 Results for the models w.r.t. concatenation 127
6.6 Confusion matrix for the predicted and Wikidata classes 129

7.1 Illustration of aspect-free vs. aspect-based similarity 133
7.2 Dataset with citations' section titles 135
7.3 A Transformer model with titles and abstracts as input is used for classification. 137

8.1 Illustration of generic and aspect-specific embeddings 146
8.2 Results of the pairwise SciNCL baseline 152
8.3 Precision and MAP@k for two generic and two specialized embeddings 154
8.4 Visualization of generic and specialized embedding spaces 156

9.1 Example of aspect information being integrated into a research paper recommendation system 167

List of Tables

2.1 Overview of literature concerned with aspects grouped by research task. 44

3.1 Approximated runtimes for each task. 59
3.2 Overview of seed articles selected for the study. 62
3.3 Results based on “See also” links and clickstreams 64
3.4 Recommendations for the “Technical University of Berlin” Wikipedia article . 67
3.5 Recommendations for the “Elvis Presley” Wikipedia article 68
3.6 Recommendations for the “Newspaper” Wikipedia article 69

4.1 Distribution of relevance annotations for Open Case Book and Wikisource. . . 81
4.2 Overall scores for recommendations from Open Case Book and Wikisource . . 85
4.3 Example recommendations for Open Case Book 91
4.4 Example recommendations for Wikisource 92

5.1 Results on the SciDocs test set 106
5.2 Ablation analysis 111

6.1 The aspect classes with their Wikidata PIDs and examples 121
6.2 Overall results for Wikipedia pairwise classification 125
6.3 Aspect-level results for Wikipedia articles 128
6.4 Examples for the aspect-based similarity between Wikipedia articles 130

7.1 Label class distribution for ACL Anthology and CORD-19 136
7.2 Overall results for research paper classification 139
7.3 Results of SciBERT on ACL Anthology and CORD-19 datasets per aspect class 140
7.4 Confusion matrix of selected multi-labels for SciBERT 141
7.5 Example aspect-based similarity of research paper pairs 142

8.1 Dataset statistics for each aspect 149
8.2 Classification results for Pairwise SciNCL. 152
8.3 Overall results for generic and aspect-specific representations 153
8.4 Intersection of recommendation sets 155
8.5 Example recommendations from SciNCL (generic) and Siamese SciNCL (aspect-specific) 158

9.1 Our other contributions, including datasets and open source implementations. . 172

Bibliography of Publications

Rémi Calizzano, **Malte Ostendorff**, and Georg Rehm (Aug. 2021). "Ordering Sentences and Paragraphs with Pre-trained Encoder-Decoder Transformers and Pointer Ensembles". In: *Proceedings of the 21st ACM Symposium on Document Engineering (DocEng)*. Limerick, Ireland: Association for Computing Machinery, pp. 1–9 (cit. on pp. 11, 168).

Rémi Calizzano, **Malte Ostendorff**, Qian Ruan, and Georg Rehm (June 2022). "Generating Extended and Multilingual Summaries with Pre-trained Transformers". In: *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*. June 20-25, 2022. Marseille, France: European Language Resources Association (ELRA) (cit. on pp. 11, 47, 168).

Niklas Dehio, **Malte Ostendorff**, and Georg Rehm (June 2022). "Claim Extraction and Law Matching for COVID-19-related Legislation". In: *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*. Marseille, France: European Language Resources Association (ELRA) (cit. on pp. 5, 11, 168).

Maria Gonzalez Garcia, Julian Moreno-Schneider, **Malte Ostendorff**, and Georg Rehm (Apr. 2023). "Integration of a Semantic Storytelling Recommender System in Speech Assistants". In: *Proceedings of Text2Story – Sixth International Workshop on Narrative Extraction from Texts held in conjunction with the 45th European Conference on Information Retrieval (ECIR)*. Dublin, Ireland, pp. 5–11 (cit. on pp. 12, 168).

Malte Ostendorff (2020). "Contextual Document Similarity for Content-based Literature Recommender Systems". In: *Proceedings of the Doctoral Consortium at ACM/IEEE Joint Conference on Digital Libraries (JCDL)* (cit. on pp. 10, 168).

Malte Ostendorff, Elliott Ash, Terry Ruas, Bela Gipp, Julian Moreno-Schneider, and Georg Rehm (2021a). "Evaluating Document Representations for Content-Based Legal Literature Recommendations". In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law (ICAIL)*. São Paulo, Brazil: Association for Computing Machinery, pp. 109–118. DOI: 10.1145/3462757.3466073 (cit. on pp. 10, 79, 91, 168, 172).

Malte Ostendorff, Till Blume, and Saskia Ostendorff (Aug. 2020a). "Towards an Open Platform for Legal Information". In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. New York, NY, USA: ACM, pp. 385–388. DOI: 10.1145/3383583.3398616 (cit. on pp. 11, 81, 168).

Malte Ostendorff, Till Blume, Terry Ruas, Bela Gipp, and Georg Rehm (2022a). "Specialized Document Embeddings for Aspect-based Similarity of Research Papers". In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. Vol. 1. 1. Cologne, Germany: Association for Computing Machinery. DOI: 10.1145/3529372.3530912 (cit. on pp. 10, 145, 168, 171, 172).

Malte Ostendorff, Peter Bourgonje, Maria Berger, Julian Moreno-Schneider, Georg Rehm, and Bela Gipp (2019). "Enriching BERT with Knowledge Graph Embeddings for Document Classification". In: *Proceedings of the GermEval Workshop 2019 – Shared Task on the Hierarchical Classification of Blurbs co-located with the 15th Conference on Natural Language Processing (KONVENS)*. Erlangen, Germany: German Society for Computational Linguistics & Language Technology, pp. 305–312. DOI: 10.48550/arXiv.1909.08402 (cit. on pp. 11, 119, 137, 168).

Malte Ostendorff, Corinna Breitinger, and Bela Gipp (2021b). "A Qualitative Evaluation of User Preference for Link-Based vs. Text-Based Recommendations of Wikipedia Articles". In: *Towards Open and Trustworthy Digital Societies: 23rd International Conference on Asia-Pacific Digital Libraries (ICADL)*. Virtual Event: Springer-Verlag, pp. 63–79. DOI: 10.1007/978-3-030-91669-5_6 (cit. on pp. 10, 53, 168).

Malte Ostendorff and Georg Rehm (2023). "Efficient Language Model Training through Cross-Lingual and Progressive Transfer Learning". In: *Practical ML for Developing Countries Workshop co-located with the International Conference on Learning Representations (PML4DC@ICLR)*. DOI: 10.48550/ARXIV.2301.09626 (cit. on pp. 11, 168).

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm (Dec. 2022b). "Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings". In: *The 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Abu Dhabi: Association for Computational Linguistics. DOI: 10.48550/arXiv.2202.06671 (cit. on pp. 10, 97, 101, 102, 104, 145, 168, 169, 172).

Malte Ostendorff, Terry Ruas, Till Blume, Bela Gipp, and Georg Rehm (Dec. 2020b). "Aspect-based Document Similarity for Research Papers". In: *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 6194–6206. DOI: 10.18653/v1/2020.coling-main.545 (cit. on pp. 10, 133, 168, 170, 172).

Malte Ostendorff, Terry Ruas, Moritz Schubotz, Georg Rehm, and Bela Gipp (2020c). "Pairwise Multi-Class Document Classification for Semantic Relations between Wikipedia Articles". In: *Proceedings of the 2020 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. DOI: 10.1145/3383583.3398525 (cit. on pp. 10, 117, 168, 170).

Michael Raring, **Malte Ostendorff**, and Georg Rehm (June 2022). "Semantic Relations between Text Segments for Semantic Storytelling: Annotation Tool – Dataset – Evaluation". In: *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*. June 20-25, 2022. Marseille, France: European Language Resources Association (ELRA) (cit. on pp. 11, 168, 174).

Georg Rehm, **Malte Ostendorff**, Rémi Calizzano, Karolina Zaczynska, and Julián Moreno Schneider (Sept. 2022). "Identification of Relations between Text Segments for Semantic Storytelling". In: *Proceedings of the Conference on Digital Curation Technologies (QURATOR)*. 23 September 2022. Berlin, Germany (cit. on pp. 5, 11, 168, 174).

Georg Rehm, Karolina Zaczynska, Peter Bourgonje, **Malte Ostendorff**, Julián Moreno-Schneider, Maria Berger, Jens Rauenbusch, André Schmidt, Mikka Wild, Joachim Böttger, Joachim Quantz, Jan Thomsen, and Rolf Fricke (Nov. 2021). "Semantic Storytelling: From Experiments and Prototypes to a Technical Solution". In: *Computational Analysis of Storylines: Making Sense of Events*. Studies in Natural Language Processing. Cambridge: Cambridge University Press, pp. 240–259. DOI: 10.1017/9781108854221.015 (cit. on pp. 11, 168).

Georg Rehm, Karolina Zaczynska, Julian Moreno Schneider, **Malte Ostendorff**, Peter Bourgonje, Maria Berger, Jens Rauenbusch, Andre Schmidt, and Mikka Wild (2020a). "Towards Discourse Parsing-inspired Semantic Storytelling". In: *Proceedings of the Conference on Digital Curation Technologies (QURATOR)* (cit. on p. 11).

Georg Rehm, Karolina Zaczynska, Julian Moreno Schneider, **Malte Ostendorff**, Peter Bourgonje, Maria Berger, Jens Rauenbusch, Andre Schmidt, and Mikka Wild (2020b). "Towards Discourse Parsing-inspired Semantic Storytelling". In: *Proceedings of the Conference on Digital Curation Technologies (QURATOR)*. Berin, Germany (cit. on p. 168).

Qian Ruan, **Malte Ostendorff**, and Georg Rehm (May 2022). "HiStruct+: Improving Extractive Text Summarization with Hierarchical Structure Information". In: *Findings of the Association for Computational Linguistics (ACL)*. 22-27 May 2022. Association for Computational Linguistics. DOI: 10.18653/v1/2022.findings-acl.102 (cit. on pp. 11, 47, 168, 175).

Sarah Schulz, Jurica Ševa, Samuel Rodriguez, **Malte Ostendorff**, and Georg Rehm (2020). "Named Entities in Medical Case Reports: Corpus and Experiments". In: *Proceedings of the 12th Language Resources and Evaluation Conference (LREC)*. URL: https://www.aclweb.org/anthology/2020.lrec-1.553/ (cit. on pp. 11, 168).

Malte Schwarzer, Corinna Breitinger, Moritz Schubotz, Norman Meuschke, and Bela Gipp (2017). "Citolytics: A Link-based Recommender System for Wikipedia". In: *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys)*. ACM. New York, New York, USA: ACM Press, pp. 360–361. DOI: 10.1145/3109859.3109981 (cit. on pp. 10, 42, 53, 168, 172).

Malte Schwarzer, Jonas Düver, Danuta Ploch, and Andreas Lommatzsch (2016a). "An Interactive e-Government Question Answering System". In: *LWDA 2016 conference - Lernen, Wissen, Daten, Analysen (LWDA)*. Vol. 1670. September, pp. 74–82 (cit. on pp. 5, 11, 168).

Malte Schwarzer, Moritz Schubotz, Norman Meuschke, Corinna Breitinger, Volker Markl, and Bela Gipp (2016b). "Evaluating Link-based Recommendations for Wikipedia". In: *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries (JCDL)*. New York, New York, USA: ACM Press, pp. 191–200. DOI: 10.1145/2910896.2910908 (cit. on pp. 10, 41, 53, 168, 172).

Bibliography

Gediminas Adomavicius and Alexander Tuzhilin (June 2005). "Toward the next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions". In: *IEEE Transactions on Knowledge and Data Engineering* 17.6, pp. 734–749. ISSN: 1041-4347. DOI: 10.1109/TKDE.2005.99 (cit. on p. 13).

Nitin Agarwal, Ravi Shankar Reddy, Kiran Gvr, and Carolyn Penstein Rosé (2011). "SciSumm: A Multi-Document Summarization System for Scientific Articles". In: *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of Student Session (ACL HLT 2011)*, pp. 115–120 (cit. on p. 142).

Charu C. Aggarwal (2016). "Content-Based Recommender Systems". In: *Recommender Systems*. Cham: Springer International Publishing, pp. 139–166. ISBN: 978-3-319-29657-9. DOI: 10.1007/978-3-319-29659-3_4 (cit. on pp. 13, 14).

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre (June 2012). "SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity". In: **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Montréal, Canada: Association for Computational Linguistics, pp. 385–393. URL: https://aclanthology.org/S12-1051 (cit. on pp. 141, 142).

Md. Hijbul Alam, Woo-Jong Ryu, and SangKeun Lee (Apr. 2016). "Joint Multi-Grain Topic Sentiment: Modeling Semantic Aspects for Online Reviews". In: *Information Sciences* 339, pp. 206–223. ISSN: 0020-0255. DOI: 10.1016/j.ins.2016.01.013 (cit. on pp. 44, 46).

Alexander Alexandrov, Rico Bergmann, Stephan Ewen, Johann-Christoph Freytag, Fabian Hueske, Arvid Heise, Odej Kao, Marcus Leich, Ulf Leser, Volker Markl, et al. (Dec. 2014). "The Stratosphere Platform for Big Data Analytics". In: *The VLDB Journal* 23.6, pp. 939–964. ISSN: 1066-8888. DOI: 10.1007/s00778-014-0357-y (cit. on p. 58).

Haifa Alharthi, Diana Inkpen, and Stan Szpakowicz (Aug. 2018a). "A Survey of Book Recommender Systems". In: *Journal of Intelligent Information Systems* 51.1, pp. 139–160. ISSN: 0925-9902, 1573-7675. DOI: 10.1007/s10844-017-0489-9 (cit. on p. 17).

Haifa Alharthi, Diana Inkpen, and Stan Szpakowicz (2018b). "Authorship Identification for Literary Book Recommendations". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 390–400 (cit. on p. 17).

Zafar Ali, Guilin Qi, Khan Muhammad, Bahadar Ali, and Waheed Ahmed Abro (Dec. 2020). "Paper Recommendation Based on Heterogeneous Network Embedding". In: *Knowledge-Based Systems* 210, p. 106438. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2020.106438 (cit. on pp. 14, 20, 85).

Zafar Ali, Irfan Ullah, Amin Khan, Asim Ullah Jan, and Khan Muhammad (May 2021). "An Overview and Evaluation of Citation Recommendation Models". In: *Scientometrics* 126.5, pp. 4083–4119. ISSN: 0138-9130, 1588-2861. DOI: 10.1007/s11192-021-03909-y (cit. on pp. 14, 15).

Rana Alshaikh, Zied Bouraoui, and Steven Schockaert (2019). "Learning Conceptual Spaces with Disentangled Facets". In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, pp. 131–139. DOI: 10.18653/v1/K19-1013 (cit. on pp. 44, 48).

Xavier Amatriain and Justin Basilico (2015). "Recommender Systems in Industry: A Netflix Case Study". In: *Recommender Systems Handbook*. Boston, MA: Springer US, pp. 385–419. ISBN: 978-1-4899-7637-6. DOI: 10.1007/978-1-4899-7637-6_11 (cit. on p. 20).

Enrique Amigó, Fernando Giner, Julio Gonzalo, and Felisa Verdejo (2017). "An Axiomatic Account of Similarity". In: *Proceedings of the SIGIR'17 Workshop on Axiomatic Thinking for Information Retrieval and Related Tasks (ATIR)*, p. 10 (cit. on p. 26).

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata (2021). "Aspect-Controllable Opinion Summarization". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 6578–6593. DOI: 10.18653/v1/2021.emnlp-main.528 (cit. on pp. 44, 47).

Adam Amram, Anat Ben David, and Reut Tsarfaty (Aug. 2018). "Representations and Architectures in Neural Sentiment Analysis for Morphologically Rich Languages: A Case Study from Modern Hebrew". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for

Computational Linguistics, pp. 2242–2252. URL: https://aclanthology.org/C18-1190 (cit. on pp. 157, 158).

Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata (Mar. 2021). "Extractive Opinion Summarization in Quantized Transformer Spaces". In: *Transactions of the Association for Computational Linguistics* 9, pp. 277–293. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00366 (cit. on pp. 44, 47).

Stéphane Aroca-Ouellette and Frank Rudzicz (Nov. 2020). "On Losses for Modern Language Models". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 4970–4981. DOI: 10.18653/v1/2020.emnlp-main.403 (cit. on p. 36).

Sanjeev Arora, Yingyu Liang, and Tengyu Ma (2017). "A Simple But Though-To-Beat Baseline for Sentence Embeddings". In: *5th International Conference on Learning Representations (ICLR 2017)*. Vol. 15, pp. 416–424. URL: https://openreview.net/forum?id=SyK00v5xx (cit. on pp. 32, 92, 105, 106, 131).

Elliott Ash and Daniel L. Chen (May 2018). "Case Vectors: Spatial Representations of the Law Using Document Embeddings". In: *SSRN Electronic Journal* 11.2017, pp. 313–337. ISSN: 1556-5068. DOI: 10.2139/ssrn.3204926 (cit. on p. 16).

Javed A Aslam and Meredith Frost (2003). "An Information-theoretic Measure for Document Similarity". In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, p. 2 (cit. on p. 26).

Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull (2017). "ANN-Benchmarks: A Benchmarking Tool for Approximate Nearest Neighbor Algorithms". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 10609 LNCS. 614331. Springer, pp. 34–49. ISBN: 9783319684734. DOI: 10.1007/978-3-319-68474-1_3 (cit. on p. 148).

Ali Taleb Mohammed Aymen and Saidi Imène (2022). "Scientific Paper Recommender Systems: A Review". In: *Artificial Intelligence and Heuristics for Smart Energy Efficiency in Smart Cities*. Vol. 361. Cham: Springer International Publishing, pp. 896–906. ISBN: 978-3-030-92038-8. DOI: 10.1007/978-3-030-92038-8_92 (cit. on pp. 13, 14).

Amos Azaria, Avinatan Hassidim, Sarit Kraus, Adi Eshkol, Ofer Weintraub, and Irit Netanely (2013). "Movie Recommender System for Profit Maximization". In: *Proceedings of the 7th ACM Conference on Recommender Systems*. Hong Kong, China: Association for Computing Machinery, pp. 121–128. DOI: 10.1145/2507157.2507162 (cit. on pp. 13, 19).

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton (2016). "Layer Normalization". In: *arXiv preprint arXiv:1607.06450*. DOI: 10.48550/arXiv.1607.06450 (cit. on p. 35).

Marcos Baez, Daniil Mirylenka, and Cristhian Parra (2011). "Understanding and Supporting Search for Scholarly Knowledge". In: *Proceeding of the 7th European Computer Science Summit*, p. 8 (cit. on p. 14).

Ricardo Baeza-Yates and Berthier Ribeiro-Neto (1999). "Modern Information Retrieval". In: *New York* 9, p. 513. ISSN: 0022-541X. DOI: 10.1080/14735789709366603 (cit. on p. 13).

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *arXiv preprint arXiv:1409.0473* (cit. on p. 34).

Xiaomei Bai, Mengyang Wang, Ivan Lee, Zhuo Yang, Xiangjie Kong, and Feng Xia (2019). "Scientific Paper Recommendation: A Survey." In: *IEEE Access* 7, pp. 9324–9339 (cit. on p. 14).

Protima Banerjee and Hyoil Han (2009). "Language Modeling Approaches to Information Retrieval". In: *Journal of Computing Science and Engineering* 3.3, p. 22 (cit. on p. 17).

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch (2012). "UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures". In: *1st Joint Conference on Lexical and Computational Semantics (SEM 2012)* 2, pp. 435–440. URL: https://www.aclweb.org/anthology/S12-1059 (cit. on pp. 141, 142).

Daniel Bär, Torsten Zesch, and Iryna Gurevych (Sept. 2011). "A Reflective View on Text Similarity". In: *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*. Hissar, Bulgaria: Association for Computational Linguistics, pp. 515–520. URL: https://www.aclweb.org/anthology/R11-1071 (cit. on pp. 44, 49, 77, 117).

Pierpaolo Basile, Danilo Croce, Valerio Basile, and Marco Polignano (2018). "Overview of the EVALITA 2018 Aspect-based Sentiment Analysis Task (ABSITA)". In: *EVALITA Evaluation of NLP and Speech Tools for Italian*. Accademia University Press, pp. 10–16. ISBN: 978-88-319-7869-9. DOI: 10.4000/books.aaccademia.4451 (cit. on pp. 44, 46).

Joeran Beel, Corinna Breitinger, Stefan Langer, Andreas Lommatzsch, and Bela Gipp (2016a). "Towards reproducibility in recommender-systems research". In: *User Modeling and User-Adapted Interaction (UMAI)*. Vol. 26. DOI: 10.1007/s11257-016-9174-x (cit. on pp. 19, 79, 80, 92, 94).

Joeran Beel, Siddharth Dinesh, Philipp Mayr, Zeljko Carevic, and Jain Raghvendra (Mar. 2017). "Stereotype and Most-Popular Recommendations in the Digital Library Sowiport". In: DOI: 10.18452/1441 (cit. on p. 18).

Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger (2016b). "Research-paper recommender systems: a literature survey". In: *International Journal on Digital Libraries* 17.4, pp. 305–338. ISSN: 1432-1300. DOI: 10.1007/s00799-015-0156-0 (cit. on pp. 3, 8, 13, 14, 18–20, 54, 77, 134).

Joeran Beel, Bela Gipp, Stefan Langer, Marcel Genzmehr, Erik Wilde, Andreas Nürnberger, Jim Pitman (2011). "Introducing Mr. DLib, a Machine-readable Digital Library". In: *Proceeding of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries - JCDL '11*. New York, New York, USA: ACM Press, p. 463. DOI: 10.1145/1998076.1998187 (cit. on pp. 14, 20).

Joeran Beel and Stefan Langer (2015). "A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems". In: *Research and Advanced Technology for Digital Libraries*. Cham: Springer International Publishing, pp. 153–168. ISBN: 978-3-319-24592-8 (cit. on p. 19).

Nicholas J Belkin and W Bruce Croft (1992). "Information filtering and information retrieval: Two sides of the same coin?" In: *Communications of the ACM* 35.12, pp. 29–38 (cit. on p. 13).

F. Bellomi and R. Bonato (2005). "Network Analysis for Wikipedia". In: *Proceedings of Wikimania* (cit. on pp. 55, 67).

Iz Beltagy, Kyle Lo, and Arman Cohan (2019). "SciBERT: A Pretrained Language Model for Scientific Text". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 3613–3618. DOI: 10.18653/v1/D19-1371 (cit. on pp. 37, 97, 100, 101, 105, 106, 131, 132, 137, 142, 147, 149, 150).

Iz Beltagy, Matthew E. Peters, and Arman Cohan (2020). "Longformer: The Long-Document Transformer". In: DOI: 10.48550/arXiv.2004.05150 (cit. on pp. 5, 37, 83).

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel (May 2022). "BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 1–9. DOI: 10.18653/v1/2022.acl-short.1 (cit. on pp. 105, 111).

Kamal Berahmand, Elahe Nasiri, Mehrdad Rostami, and Saman Forouzandeh (Oct. 2021). "A Modified DeepWalk Method for Link Prediction in Attributed Social Network". In: *Computing* 103.10, pp. 2227–2249. ISSN: 1436-5057. DOI: 10.1007/s00607-021-00982-2 (cit. on p. 43).

Lucas Bernardi, Jaap Kamps, Julia Kiseleva, and Melanie JI Müller (Aug. 2015). "The Continuous Cold Start Problem in E-Commerce Recommender Systems". In: *2nd Workshop on New Trends in Content-Based Recommender Systems* (cit. on p. 18).

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen-tau Yih, Sebastian Riedel, and Fabio Petroni (Apr. 2022). *Autoregressive Search Engines: Generating Substrings as Document Identifiers*. DOI: 10.48550/arXiv.2204.10628 (cit. on p. 173).

Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar (2018). "Content-based citation recommendation". In: *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* 1, pp. 238–251. DOI: 10.18653/v1/n18-1022 (cit. on pp. 14, 20, 39, 100, 105, 106).

Paheli Bhattacharya, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Parth Mehta, Arnab Bhattacharya, Prasenjit Majumder (2019). "FIRE 2019 AILA Track: Artificial Intelligence for Legal Assistance". In: *Proceedings of the 11th Forum for Information Retrieval Evaluation*. Kolkata, India: Association for Computing Machinery, pp. 4–6. DOI: 10.1145/3368567.3368587 (cit. on p. 16).

Paheli Bhattacharya, Kripabandhu Ghosh, Arindam Pal, and Saptarshi Ghosh (2020a). "Methods for Computing Legal Document Similarity: A Comparative Study". In: *arXiv 2004.12307*. DOI: 10.48550/arXiv.2004.12307 (cit. on pp. 16, 94).

Paheli Bhattacharya, Parth Mehta, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Arnab Bhattacharya, Prasenjit Majumder (Dec. 2020b). "FIRE 2020 AILA Track: Artificial Intelligence for Legal Assistance". In: *Forum for Information Retrieval Evaluation*. Hyderabad India: ACM, pp. 1–3. DOI: 10.1145/3441501.3441510 (cit. on p. 16).

Steven Bird, Robert Dale, Bonnie J. Dorr, Bryan Gibson, Mark T. Joseph, Min Yen Kan, Dongwon Lee, Brett Powley, Dragomir R. Radev, and Yee Fan Tan (2008). "The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics". In: *Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008*, pp. 1755–1759 (cit. on pp. 134, 135).

David M. Blei, Andrew Y. Ng, and Michael I. Jordan (2003). "Latent Dirichlet Allocation". In: *Journal of Machine Learning Research* 3.Jan, pp. 993–1022 (cit. on pp. 16, 46, 216).

Donald S Blough (2001). "The perception of similarity". In: *Avian visual cognition* 6, pp. 23–25 (cit. on pp. 23, 24, 145).

Guido Boella, Luigi Di Caro, Llio Humphreys, Livio Robaldo, Piercarlo Rossi, and Leendert van der Torre (2016). "Eunomos, a legal document and knowledge management system for the Web to provide relevant, reliable and up-to-date information on the law". In: *Artificial Intelligence and Law* 24.3, pp. 245–283. ISSN: 1572-8382 (cit. on pp. 16, 29).

Alexander Boer and Radboud Winkels (2016). "Making a cold start in legal recommendation: An experiment". In: *Frontiers in Artificial Intelligence and Applications* 294, pp. 131–136. ISSN: 0922-6389. DOI: 10.3233/978-1-61499-726-9-131 (cit. on pp. 16, 80).

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov (2017). "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146. URL: http://arxiv.org/abs/1607.04606 (cit. on pp. 32, 82, 137, 147, 150).

Dheeraj Bokde, Sheetal Girase, and Debajyoti Mukhopadhyay (2015). "Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey". In: *Procedia Computer Science* 49, pp. 136–146. ISSN: 1877-0509. DOI: 10.1016/j.procs.2015.04.237 (cit. on p. 18).

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning (2015). "A large annotated corpus for learning natural language inference". In: *Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pp. 632–642. DOI: 10.18653/v1/d15-1075 (cit. on pp. 44, 49, 83).

Corinna Breitinger, Birkan Kolcu, Monique Meuschke, Norman Meuschke, and Bela Gipp (Aug. 2020). "Supporting the Exploration of Semantic Features in Academic Literature using Graph-based Visualizations". In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. Virtual Event, China. DOI: 10.1145/3383583.3398599 (cit. on p. 5).

Robin Brochier, Adrien Guille, and Julien Velcin (2019). "Global Vectors for Node Representations". In: *The World Wide Web Conference on - WWW '19*. Vol. 2. New York, New York, USA: ACM Press, pp. 2587–2593. DOI: 10.1145/3308558.3313595 (cit. on p. 100).

Jane Bromley, J.W. Bentz, Leon Bottou, I. Guyon, Yann Lecun, C. Moore, Eduard Sackinger, R. Shah (1993). "Signature verification using a Siamese time delay neural network". In: *International Journal of Pattern Recognition and Artificial Intelligence* 7.4. ISSN: 0302-2838 (cit. on pp. 83, 119, 123, 136, 151).

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. (2020). "Language models are few-shot learners". In: *Advances in Neural Information Processing Systems* 2020-Decem. ISSN: 1049-5258. URL: https://arxiv.org/abs/2005.14165 (cit. on pp. 5, 174).

Maxime Bucher, Stéphane Herbin, and Frédéric Jurie (2016). "Hard Negative Mining for Metric Learning Based Zero-Shot Classification". In: *Computer Vision – ECCV 2016 Workshops*. Cham: Springer International Publishing, pp. 524–531. ISBN: 978-3-319-49409-8. URL: https://arxiv.org/abs/1608.07441 (cit. on p. 102).

Alexander Budanitsky and Graeme Hirst (2006). "Evaluating WordNet-based Measures of Lexical Semantic Relatedness". In: *Computational Linguistics* 32.1, p. 36 (cit. on p. 23).

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld (Nov. 2020). "TLDR: Extreme Summarization of Scientific Documents". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 4766–4777. DOI: 10.18653/v1/2020.findings-emnlp.428 (cit. on p. 47).

Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang (Sept. 2018). "A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications". In: *IEEE Transactions on Knowledge and Data Engineering* 30.09, pp. 1616–1637. ISSN: 1558-2191. DOI: 10.1109/TKDE.2018.2807452 (cit. on p. 43).

Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao (2009). *Clueweb09 data set*. https://lemurproject.org/clueweb09/. URL: https://lemurproject.org/clueweb09/ (cit. on p. 138).

Jose Camacho-Collados and Mohammad Taher Pilehvar (Dec. 2018). "From Word To Sense Embeddings: A Survey on Vector Representations of Meaning". In: *Journal of Artificial Intelligence Research* 63, pp. 743–788. ISSN: 1076-9757. DOI: 10.1613/jair.1.11259 (cit. on pp. 30, 44, 47, 48, 145, 160).

Shuyang Cao and Lu Wang (June 2021). "Inference Time Style Control for Summarization". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 5942–5953. DOI: 10.18653/v1/2021.naacl-main.476 (cit. on pp. 44, 47).

Cornelia Caragea, Adrian Silvescu, Prasenjit Mitra, and C. Lee Giles (2013). "Can't See the Forest for the Trees?: A Citation Recommendation System". In: *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries - JCDL '13*. Indianapolis, Indiana, USA: ACM Press, p. 111. DOI: 10.1145/2467696.2467743 (cit. on p. 39).

Rudolf Carnap (1967). "The Logical Structure of the World & Pseudoproblems in Philosophy". In: p. 152 (cit. on p. 23).

Luca Cazzanti and Maya Gupta (July 2006). "Information-Theoretic and Set-theoretic Similarity". In: *2006 IEEE International Symposium on Information Theory*. Seattle, WA: IEEE, pp. 1836–1840. DOI: 10.1109/ISIT.2006.261752 (cit. on p. 26).

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia (2017). "SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vol. 371. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 1–14. DOI: 10.18653/v1/S17-2001 (cit. on p. 83).

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. (2018). "Universal Sentence Encoder". In: *arXiv:1803.11175*. DOI: 10.48550/arXiv.1803.11175 (cit. on pp. 14, 17, 104, 106).

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos (2020). "LEGAL-BERT: The Muppets straight out of Law School". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. i. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 2898–2904. DOI: 10.18653/v1/2020.findings-emnlp.261 (cit. on pp. 17, 37, 83).

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, Nikolaos Aletras (2022). "LexGLUE: A Benchmark Dataset for Legal Language Understanding in English". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 4310–4330. DOI: 10.18653/v1/2022.acl-long.297 (cit. on p. 16).

Branden Chan (2020). *CORD-19 BERT Model*. https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge/discussion/138250. (Visited on 06/30/2020) (cit. on pp. 37, 137).

Joel Chan, Joseph Chee Chang, Tom Hope, Dafna Shahaf, and Aniket Kittur (Nov. 2018). "SOLVENT: A Mixed Initiative System for Finding Analogies between Research Papers". In: *Proceedings of the ACM on Human-Computer Interaction*. Vol. 2. CSCW, pp. 1–21. DOI: 10.1145/3274300 (cit. on pp. 6, 15, 19, 44, 48, 49, 133, 136, 145, 146, 148, 149, 161, 162, 173).

Chaomei Chen (2017). "Science Mapping: A Systematic Review of the Literature". In: *Journal of Data and Information Science* 2, pp. 1–40 (cit. on p. 40).

Lei Chen, Ruifeng Xu, and Min Yang (2020). "Overview of the NLPCC 2020 Shared Task: Multi-aspect-based Multi-Sentiment Analysis (MAMS)". In: *Natural Language Processing and Chinese Computing*. Cham: Springer International Publishing, pp. 579–585. ISBN: 978-3-030-60457-8 (cit. on pp. 44, 46).

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel (2019a). "A Multi-Task Approach for Disentangling Syntax and Semantics in Sentence Representations". In: *Proceedings of the 2019 Conference of the North*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 2453–2464. DOI: 10.18653/v1/N19-1254 (cit. on pp. 44, 48).

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei (2018). "Neural Natural Language Inference Models Enhanced with External Knowledge". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 2406–2417. DOI: 10.18653/v1/P18-1224 (cit. on pp. 44, 49).

Xi Chen, Huan-jing Zhao, Shu Zhao, Jie Chen, and Yan-ping Zhang (Nov. 2019b). "Citation Recommendation Based on Citation Tendency". In: *Scientometrics* 121.2, pp. 937–956. ISSN: 0138-9130, 1588-2861. DOI: 10.1007/s11192-019-03225-6 (cit. on p. 43).

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. (Sept. 2016). "Wide & Deep Learning for Recommender Systems". In: *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. Boston MA USA: ACM, pp. 7–10. DOI: 10.1145/2988450.2988454 (cit. on p. 18).

Uthsav Chitra and Christopher Musco (2020). "Analyzing the Impact of Filter Bubbles on Social Network Polarization". In: *Proceedings of the 13th International Conference on Web Search and Data Mining* (cit. on p. 18).

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning (2020). "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators". In: *International Conference on Learning Representations*, pp. 1–18. URL: http://arxiv.org/abs/2003.10555 (cit. on pp. 37, 138).

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady (2019). "Structural Scaffolds for Citation Intent Classification in Scientific Publications". In: *Proceedings of the 2019 Conference of the North*. Vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 3586–3596. DOI: 10.18653/v1/N19-1361 (cit. on pp. 101, 107).

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld (2020). "SPECTER: Document-level Representation Learning using Citation-informed Transformers". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 2270–2282. DOI: 10.18653/v1/2020.acl-main.207 (cit. on pp. 18, 39, 97, 98, 100, 101, 103–106, 109, 134, 135, 143, 145, 147, 149, 150, 166, 170).

Andrew Collins and Joeran Beel (2019). "Document Embeddings vs. Keyphrases vs. Terms: An Online Evaluation in Digital Library Recommender Systems". In: *ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pp. 130–133. URL: http://arxiv.org/abs/1905.11244 (cit. on pp. 14, 20, 23, 29, 53).

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes (Sept. 2017). "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 670–680. DOI: 10.18653/v1/D17-1070 (cit. on p. 124).

Thomas H. Cormen (2009). *Introduction to Algorithms*. 3rd ed. Cambridge, Mass: MIT Press. ISBN: 978-0-262-53305-8 (cit. on p. 38).

Thomas M. Cover and Peter E. Hart (1967). "Nearest Neighbor Pattern Classification". In: *IEEE Transactions on Information Theory* 13.1, pp. 21–27. ISSN: 1557-9654 (cit. on p. 151).

Thomas M. Cover and Joy A. Thomas (2006). *Elements of Information Theory*. 2nd ed. Hoboken, N.J: Wiley-Interscience. ISBN: 978-0-471-24195-9 (cit. on p. 25).

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto (2013). *Recognizing Textual Entailment: Models and Applications*. Cham: Springer International Publishing. ISBN: 978-3-031-02151-0. DOI: 10.1007/978-3-031-02151-0 (cit. on pp. 44, 49).

Andrew M. Dai and Quoc V. Le (Nov. 2015). *Semi-Supervised Sequence Learning* (cit. on p. 35).

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov (2019). "Transformer-XL: Attentive Language Models beyond a Fixed-Length Context". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988. DOI: 10.18653/v1/P19-1285 (cit. on p. 124).

Hoa Trang Dang (2006). "DUC 2005: Evaluation of Question-Focused Summarization Systems". In: *Proceedings of the Workshop on Task-Focused Summarization and Question Answering - SumQA '06*. Sydney, Australia: Association for Computational Linguistics, p. 48. DOI: 10.3115/1654679.1654689 (cit. on pp. 44, 47).

James Davidson, Blake Livingston, Dasarathi Sampath, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, et al. (2010). "The YouTube Video Recommendation System". In: *Proceedings of the fourth ACM conference on Recommender systems - RecSys '10*, p. 293. ISSN: 1605-5890. DOI: 10.1145/1864708.1864770 (cit. on pp. 3, 8, 13, 20).

Marco De Gemmis, Pasquale Lops, Giovanni Semeraro, and Cataldo Musto (2015). "An investigation on the serendipity problem in recommender systems". In: *Information Processing and Management* 51.5, pp. 695–717. ISSN: 0306-4573. DOI: 10.1016/j.ipm.2015.06.008 (cit. on p. 54).

Jeffrey Dean and Sanjay Ghemawat (Jan. 2008). "MapReduce: Simplified Data Processing on Large Clusters". In: *Communications of the ACM* 51.1, pp. 107–113. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/1327452.1327492 (cit. on p. 58).

Lieven Decock and Igor Douven (Mar. 2011). "Similarity After Goodman". In: *Review of Philosophy and Psychology* 2.1, pp. 61–75. ISSN: 1878-5158, 1878-5166. DOI: 10.1007/s13164-010-0035-y (cit. on pp. 23, 25).

Yashar Deldjoo, Markus Schedl, Paolo Cremonesi, and Gabriella Pasi (Sept. 2020). "Recommender Systems Leveraging Multimedia Content". In: *ACM Comput. Surv.* 53.5. ISSN: 0360-0300. DOI: 10.1145/3407190 (cit. on p. 14).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (Oct. 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423 (cit. on pp. 4, 5, 35, 36, 82, 105, 106, 109, 118, 122, 125, 136–138, 145, 149, 150, 213).

Michel Marie Deza and Elena Deza (2013). *Encyclopedia of Distances*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-30958-8. DOI: 10.1007/978-3-642-30958-8 (cit. on p. 27).

Ying Ding, Jianfei Yu, and Jing Jiang (2017). "Recurrent Neural Networks with Auxiliary Labels for Cross-Domain Opinion Target Extraction". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. San Francisco, California, USA: AAAI Press, pp. 3436–3442 (cit. on pp. 44, 46).

Ying Ding, Guo Zhang, Tamy Chambers, Min Song, Xiaolong Wang, and Chengxiang Zhai (2014). "Content-based citation analysis: The next generation of citation analysis". In: *Journal of the Association for Information Science and Technology* 65.9, pp. 1820–1833. DOI: https://doi.org/10.1002/asi.23256 (cit. on p. 3).

Hai Ha Do, P. W.C. Prasad, Angelika Maag, and Abeer Alsadoon (2019). "Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review". In: *Expert Systems with Applications* 118, pp. 272–299. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2018.10.003 (cit. on pp. 44, 46).

William B. Dolan and Chris Brockett (2005). "Automatically Constructing a Corpus of Sentential Paraphrases". In: *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, pp. 9–16 (cit. on p. 119).

R. Dong, L. Tokarchuk, and A. Ma (2009). "Digging Friendship: Paper Recommendation in Social Network". In: *Proceedings of Networking & Electronic Commerce Research Conference (NAEC 2009)*, p. 7 (cit. on p. 39).

Nana Du, Jun Guo, Chase Q. Wu, Aiqin Hou, Zimin Zhao, and Daguang Gan (Nov. 2020). "Recommendation of Academic Papers Based on Heterogeneous Information Networks". In: *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*. Antalya, Turkey: IEEE, pp. 1–6. DOI: 10.1109/AICCSA50499.2020.9316516 (cit. on p. 14).

Daniel Duma and Ewan Klein (2014). "Citation Resolution: A Method for Evaluating Context-Based Citation Recommendation Systems". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 358–363. DOI: 10.3115/v1/P14-2059 (cit. on pp. 15, 29).

Travis Ebesu and Yi Fang (2017). "Neural citation network for context-Aware citation recommendation". In: *SIGIR 2017 - Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1093–1096. DOI: 10.1145/3077136.3080730 (cit. on p. 15).

Leo Egghe and Ronald Rousseau (1990). *Introduction to informetrics: Quantitative methods in library, documentation and information science*. Elsevier Science Publishers, p. 204 (cit. on p. 38).

Michael D. Ekstrand, Praveen Kannan, James A. Stemper, John T. Butler, Joseph A. Konstan, and John T. Riedl (2010). "Automatically Building Research Reading Lists". In: *Proceedings of the Fourth ACM Conference on Recommender Systems - RecSys '10*. Barcelona, Spain: ACM Press, p. 159. DOI: 10.1145/1864708.1864740 (cit. on pp. 14, 19).

Gil Elbaz (2007). *Common Crawl*. URL: http://commoncrawl.org (visited on 06/30/2020) (cit. on p. 138).

Aaron Elkiss, Siwei Shen, Anthony Fader, Güneş Erkan, David States, and Dragomir Radev (Jan. 2008). "Blind men and elephants: What do citation summaries tell us about a research article?" In: *Journal of the American Society for Information Science and Technology* 59.1, pp. 51–62. ISSN: 1532-2882. DOI: 10.1002/asi.20707 (cit. on p. 101).

David Ellis, Jonathan Furner-Hines, and Peter Willett (1993). "Measuring the Degree of Similarity Between Objects in Text Retrieval Systems". In: *Perspectives in Information Management* 3.2, pp. 128–149 (cit. on p. 145).

Mojisola Erdt, Alejandro Fernández, and Christoph Rensing (2015). "Evaluating Recommender Systems for Technology Enhanced Learning: A Quantitative Survey". In: *IEEE Transactions on Learning Technologies* 8, pp. 326–344 (cit. on pp. 19, 20).

Günes Erkan and Dragomir R. Radev (Dec. 2004). "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization". In: 22.1, pp. 457–479. ISSN: 1076-9757 (cit. on p. 47).

Francesco Fabbri, Yanhao Wang, Francesco Bonchi, Carlos Castillo, and Michael Mathioudakis (2022). "Rewiring What-to-Watch-Next Recommendations to Reduce Radicalization Pathways". In: *Proceedings of the ACM Web Conference 2022*. Virtual Event, Lyon, France: Association for Computing Machinery, pp. 2719–2728. DOI: 10.1145/3485447.3512143 (cit. on p. 18).

Kim Falk and Chen Karako (2022). "Optimizing Product Recommendations for Millions of Merchants". In: *Proceedings of the 16th ACM Conference on Recommender Systems*. Seattle, WA, USA: Association for Computing Machinery, pp. 499–501. DOI: 10.1145/3523227.3547393 (cit. on p. 20).

Angela Fan, David Grangier, and Michael Auli (July 2018). "Controllable Abstractive Summarization". In: *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. Melbourne, Australia: Association for Computational Linguistics, pp. 45–54. DOI: 10.18653/v1/W18-2706 (cit. on pp. 44, 47).

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie (2020). "CERT: Contrastive Self-supervised Learning for Language Understanding". In: *arXiv:2005.12766*, pp. 1–16. ISSN: 2331-8422. DOI: 10.36227/techrxiv.12308378 (cit. on p. 100).

Mehrdad Farajtabar, Jiachen Yang, Xiaojing Ye, Huan Xu, Rakshit S. Trivedi, Elias Boutros Khalil, Shuang Li, Le Song, Hongyuan Zha (2017). "Fake News Mitigation via Point Process Based Intervention". In: *ICML* (cit. on pp. 157, 158).

Michael Farber, Timo Klein, and Joan Sigloch (2020). "Neural Citation Recommendation: A Reproducibility Study". In: *BIRECIR 2020*, p. 9 (cit. on p. 15).

Michael Färber and Adam Jatowt (Dec. 2020). "Citation Recommendation: Approaches and Datasets". In: *International Journal on Digital Libraries* 21.4, pp. 375–405. ISSN: 1432-5012, 1432-1300. DOI: 10.1007/s00799-020-00288-2 (cit. on p. 15).

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith (2015). "Retrofitting Word Vectors to Semantic Lexicons". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 1606–1615. DOI: 10.3115/v1/N15-1184 (cit. on pp. 146, 150).

Zeshan Fayyaz, Mahsa Ebrahimian, Dina Nawara, Ahmed Ibrahim, and Rasha Kashef (2020). "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities". In: *Applied Sciences (Switzerland)* 10.21, pp. 1–20. ISSN: 2076-3417. DOI: 10.3390/app10217748 (cit. on p. 13).

Christiane Fellbaum (2010). "WordNet". In: *Theory and Applications of Ontology: Computer Applications*, pp. 231–243. ISBN: 978-90-481-8846-8. DOI: 10.1007/978-90-481-8847-5_10 (cit. on p. 17).

Jinzhan Feng, Shuqin Cai, and Xiaomeng Ma (May 2019a). "Enhanced Sentiment Labeling and Implicit Aspect Identification by Integration of Deep Convolution Neural Network and Sequential Algorithm". In: *Cluster Computing* 22.S3, pp. 5839–5857. ISSN: 1386-7857, 1573-7543. DOI: 10.1007/s10586-017-1626-5 (cit. on pp. 44, 46).

Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, Keping Yang (Aug. 2019b). "Deep Session Interest Network for Click-Through Rate Prediction". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. Macao, China: International Joint Conferences on Artificial Intelligence Organization, pp. 2301–2307. DOI: 10.24963/ijcai.2019/319 (cit. on p. 13).

Felice Ferrara, Nirmala Pudota, and Carlo Tasso (2011). "A Keyphrase-Based Paper Recommender System". In: *Communications in Computer and Information Science* 249 CCIS, pp. 14–25. ISSN: 1865-0929. DOI: 10.1007/978-3-642-27302-5_2 (cit. on p. 14).

João J. M. Ferreira, Cristina Isabel Fernandes, and Vanessa Ratten (2016). "A Co-Citation Bibliometric Analysis of Strategic Management Research". In: *Scientometrics* 109, pp. 1–32 (cit. on p. 40).

Besnik Fetahu, Katja Markert, and Avishek Anand (Oct. 2015). "Automated News Suggestions for Populating Wikipedia Entity Pages". In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. Melbourne Australia: ACM, pp. 323–332. DOI: 10.1145/2806416.2806531 (cit. on p. 16).

Tomáš Foltýnek, Norman Meuschke, and Bela Gipp (Oct. 2019). "Academic Plagiarism Detection: A Systematic Literature Review". In: *ACM Comput. Surv.* 52.6. ISSN: 0360-0300. DOI: 10.1145/3345317 (cit. on p. 4).

Antonino Freno (2017). "Practical Lessons from Developing a Large-Scale Recommender System at Zalando". In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*. Como, Italy: Association for Computing Machinery, pp. 251–259. DOI: 10.1145/3109859.3109897 (cit. on p. 20).

Lea Frermann and Alexandre Klementiev (2019). "Inducing Document Structure for Aspect-based Summarization". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 6263–6273. DOI: 10.18653/v1/P19-1630 (cit. on pp. 44, 47).

Jerome H. Friedman (2001). "Greedy Function Approximation: A Gradient Boosting Machine." In: *The Annals of Statistics* 29.5, pp. 1189–1232. DOI: 10.1214/aos/1013203451 (cit. on p. 43).

Sudeep Gandhe, Andrew S. Gordon, and David Traum (2006). "Improving question-answering with linking dialogues". In: *International Conference on Intelligent User Interfaces, Proceedings IUI* 2006, pp. 369–371. DOI: 10.1145/1111449.1111540 (cit. on p. 142).

Tianyu Gao, Xingcheng Yao, and Danqi Chen (2021). "SimCSE: Simple Contrastive Learning of Sentence Embeddings". In: *arXiv:2104.08821*. URL: http://arxiv.org/abs/2104.08821 (cit. on pp. 97, 100, 101, 111).

Peter Gardenfors (2004). *Conceptual spaces: The geometry of thought*. MIT press (cit. on pp. 25, 48, 146, 166).

Eugene Garfield (1972). "Citation Analysis as a Tool in Journal Evaluation: Journals Can Be Ranked by Frequency and Impact of Citations for Science Policy Studies." In: *Science (New York, N.Y.)* 178.4060, pp. 471–479 (cit. on p. 38).

Eugene Garfield (1955). "Citation Indexes for Science: A New Dimension in Documentation through Association of Ideas". In: *Science (New York, N.Y.)* 122.3159, pp. 108–111 (cit. on p. 38).

Eugene Garfield (2001). *From Bibliographic Coupling to Co-Citation Analysis Via Algorithmic Historio-Bibliography* (cit. on p. 4).

Eugene Garfield (May 1964). "Science Citation Index – A New Dimension in Indexing". In: *Science* 144.3619, pp. 649–654. ISSN: 0036-8075. DOI: 10.1126/science.144.3619.649 (cit. on p. 41).

Angel L. Garrido, Maria Soledad Pera, and Sergio Ilarri (July 2014). "SOLE-R: A Semantic and Linguistic Approach for Book Recommendations". In: *2014 IEEE 14th International Conference on Advanced Learning Technologies*. Athens, Greece: IEEE, pp. 524–528. DOI: 10.1109/ICALT.2014.155 (cit. on p. 17).

Angel Luis Garrido, Maria G. Buey, Sandra Escudero, Sergio Ilarri, Eduardo Mena, and Sara B. Silveira (Nov. 2013). "TM-Gen: A Topic Map Generator from Text Documents". In: *2013 IEEE 25th International Conference on*

Tools with Artificial Intelligence. Herndon, VA, USA: IEEE, pp. 735–740. DOI: 10.1109/ICTAI.2013.113 (cit. on p. 17).

Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach (2010). "Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity". In: *Proceedings of the Fourth ACM Conference on Recommender Systems*. Barcelona, Spain: Association for Computing Machinery, pp. 257–260. DOI: 10.1145/1864708.1864761 (cit. on pp. 22, 54, 76, 145, 160).

Yingqiang Ge, Shuya Zhao, Honglu Zhou, Changhua Pei, Fei Sun, Wenwu Ou, Yongfeng Zhang (July 2020). "Understanding Echo Chambers in E-commerce Recommender Systems". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2261–2270. DOI: 10.1145/3397271.3401431 (cit. on p. 18).

Mary L. Gick and Keith J. Holyoak (1983). "Schema induction and analogical transfer". In: *Cognitive Psychology* 15.1, pp. 1–38. ISSN: 0010-0285. DOI: 10.1016/0010-0285(83)90002-6 (cit. on p. 117).

C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence (1998). "CiteSeer: An Automatic Citation Indexing System". In: *Proceedings of the Third ACM Conference on Digital Libraries - DL '98*. Pittsburgh, Pennsylvania, United States: ACM Press, pp. 89–98. DOI: 10.1145/276675.276685 (cit. on p. 14).

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader (June 2021). "DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 879–895. DOI: 10.18653/v1/2021.acl-long.72 (cit. on pp. 100, 105, 106).

Bela Gipp and Joeran Beel (2009). "Citation Proximity Analysis (CPA) - A new approach for identifying related work based on Co-Citation Analysis". In: *Proceedings of the 12th International Conference on Scientometrics and Informetrics (ISSI '09:)* vol. 2. July, pp. 571–575. DOI: 10.1045/november2009-inbrief.URL (cit. on pp. 4, 40, 41, 58, 61, 77, 98, 214).

Goran Glavaš and Ivan Vulić (2018). "Explicit Retrofitting of Distributional Word Vectors". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 37. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 34–45 (cit. on pp. 146, 147, 150, 153, 159, 215).

Xavier Glorot, Antoine Bordes, and Yoshua Bengio (2011). "Deep Sparse Rectifier Neural Networks". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, pp. 315–323 (cit. on p. 29).

Aaron Gokaslan and Vanya Cohen (2019). *OpenWebText Corpus*. https://skylion007.github.io/OpenWebTextCorpus/ (cit. on p. 137).

David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry (Dec. 1992). "Using Collaborative Filtering to Weave an Information Tapestry". In: *Communications of the ACM* 35.12, pp. 61–70. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/138859.138867 (cit. on p. 18).

Wael Gomaa and Aly Fahmy (Apr. 2013). "A Survey of Text Similarity Approaches". In: *International Journal of Computer Applications* 68.13, pp. 13–18. ISSN: 0975-8887. DOI: 10.5120/11638-7118 (cit. on p. 27).

Hongyu Gong, Yelong Shen, Dian Yu, Jianshu Chen, and Dong Yu (2020). "Recurrent Chunking Mechanisms for Long-Text Machine Reading Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: ACL, pp. 6751–6761 (cit. on p. 49).

Nelson Goodman (1972). "Seven Strictures on Similarity". In: *Problems and Projects* (cit. on pp. 5, 23–25, 78, 117, 165, 166).

Palash Goyal and Emilio Ferrara (July 2018). "Graph Embedding Techniques, Applications, and Performance: A Survey". In: *Knowledge-Based Systems* 151, pp. 78–94. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2018.03.022 (cit. on p. 43).

Pietro Gravino, Bernardo Monechi, and Vittorio Loreto (2019). "Towards novelty-driven recommender systems". In: *Comptes Rendus Physique* 20.4, pp. 371–379. ISSN: 1631-0705. DOI: 10.1016/j.crhy.2019.05.014 (cit. on p. 54).

Aditya Grover and Jure Leskovec (2016). "Node2vec: Scalable Feature Learning for Networks". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*. New York, New York, USA: ACM Press, pp. 855–864. DOI: 10.1145/2939672.2939754 (cit. on pp. 16, 43, 86, 100).

Varun Grover, Seung Ryul Jeong, and Albert H. Segars (1996). "Information systems effectiveness: The construct space and patters of application". In: *Inf. Manag.* 31, pp. 177–191 (cit. on p. 19).

Lantian Guo, Xiaoyan Cai, Hao Hua Qin, Yang-ming Guo, Fei Li, and Gang Tian (2019). "Citation Recommendation with a Content-Sensitive DeepWalk Based Approach". In: *2019 International Conference on Data Mining Workshops (ICDMW)*, pp. 538–543 (cit. on p. 43).

Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, Sanjiv Kumar (2020). "Accelerating Large-Scale Inference with Anisotropic Vector Quantization". In: *International Conference on Machine Learning*. URL: https://arxiv.org/abs/1908.10396 (cit. on p. 162).

Rahul Gupta (May 2019). "Data Augmentation for Low Resource Sentiment Analysis Using Generative Adversarial Networks". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 2019-May. IEEE, pp. 7380–7384. ISBN: 978-1-4799-8131-1 (cit. on pp. 157, 158).

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, Noah A. Smith (July 2020). "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8342–8360. DOI: 10.18653/v1/2020.acl-main.740 (cit. on p. 37).

Michael U Gutmann and Aapo Hyvärinen (2012). "Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics". In: *Journal of Machine Learning Research* 13, pp. 307–361 (cit. on p. 31).

Raja Habib and Muhammad Tanvir Afzal (May 2019). "Sections-Based Bibliographic Coupling for Research Paper Recommendation". In: *Scientometrics* 119.2, pp. 643–656. ISSN: 0138-9130, 1588-2861. DOI: 10.1007/s11192-019-03053-8 (cit. on pp. 14, 15, 20).

Jialong Han, Yan Song, Wayne Xin Zhao, Shuming Shi, and Haisong Zhang (2018). "hyperdoc2vec: Distributed Representations of Hypertext Documents". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 2384–2394. DOI: 10.18653/v1/P18-1222 (cit. on pp. 100, 145).

Yi Han, Shanika Karunasekera, and Christopher Leckie (2020). "Graph Neural Networks with Continual Learning for Fake News Detection from Social Media". In: *ArXiv* abs/2007.03316 (cit. on pp. 157, 158).

Zellig S. Harris (Aug. 1954). "Distributional Structure". In: *WORD* 10.2-3, pp. 146–162. ISSN: 0043-7956. DOI: 10.1080/00437956.1954.11659520 (cit. on p. 30).

Hua He, Kevin Gimpel, and Jimmy Lin (2015). "Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1576–1586. DOI: 10.18653/v1/D15-1181 (cit. on pp. 8, 44, 49, 50).

Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, and Lee Giles (2010). "Context-aware citation recommendation". In: *Proceedings of the 19th international conference on World wide web - WWW '10*, p. 421. DOI: 10.1145/1772690.1772734 (cit. on pp. 15, 20, 39).

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua (Aug. 2017). "Neural Collaborative Filtering". In: *Proceedings of the 26th International Conference on World Wide Web*. arXiv, pp. 173–182. DOI: 10.1145/3038912.3052569 (cit. on p. 18).

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, Ray Kurzweil (May 2017). "Efficient Natural Language Response Suggestion for Smart Reply". In: *arXiv:1705.00652*. DOI: 10.48550/arXiv.1705.00652 (cit. on p. 151).

William Hersh, Andrew Turpin, Susan Price, Benjamin Chan, Dale Kramer, Lynetta Sacherek, Daniel Olson (2000). "Do Batch and User Evaluations Give the Same Results?" In: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Athens, Greece: Association for Computing Machinery, pp. 17–24. DOI: 10.1145/345508.345539 (cit. on p. 20).

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, Alexander Lerchner (Dec. 2018). "Towards a Definition of Disentangled Representations". In: *arXiv:1812.02230*. DOI: 10.48550/arXiv.1812.02230 (cit. on pp. 48, 146).

Sepp Hochreiter and Jürgen Schmidhuber (Nov. 1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735 (cit. on pp. 33, 137, 217).

Katja Hofmann, Krisztian Balog, Toine Bogers, and M. de Rijke (2010). "Contextual Factors for Finding Similar Experts". In: *Journal of the American Society for Information Science and Technology* 61 (cit. on pp. 8, 44, 45).

Thomas Hofmann (1999). "Probabilistic Latent Semantic Indexing". In: *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Berkeley, California, USA: Association for Computing Machinery, pp. 50–57. DOI: 10.1145/312624.312649 (cit. on p. 45).

Andreas Nugaard Holm, Barbara Plank, Dustin Wright, and Isabelle Augenstein (2022). "Longitudinal Citation Prediction using Temporal Graph Neural Networks". In: *AAAI 2022 Workshop on Scientific Document Understanding (SDU 2022)* (cit. on p. 100).

Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme (2020). "A dataset for statutory reasoning in tax law entailment and question answering". In: *Proceedings of the 2020 Natural Legal Language Processing Workshop*, pp. 31–38 (cit. on pp. 17, 37, 83).

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd (2020). "spaCy: Industrial-strength Natural Language Processing in Python". In: DOI: 10.5281/zenodo.1212303 (cit. on p. 137).

Benjamin D. Horne and Sibel Adali (2017). "This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News". In: *ArXiv* abs/1703.09398 (cit. on pp. 157, 158).

Jeremy Howard and Sebastian Ruder (July 2018). "Universal Language Model Fine-tuning for Text Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 328–339. DOI: 10.18653/v1/P18-1031 (cit. on p. 35).

Meiqun Hu, Ee-Peng Lim, Aixin Sun, Hady Wirawan Lauw, and Ba-Quy Vuong (2007). "Measuring Article Quality in Wikipedia". In: *Proceedings of the 16th ACM Conference on Information and Knowledge Management - CIKM '07*. DOI: 10.1145/1321440.1321476 (cit. on p. 64).

Minqing Hu and Bing Liu (2004). "Mining and Summarizing Customer Reviews". In: *KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 168–177. DOI: 10.1145/1014052.1014073 (cit. on pp. 44, 46, 47).

Ting-Hao Kenneth Huang, Chieh-Yang Huang, Chien-Kuang Cornelia Ding, Yen-Chia Hsu, and C. Lee Giles (July 2020). "CODA-19: Using a Non-Expert Crowd to Annotate Research Aspects on 10,000+ Abstracts in the COVID-19 Open Research Dataset". In: *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. Online: Association for Computational Linguistics. URL: https://aclanthology.org/2020.nlpcovid19-acl.6 (cit. on pp. 15, 49, 133, 136, 146, 162).

Wenyi Huang, Zhaohui Wu, Chen Liang, Prasenjit Mitra, and C Lee Giles (2015). "A Neural Probabilistic Model for Context Based Citation Recommendation". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2404–2410. DOI: 10.5555/2886521.2886655 (cit. on p. 15).

Sven Husmann, Antoniya Shivarova, and Rick Steinert (June 2022). "Company Classification Using Machine Learning". In: *Expert Syst. Appl.* 195.C. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2022.116598 (cit. on p. 158).

Paul Jaccard (Feb. 1912). "The Distribution of the Flora in the Alpine Zone". In: *New Phytologist* 11.2, pp. 37–50. ISSN: 0028-646X (cit. on p. 27).

Sarthak Jain, Edward Banner, Jan-Willem van de Meent, Iain J. Marshall, and Byron C. Wallace (2018). "Learning Disentangled Representations of Texts with Application to Biomedical Abstracts". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 4683–4693. DOI: 10.18653/v1/D18-1497 (cit. on pp. 20, 44, 48).

Dietmar Jannach and Gediminas Adomavicius (July 2017). "Price and Profit Awareness in Recommender Systems". In: arXiv:1707.08029. DOI: 10.48550/arXiv.1707.08029 (cit. on pp. 13, 19).

Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich (2010). *Recommender Systems - An Introduction*. 1st ed. United Kingdom: Cambridge University Press. ISBN: 978-0-521-49336-9 (cit. on p. 77).

Bo Jarneving (Jan. 2007). "Bibliographic Coupling and Its Application to Research-Front and Other Core Documents". In: *Journal of Informetrics* 1.4, pp. 287–307. ISSN: 1751-1577. DOI: 10.1016/j.joi.2007.07.004 (cit. on p. 39).

Chanwoo Jeong, Sion Jang, Hyuna Shin, Eunjeong Lucy Park, and Sungchul Choi (2020). "A context-aware citation recommendation model with BERT and graph convolutional networks". In: *Scientometrics*, pp. 1–16. DOI: 10.1007/s11192-020-03561-y (cit. on pp. 100, 109).

Yoo Kyung Jeong, Min Song, and Ying Ding (2014). "Content-Based Author Co-Citation Analysis". In: *Journal of Informetrics* 8.1, pp. 197–211. ISSN: 1751-1577. DOI: 10.1016/j.joi.2013.12.001 (cit. on p. 40).

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu (Feb. 2022). "A Survey on Knowledge Graphs: Representation, Acquisition and Applications". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.2, pp. 494–514. ISSN: 2162-237X, 2162-2388. DOI: 10.1109/TNNLS.2021.3070843 (cit. on p. 43).

Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li (July 2020). "A Re-visit of the Popularity Baseline in Recommender Systems". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Virtual Event China: ACM, pp. 1749–1752. DOI: 10.1145/3397271.3401233 (cit. on pp. 18, 172).

Ray Jiang, Silvia Chiappa, Tor Lattimore, András György, and Pushmeet Kohli (2019). "Degenerate Feedback Loops in Recommender Systems". In: *AIES 2019 - Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society* 2016, pp. 383–390. DOI: 10.1145/3306618.3314288 (cit. on pp. 18, 134–136).

Yichen Jiang, Aixia Jia, Yansong Feng, and Dongyan Zhao (2012). "Recommending Academic Papers via Users' Reading Purposes". In: *Proceedings of the Sixth ACM Conference on Recommender Systems - RecSys '12*. Dublin, Ireland: ACM Press, p. 241. DOI: 10.1145/2365952.2366004 (cit. on pp. 14, 15, 20).

Thorsten Joachims, Laura Granka, and Bing Pan (2005). "Accurately interpreting clickthrough data as implicit feedback". In: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. DOI: 10.1145/1076034.1076063 (cit. on p. 54).

Jeff Johnson, Matthijs Douze, and Herve Jegou (July 2021). "Billion-Scale Similarity Search with GPUs". In: *IEEE Transactions on Big Data* 7.3, pp. 535–547. ISSN: 2332-7790. DOI: 10.1109/TBDATA.2019.2921572 (cit. on pp. 105, 162).

Rob Johnson, Anthony Watkinson, and Michael Mabe (2018). *The STM Report: An Overview of Scientific and Scholarly Publishing* (cit. on p. 3).

Karen Sparck Jones (1972). "A Statistical Interpretation of Term Specificity and Its Application in Retrieval". In: *Journal of Documentation* 28.1, pp. 11–21. ISSN: 0022-0418. DOI: 10.1108/eb026526 (cit. on pp. 3, 28).

Karen Sparck Jones (Nov. 1973). "Index term weighting". In: *Information Storage and Retrieval* 9.11, pp. 619–633. ISSN: 0020-0271. DOI: 10.1016/0020-0271(73)90043-0 (cit. on p. 77).

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy (Jan. 2020). "SpanBERT: Improving Pre-Training by Representing and Predicting Spans". In: *Transactions of the Association for Computational Linguistics* 8, pp. 64–77. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00300 (cit. on p. 36).

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov (2017). "Bag of Tricks for Efficient Text Classification". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Stroudsburg, PA, USA: ACL, pp. 427–431 (cit. on p. 82).

Dan Jurafsky and James H. Martin (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2nd ed. Prentice Hall Series in Artificial Intelligence. Upper Saddle River, N.J: Pearson Prentice Hall. ISBN: 978-0-13-187321-6 (cit. on p. 27).

Marius Kaminskas and Derek Bridge (Mar. 2017). "Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems". In: *ACM Transactions on Interactive Intelligent Systems* 7.1, pp. 1–42. ISSN: 2160-6455. DOI: 10.1145/2926720 (cit. on pp. 13, 18, 54, 76).

Marius Kaminskas and Derek Bridge (2014). "Measuring Surprise in Recommender Systems". In: *RecSys REDD 2014: International Workshop on Recommender Systems Evaluation: Dimensions and Design* 69, pp. 2–7 (cit. on p. 54).

Anshul Kanakia, Zhihong Shen, Darrin Eide, and Kuansan Wang (2019). "A Scalable Hybrid Research Paper Recommender System for Microsoft Academic". In: *The World Wide Web Conference (WWW '19)*. New York, New York, USA: ACM Press, pp. 2893–2899. DOI: 10.1145/3308558.3313700 (cit. on pp. 14, 19, 20, 29).

Hyeonsu B. Kang, Sheshera Mysore, Kevin Huang, Haw-Shiuan Chang, Thorben Prein, Andrew McCallum, Aniket Kittur, Elsa Olivetti (2022). "Augmenting Scientific Creativity with Retrieval across Knowledge Domains". In: *arXiv:2206.01328*. DOI: 10.48550/ARXIV.2206.01328 (cit. on p. 6).

Marcin Kardas, Piotr Czapla, Pontus Stenetorp, Sebastian Ruder, Sebastian Riedel, Ross Taylor, Robert Stojnic (Nov. 2020). "AxCell: Automatic Extraction of Results from Machine Learning Papers". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 8580–8594. DOI: 10.18653/v1/2020.emnlp-main.692 (cit. on p. 149).

Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac (2018). "News Recommender Systems – Survey and Roads Ahead". In: *Information Processing and Management* 54.6, pp. 1203–1227. ISSN: 0306-4573. DOI: 10.1016/j.ipm.2018.04.008 (cit. on pp. 8, 13).

Maryam Karimzadehgan and ChengXiang Zhai (2009). "Constrained Multi-Aspect Expertise Matching for Committee Review Assignment". In: *Proceedings of the 18th ACM conference on Information and knowledge management* (cit. on pp. 44, 45).

Maryam Karimzadehgan, ChengXiang Zhai, and Geneva Belford (2008). "Multi-Aspect Expertise Matching for Review Assignment". In: *Proceeding of the 17th ACM Conference on Information and Knowledge Mining - CIKM '08*. Napa Valley, California, USA: ACM Press, p. 1113. DOI: 10.1145/1458082.1458230 (cit. on pp. 44, 45).

Yael Karov and Shimon Edelman (1998). "Similarity-based Word Sense Disambiguation". In: *Computational Linguistics* 24.1. ISSN: 0891-2017 (cit. on p. 142).

Parminder Kaur, Husanbir Singh Pannu, and Avleen Kaur Malhi (2021). "Comparative analysis on cross-modal information retrieval: A review". In: *Computer Science Review* 39, p. 100336. ISSN: 1574-0137. DOI: 10.1016/j.cosrev.2020.100336 (cit. on p. 101).

Myer M. Kessler (Jan. 1963). "Bibliographic coupling between scientific papers". In: *American Documentation* 14.1, pp. 10–25. ISSN: 0096-946X. DOI: 10.1002/asi.5090140103 (cit. on pp. 4, 38, 39, 101).

Zahid Younas Khan, Zhendong Niu, Sulis Sandiwarno, and Rukundo Prince (2021). "Deep learning techniques for rating prediction: a survey of the state-of-the-art". In: *Artificial Intelligence Review* 54.1, pp. 95–135 (cit. on p. 18).

Christopher S. G. Khoo and Jin-Cheon Na (Sept. 2007). "Semantic relations in information science". In: *Annual Review of Inf. Science and Technology* 40.1, pp. 157–228. ISSN: 0066-4200. DOI: 10.1002/aris.1440400112 (cit. on p. 117).

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, Dilip Krishnan (2020). "Supervised Contrastive Learning". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 18661–18673 (cit. on p. 100).

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura (Nov. 2016). "Controlling Output Length in Neural Encoder-Decoders". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1328–1338. DOI: 10.18653/v1/D16-1140 (cit. on pp. 44, 47).

Ha Jin Kim, Yoo Kyung Jeong, and Min Song (2016). "Content- and Proximity-Based Author Co-Citation Analysis Using Citation Sentences". In: *Journal of Informetrics* 10.4, pp. 954–966. ISSN: 1751-1577. DOI: 10.1016/j.joi.2016.07.007 (cit. on p. 40).

Jooyeon Kim, Dongkwan Kim, and Alice H. Oh (2019). "Homogeneity-Based Transmissive Process to Model True and False News in Social Networks". In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (cit. on p. 158).

Jooyeon Kim, Behzad Tabibian, Alice H. Oh, Bernhard Schölkopf, and Manuel Gomez-Rodriguez (2018). "Leveraging the Crowd to Detect and Reduce the Spread of Fake News and Misinformation". In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (cit. on p. 158).

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee (Aug. 2021). "Self-Guided Contrastive Learning for BERT Sentence Representations". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 2528–2540. DOI: 10.18653/v1/2021.acl-long.197 (cit. on p. 100).

Young Jin Kim and Hany Hassan Awadalla (Oct. 2020). *FastFormers: Highly Efficient Transformer Models for Natural Language Understanding* (cit. on p. 37).

Diederik P. Kingma and Jimmy Lei Ba (2015). "Adam: A method for stochastic optimization". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15 (cit. on p. 105).

Bart P. Knijnenburg (2012). "Conducting User Experiments in Recommender Systems". In: *Proceedings of the Sixth ACM Conference on Recommender Systems - RecSys '12*. Dublin, Ireland: ACM Press, p. 3. DOI: 10.1145/2365952.2365956 (cit. on p. 19).

Yuta Kobayashi, Masashi Shimbo, and Yuji Matsumoto (May 2018). "Citation Recommendation Using Distributed Representation of Discourse Facets in Scientific Articles". In: *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*. New York, NY, USA: ACM, pp. 243–251. DOI: 10.1145/3197026.3197059 (cit. on pp. 15, 49, 146, 162).

Lasse Kohlmeyer, Tim Repke, and Ralf Krestel (Dec. 2021). "Novel Views on Novels:Embedding Multiple Facets of Long Texts". In: *IEEE/WIC/ACM International Conference on Web Intelligence*. ESSENDON VIC Australia: ACM, pp. 670–675. DOI: 10.1145/3486622.3494006 (cit. on pp. 44, 49).

Xiangjie Kong, Mengyi Mao, Wei Wang, Jiaying Liu, and Bo Xu (2018). "VOPRec: Vector Representation Learning of Papers with Text Information and Structural Identity for Recommendation". In: *IEEE Transactions on Emerging Topics in Computing* May. ISSN: 2168-6750. DOI: 10.1109/TETC.2018.2830698 (cit. on p. 14).

Yehuda Koren, Robert Bell, and Chris Volinsky (Aug. 2009). "Matrix Factorization Techniques for Recommender Systems". In: *Computer* 42.8, pp. 30–37. ISSN: 0018-9162. DOI: 10.1109/MC.2009.263 (cit. on p. 18).

Yehuda Koren, Steffen Rendle, and Robert Bell (2022). "Advances in collaborative filtering". In: *Recommender systems handbook*, pp. 91–142 (cit. on p. 18).

Denis Kotkov, Jari Veijalainen, and Shuaiqiang Wang (2020). "How Does Serendipity Affect Diversity in Recommender Systems? A Serendipity-Oriented Greedy Algorithm". In: *Computing* 102.2, pp. 393–411. ISSN: 1436-5057. DOI: 10.1007/s00607-018-0687-5 (cit. on p. 18).

Christin Katharina Kreutz and Ralf Schenkel (Oct. 2022). "Scientific paper recommendation systems: a literature review of recent publications". In: *International Journal on Digital Libraries*. DOI: 10.1007/s00799-022-00339-w (cit. on pp. 14, 20).

Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá (2010). "Hyperbolic geometry of complex networks". In: *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 82.3, pp. 1–18. ISSN: 1539-3755 (cit. on p. 83).

Kundan Krishna and Balaji Vasan Srinivasan (2018). "Generating Topic-Oriented Summaries Using Neural Attention". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 1697–1705. DOI: 10.18653/v1/N18-1153 (cit. on pp. 44, 47).

Tejas D Kulkarni, William F. Whitney, Pushmeet Kohli, and Josh Tenenbaum (2015). "Deep Convolutional Inverse Graphics Network". In: *Advances in Neural Information Processing Systems*. Vol. 28. Curran Associates, Inc. (cit. on p. 49).

Sushanta Kumar, P. Krishna Reddy, V. Balakista Reddy, and Aditya Singh (2011). "Similarity analysis of legal judgments". In: *Compute 2011 - 4th Annual ACM Bangalore Conference*. COMPUTE '11. DOI: 10.1145/1980422.1980439 (cit. on pp. 16, 29, 80, 82).

Matevž Kunaver and Tomaž Požrl (2017). "Diversity in recommender systems - A survey". In: *Knowledge-Based Systems* 123, pp. 154–162. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2017.02.009 (cit. on pp. 13, 54, 145).

Johannes Kunkel, Tim Donkers, Lisa Michael, Catalin Mihai Barbu, and Jürgen Ziegler (2019). "Let me explain: Impact of personal and impersonal explanations on trust in recommender systems". In: *Conference on Human Factors in Computing Systems - Proceedings*. DOI: 10.1145/3290605.3300717 (cit. on pp. 6, 77, 160, 174).

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut (2020). "ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations". In: *International Conference on Learning Representations* (cit. on p. 36).

J. Richard Landis and Gary G. Koch (1977). "The Measurement of Observer Agreement for Categorical Data". In: *Biometrics* 33.1, pp. 159–174. ISSN: 0006-341X. DOI: 10.2307/2529310 (cit. on p. 74).

Jörg Landthaler, Bernhard Waltl, Patrick Holl, and Florian Matthes (2016). "Extending full text search for legal document collections using word embeddings". In: *Frontiers in Artificial Intelligence and Applications* 294, pp. 73–82. ISSN: 0922-6389 (cit. on p. 16).

Steven A. Lastres (2013). *Rebooting Legal Research in a Digital Age*. Tech. rep. LexisNexis. URL: https://www.lexisnexis.com/documents/pdf/20130806061418%5C%5Flarge.pdf (cit. on p. 79).

Quoc V. Le and Tomas Mikolov (2014). "Distributed Representations of Sentences and Documents". In: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32, pp. 1188–1196 (cit. on pp. 3, 33, 82, 104, 106, 118, 122, 136, 145, 215).

Mark Ledwich and Anna Zaitsev (Feb. 2020). "Algorithmic Extremism: Examining YouTube's Rabbit Hole of Radicalization". In: *First Monday* 25.3. DOI: 10.5210/fm.v25i3.10419 (cit. on p. 18).

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, Jaewoo Kang (Sept. 2019). "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics*, pp. 1–8. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz682 (cit. on pp. 37, 100, 105, 106, 137, 142).

Philip Lenhart and Daniel Herzog (2016). "Combining content-based and collaborative filtering for personalized sports news recommendations". In: *Proceedings of the 3rd Workshop on New Trends in Content-Based RecommenderSystems (CBRecSys '16) at RecSys'16*. Vol. 1673, pp. 3–10 (cit. on p. 77).

Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, Alex Peysakhovich (Mar. 2019). "PyTorch-BigGraph: A Large-scale Graph Embedding System". In: *Proceedings of The Conference on Systems and Machine Learning*. DOI: 10.48550/arXiv.1903.12287 (cit. on p. 102).

Gondy Leroy (2011). *Designing user studies in informatics*. Springer Science & Business Media (cit. on p. 19).

Vladimir I. Levenshtein (1965). "Binary codes capable of correcting deletions, insertions, and reversals". In: *Soviet physics. Doklady* 10, pp. 707–710 (cit. on p. 27).

Charlton T. Lewis (1891). *An elementary Latin dictionary*. New York: Harper & Brothers. URL: http://www.perseus.tufts.edu/hopper/resolveform?type=exact&lookup=aspectus&lang=latin (cit. on pp. 7, 44).

Michael Ley (Aug. 2009). "DBLP: some lessons learned". In: *Proceedings of the VLDB Endowment* 2.2, pp. 1493–1500. ISSN: 2150-8097. DOI: 10.14778/1687553.1687577 (cit. on p. 135).

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li (2020). "On the Sentence Embeddings from Pre-trained Language Models". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 9119–9130. DOI: 10.18653/v1/2020.emnlp-main.733 (cit. on pp. 93, 97, 101).

Guanghui Li, Jiawei Luo, Qiu Xiao, Cheng Liang, Pingjian Ding, and Buwen Cao (2017). "Predicting MicroRNA-Disease Associations Using Network Topological Similarity Based on DeepWalk". In: *IEEE access : practical innovations, open solutions* 5, pp. 24032–24039 (cit. on p. 43).

Jiwei Li and Dan Jurafsky (2015). "Do Multi-Sense Embeddings Improve Natural Language Understanding?" en. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1722–1732. (Visited on 02/09/2021) (cit. on p. 146).

Jundong Li, Liang Wu, Ruocheng Guo, Chenghao Liu, and Huan Liu (Aug. 2019). "Multi-level network embedding with boosted low-rank matrix approximation". In: *Proceedings of the 2019 IEEE/ACM International Confer-*

ence on Advances in Social Networks Analysis and Mining. New York, NY, USA: ACM, pp. 49–56. ISBN: 9781450368681 (cit. on pp. 43, 83).

Zhi Li and Xiaozhu Zou (Jan. 2019). "A Review on Personalized Academic Paper Recommendation". In: *Computer and Information Science* 12.1, p. 33. ISSN: 1913-8997, 1913-8989. DOI: 10.5539/cis.v12n1p33 (cit. on p. 14).

Keng-te Liao, Pochun Chen, Kuansan Wang, and Shou-de Lin (2020a). "Explainable and Sparse Representations of Academic Articles for Knowledge Exploration". In: *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6207–6216 (cit. on p. 161).

Keng-Te Liao, Cheng-Syuan Lee, Zhong-Yu Huang, and Shou-de Lin (Dec. 2020b). "Explaining Word Embeddings via Disentangled Representation". In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pp. 720–725 (cit. on pp. 44, 48).

Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades (2014). "Facing the cold start problem in recommender systems". In: *Expert Systems with Applications* 41.4, Part 2, pp. 2065–2073. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2013.09.005 (cit. on pp. 4, 18).

Dekang Lin (1998). "An Information-Theoretic Definition of Similarity". In: *Proceedings of ICML*, pp. 296–304. ISSN: 1-55860-556-8. DOI: 10.1.1.55.1832 (cit. on pp. 25, 26).

Jimmy Lin and W John Wilbur (Jan. 2007). "PubMed Related Articles: A Probabilistic Topic-Based Model for Content Similarity." In: *BMC bioinformatics* 8, p. 423. ISSN: 1471-2105. DOI: 10.1186/1471-2105-8-423 (cit. on pp. 3, 14, 20, 23).

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu (June 2021). "A Survey of Transformers". In: *arXiv:2106.04554*. DOI: 10.48550/arXiv.2106.04554 (cit. on p. 36).

Carolyn E. Lipscomb (2000). "Medical Subject Headings (MeSH)." In: *Bulletin of the Medical Library Association* 88 3, pp. 265–6 (cit. on p. 103).

Bing Liu (2012). *Sentiment Analysis and Opinion Mining*. Cham: Springer. ISBN: 978-3-031-02145-9 (cit. on pp. 44, 46).

Haoyue Liu, Ishani Chatterjee, MengChu Zhou, Xiaoyu Sean Lu, and Abdullah Abusorrah (2020). "Aspect-Based Sentiment Analysis: A Survey of Deep Learning Methods". In: *IEEE Transactions on Computational Social Systems* 7.6, pp. 1358–1375. DOI: 10.1109/TCSS.2020.3033302 (cit. on pp. 44, 46).

Shengbo Liu and Chaomei Chen (2011). "The effects of co-citation proximity on co-citation analysis". In: *In: Proceedings of the 13th Conference of the International Society for Scientometrics and Informetrics (ISSI' 11)* (cit. on p. 40).

Ya'ning Liu, Rui Yan, and Hongfei Yan (2013). "Guess What You Will Cite: Personalized Citation Recommendation Based on Users' Preference". In: *Information Retrieval Technology*. Vol. 8281. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 428–439. ISBN: 978-3-642-45068-6. DOI: 10.1007/978-3-642-45068-6_37 (cit. on p. 15).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *arXiv:1907.11692*. DOI: 10.48550/arXiv.1907.11692 (cit. on pp. 5, 36, 83, 86, 137).

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld (2020). "S2ORC: The Semantic Scholar Open Research Corpus". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 4969–4983. DOI: 10.18653/v1/2020.acl-main.447 (cit. on pp. 102, 135).

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, Olivier Bachem (June 2019). "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations". In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. PMLR, pp. 4114–4124 (cit. on p. 49).

Christoph Lofi and Nava Tintarev (2017). "Towards analogy-based recommendation: Benchmarking of perceived analogy semantics". In: *CEUR Workshop Proceedings* 1892, pp. 9–13. ISSN: 1613-0073 (cit. on p. 117).

Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro (2011). "Content-Based Recommender Systems: State of the Art and Trends". In: *Recommender Systems Handbook*. Boston, MA: Springer US, pp. 73–105. ISBN: 978-0-387-85820-3. DOI: 10.1007/978-0-387-85820-3_3 (cit. on p. 14).

Ilya Loshchilov and Frank Hutter (2019). "Decoupled weight decay regularization". In: *7th International Conference on Learning Representations, ICLR 2019* (cit. on p. 105).

Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang (2015). "Recommender System Application Developments: A Survey". In: *Decision Support Systems* 74, pp. 12–32. ISSN: 0167-9236. DOI: 10.1016/j.dss.2015.03.008 (cit. on p. 13).

H. P. Luhn (1957). "A Statistical Approach to Mechanized Encoding and Searching of Literary Information". In: *IBM Journal of Research and Development* 1.4, pp. 309–317. DOI: 10.1147/rd.14.0309 (cit. on p. 28).

Yin-Jyun Luo, Kat Agres, and Dorien Herremans (2019). "Learning Disentangled Representations of Timbre and Pitch for Musical Instrument Sounds Using Gaussian Mixture Variational Autoencoders". In: *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, the Netherlands, November 4-8, 2019*, pp. 746–753 (cit. on p. 49).

Kelvin Luu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A. Smith (2020). "Citation Text Generation". In: *arXiv*. ISSN: 2331-8422 (cit. on p. 174).

Kelvin Luu, Xinyi Wu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A. Smith (2021). "Explaining Relationships Between Scientific Documents". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 2130–2144. DOI: 10.18653/v1/2021.acl-long.166 (cit. on p. 100).

Shutian Ma, Chengzhi Zhang, and Xiaozhong Liu (2020). "A review of citation recommendation: from textual content to enriched context". In: *Scientometrics* January. ISSN: 1588-2861. DOI: 10.1007/s11192-019-03336-0 (cit. on pp. 14, 15).

Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu, Min Zhang, and Shaoping Ma (2021a). "Retrieving Legal Cases from a Large-scale Candidate Corpus". In: *Proceedings of COLIEE 2021 workshop: Competition on Legal Information Extraction/Entailment (COLIEE 2021)*. New York, NY, USA: ACM, p. 5 (cit. on p. 17).

Yixiao Ma, Yunqiu Shao, Yueyue Wu, Yiqun Liu, Ruizhe Zhang, Min Zhang, Shaoping Ma (July 2021b). "LeCaRD: A Legal Case Retrieval Dataset for Chinese Law System". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Virtual Event Canada: ACM, pp. 2342–2348. DOI: 10.1145/3404835.3463250 (cit. on p. 16).

Mounica Maddela, Mayank Kulkarni, and Daniel Preotiuc-Pietro (May 2022). "EntSUM: A Data Set for Entity-Centric Extractive Summarization". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 3355–3366. DOI: 10.18653/v1/2022.acl-long.237 (cit. on pp. 44, 47).

Andrii Maksai, Florent Garcin, and Boi Faltings (Sept. 2015). "Predicting Online Performance of News Recommender Systems Through Richer Evaluation Metrics". In: *Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, pp. 179–186. DOI: 10.1145/2792838.2800184 (cit. on p. 54).

Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli (2017). "Embedding Words and Senses Together via Joint Knowledge-Enhanced Training". In: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. 2. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 100–111. DOI: 10.18653/v1/K17-1012 (cit. on pp. 8, 44, 48).

Arpan Mandal, Raktim Chaki, Sarbajit Saha, Kripabandhu Ghosh, Arindam Pal, and Saptarshi Ghosh (2017). "Measuring Similarity among Legal Court Case Documents". In: *Proceedings of Compute '17*, pp. 1–9. ISBN: 9781450353236 (cit. on pp. 16, 23, 80, 86).

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze (Jan. 2008). *Introduction to Information Retrieval*. Vol. 16. Cambridge: Cambridge University Press Cambridge, England. ISBN: 978-0-511-80907-1. DOI: 10.1017/CBO9780511809071 (cit. on pp. 13, 21).

Iv Marshakova (1973). "System of Document Connections Based on References". In: *Scientific and Technical Information Serial of VINITI* 6 (cit. on pp. 38, 40).

John Martyn (Apr. 1964). "Bibliographic Coupling". In: *Journal of Documentation* 20.4, pp. 236–236. DOI: 10.1108/eb026352 (cit. on p. 39).

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger (2018). "UMAP: Uniform Manifold Approximation and Projection". In: *The Journal of Open Source Software* 3.29, p. 861 (cit. on p. 156).

Sean M McNee, Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K Lam, Al Mamunur Rashid, Joseph A Konstan, John Riedl (2002). "On the recommending of citations for research papers". In: *Proceedings of the 2002 ACM conference on Computer supported cooperative work - CSCW '02*. New York, New York, USA: ACM Press, p. 116. DOI: 10.1145/587078.587096 (cit. on p. 18).

Douglas L. Medin, Robert L. Goldstone, and Dedre Gentner (1993). "Respects for Similarity". In: *Psychological Review* 100.2, pp. 254–278. ISSN: 0033-295X. DOI: 10.1037/0033-295X.100.2.254 (cit. on p. 24).

David Mellinkoff (1963). "The language of the law". In: *Boston: Little Brown and Company* (cit. on p. 79).

Marcelo Mendoza and Nicolás Torres (2020). "Evaluating Content Novelty in Recommender Systems". In: *Journal of Intelligent Information Systems* 54.2, pp. 297–316. ISSN: 1573-7675. DOI: 10.1007/s10844-019-00548-x (cit. on pp. 13, 54).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean (2013a). "Distributed Representations of Words and Phrases and Their Compositionality". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pp. 3111–3119. DOI: 10.5555/2999792.299995 (cit. on pp. 31, 136).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean (2013b). "Efficient Estimation of Word Representations in Vector Space". In: pp. 1–12. URL: http://arxiv.org/abs/1301.3781 (cit. on pp. 31–33, 43, 119, 220).

Akshay Minocha, Navjyoti Singh, and Arjit Srivastava (2015). "Finding Relevant Indian Judgments using Dispersion of Citation Network". In: *Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion*. New York, New York, USA: ACM Press, pp. 1085–1088. ISBN: 9781450334730 (cit. on p. 16).

Maryam Mirzaei, Jörg Sander, and Eleni Stroulia (May 2019). "Multi-Aspect Review-Team Assignment Using Latent Research Areas". In: *Information Processing & Management* 56.3, pp. 858–878. ISSN: 0306-4573. DOI: 10.1016/j.ipm.2019.01.007 (cit. on pp. 44–46).

Hebatallah A. Mohamed Hassan, Giuseppe Sansonetti, Fabio Gasparetti, Alessandro Micarelli, and Joeran Beel (2019). "BERT, ELMo, USE and InferSent Sentence Encoders: The Panacea for Research-Paper Recommendation?" In: *CEUR Workshop Proceedings*. Vol. 2431, pp. 6–10 (cit. on pp. 14, 20, 85).

Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M. Bronstein (2019). "Fake News Detection on Social Media using Geometric Deep Learning". In: *ArXiv* abs/1902.06673. DOI: 10.48550/arXiv.1902.06673 (cit. on pp. 157, 158).

Frederic Morin and Yoshua Bengio (Jan. 2005). "Hierarchical Probabilistic Neural Network Language Model". In: *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*. Vol. R5. PMLR, pp. 246–252 (cit. on p. 31).

Kevin Musgrave, Serge J. Belongie, and Ser-Nam Lim (2020). "A Metric Learning Reality Check". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXV*, pp. 681–699. DOI: 10.1007/978-3-030-58595-2_41 (cit. on pp. 99, 100, 111).

Sheshera Mysore, Arman Cohan, and Tom Hope (2022). "Multi-Vector Models with Textual Guidance for Fine-Grained Scientific Document Similarity". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, pp. 4453–4470. DOI: 10.18653/v1/2022.naacl-main.331 (cit. on pp. 23, 44, 49).

Sheshera Mysore, Tim O'Gorman, Andrew McCallum, and Hamed Zamani (2021). "CSFCube – A Test Collection of Computer Science Research Articles for Faceted Query by Example". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. Vol. 1. Association for Computing Machinery (cit. on pp. 8, 44).

Sebastian Nagel (2016). *Common Crawl News*. URL: http://commoncrawl.org/2016/10/news-dataset-available/ (visited on 06/20/2020) (cit. on p. 137).

Rohan Nanda, Giovanni Siragusa, Luigi Di Caro, Guido Boella, Lorenzo Grossio, Marco Gerbaudo, Francesco Costamagna (2019). "Unsupervised and supervised text similarity systems for automated identification of national implementing measures of European directives". In: *Artificial Intelligence and Law* 27.2, pp. 199–225. ISSN: 1572-8382. DOI: 10.1007/s10506-018-9236-y (cit. on pp. 16, 82).

Sharan Narang, Hyung Won Chung, Yi Tay, Liam Fedus, Thibault Fevry, Michael Matena, Karishma Malkan, Noah Fiedel, Noam Shazeer, Zhenzhong Lan, et al. (Nov. 2021). "Do Transformer Modifications Transfer Across Implementations and Applications?" In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 5758–5773. DOI: 10.18653/v1/2021.emnlp-main.465 (cit. on p. 36).

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, et al. (2021). *Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM*. Vol. 1. Association for Computing Machinery. ISBN: 978-1-4503-8442-1. DOI: 10.1145/3458817.3476209 (cit. on p. 37).

Arpit Narechania, Alireza Karduni, Ryan Wesslen, and Emily Wall (Jan. 2022). "VITALITY: Promoting Serendipitous Discovery of Academic Literature with Transformers & Visual Analytics". In: *IEEE Transactions on Visualization and Computer Graphics* 28.1, pp. 486–496. ISSN: 1077-2626. DOI: 10.1109/TVCG.2021.3114820 (cit. on p. 145).

Cristiano Nascimento, Alberto H.F. Laender, Altigran S. da Silva, and Marcos André Gonçalves (2011). "A Source Independent Framework for Research Paper Recommendation". In: *Proceeding of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries - JCDL '11*. Ottawa, Ontario, Canada: ACM Press, p. 297. DOI: 10.1145/1998076.1998132 (cit. on p. 14).

Roberto Navigli and Simone Paolo Ponzetto (2012). "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network". In: *Artificial Intelligence* 193, pp. 217–250. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2012.07.001 (cit. on p. 48).

Ambreen Nazir, Yuan Rao, Lianwei Wu, and Ling Sun (2022). "Issues and Challenges of Aspect-Based Sentiment Analysis: A Comprehensive Survey". In: *IEEE Transactions on Affective Computing* 13.2, pp. 845–863. DOI: 10.1109/TAFFC.2020.2970399 (cit. on pp. 44, 46).

Yiu-Kai Ng (2016). "Recommending Books for Children Based on the Collaborative and Content-Based Filtering Approaches". In: *Computational Science and Its Applications – ICCSA 2016*. Vol. 9789. Cham: Springer International Publishing, pp. 302–317. ISBN: 978-3-319-42089-9. DOI: 10.1007/978-3-319-42089-9_22 (cit. on p. 17).

Dai Quoc Nguyen, Dat Quoc Nguyen, Ashutosh Modi, Stefan Thater, and Manfred Pinkal (2017). "A Mixture Model for Learning Multi-Sense Word Embeddings". In: *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*. Vancouver, Canada: Association for Computational Linguistics, pp. 121–127. DOI: 10.18653/v1/S17-1015 (cit. on pp. 8, 44, 48).

Dong Nguyen, Dolf Trieschnigg, and Mariët Theune (2014a). "Using Crowdsourcing to Investigate Perception of Narrative Similarity". In: *CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management*, pp. 321–330. DOI: 10.1145/2661829.2661918 (cit. on pp. 44, 49).

Huy-Tien Nguyen, Quan-Hoang Vo, and Minh-Le Nguyen (Nov. 2018). "A Deep Learning Study of Aspect Similarity Recognition". In: *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*. Ho Chi Minh City: IEEE, pp. 181–186. DOI: 10.1109/KSE.2018.8573326 (cit. on pp. 44, 50).

Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan (2014b). "Exploring the Filter Bubble: The Effect of Using Recommender Systems on Content Diversity". In: *Proceedings of the 23rd international conference on World wide web - WWW '14*. New York, New York, USA: ACM Press, pp. 677–686. DOI: 10.1145/2566486.2568012 (cit. on p. 160).

Maximilian Nickel and Douwe Kiela (2017). "Poincaré embeddings for learning hierarchical representations". In: *Advances in Neural Information Processing Systems* 2017.Nips, pp. 6339–6348. ISSN: 1049-5258 (cit. on p. 83).

Douglas W Oard and Jinmook Kim (1998). "Implicit Feedback for Recommender Systems". In: *Proceedings of the AAAI Workshop on Recommender System*. Vol. 83, p. 3 (cit. on p. 18).

Yann Ollivier and Pierre Senellart (2007). "Finding Related Pages Using Green Measures : An Illustration with Wikipedia". In: *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pp. 1427–1433. ISBN: 1577353234 (cit. on pp. 53, 119).

Daniela Onita, Liviu P. Dinu, and Adriana Birlutiu (2019). "From Image to Text in Sentiment Analysis via Regression and Deep Learning". In: *RANLP* (cit. on p. 158).

Marc van Opijnen and Cristiana Santos (2017). "On the concept of relevance in legal information retrieval". In: *Artificial Intelligence and Law* 25.1, pp. 65–87. ISSN: 1572-8382 (cit. on p. 80).

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd (1998). "The PageRank Citation Ranking". In: *World Wide Web Internet And Web Information Systems*. Vol. 54. ISBN: 1581138741. URL: http://ilpubs.stanford.edu:8090/422 (cit. on p. 39).

Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He (Apr. 2019). "Warm Up Cold-start Advertisements". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR'19*. New York, New York, USA: ACM Press, pp. 695–704. DOI: 10.1145/3331184.3331268 (cit. on p. 13).

Vedant Parikh, Upal Bhattacharya, Parth Mehta, Ayan Bandyopadhyay, Paheli Bhattacharya, Kripa Ghosh, Saptarshi Ghosh, Arindam Pal, Arnab Bhattacharya, and Prasenjit Majumder (Dec. 2021). "AILA 2021: Shared Task on Artificial Intelligence for Legal Assistance". In: *Forum for Information Retrieval Evaluation*. Virtual Event India: ACM, pp. 12–15. DOI: 10.1145/3503162.3506571 (cit. on p. 16).

Eli Pariser (2011). *The filter bubble: What the Internet is hiding from you*. Penguin UK (cit. on p. 18).

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda (2011). *English Gigaword Fifth Edition*. URL: https://catalog.ldc.upenn.edu/LDC2011T07 (cit. on p. 138).

Simon Pasternack (1969). "The Scientific Enterprise: Public Knowledge. An Essay Concerning the Social Dimension of Science". In: *Science (New York, N.Y.)* 164.3880, pp. 669–670. DOI: 10.1126/science.164.3880.669 (cit. on pp. 38, 39, 98).

Michael J. Pazzani and Daniel Billsus (2007). "Content-Based Recommendation Systems". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4321 LNCS, pp. 325–341. ISSN: 0302-9743. DOI: 10.1007/978-3-319-29659-3_4 (cit. on p. 13).

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830 (cit. on pp. 82, 124).

Hao Peng, Jing Liu, and Chin-Yew Lin (2016). "News Citation Recommendation with Implicit and Explicit Semantics". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers). Berlin, Germany: Association for Computational Linguistics, pp. 388–398. DOI: 10.18653/v1/P16-1037 (cit. on p. 16).

Jeffrey Pennington, Richard Socher, and Christopher Manning (2014). "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Stroudsburg, PA, USA: ACL, pp. 1532–1543. DOI: 10.3115/v1/D14-1162 (cit. on pp. 31, 82, 118, 122, 136).

Bryan Perozzi, Vivek Kulkarni, Haochen Chen, and Steven Skiena (July 2017). "Don't Walk, Skip!: Online Learning of Multi-scale Network Embeddings". In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. New York, NY, USA: ACM, pp. 258–265. ISBN: 9781450349932 (cit. on pp. 38, 43, 83).

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena (Mar. 2014). "DeepWalk: online learning of social representations". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*. New York, New York, USA: ACM Press, pp. 701–710. DOI: 10.1145/2623330.2623732 (cit. on pp. 38, 42, 43, 83, 100, 102).

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer (2018). "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 2227–2237. DOI: 10.18653/v1/N18-1202 (cit. on pp. 35, 145).

Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Dmytro Okhonko, Samuel Broscheit, Gautier Izacard, Patrick Lewis, Barlas Oğuz, Edouard Grave, Wen-tau Yih, et al. (2021). "The Web Is Your Oyster - Knowledge-Intensive NLP against a Very Large Web Corpus". In: DOI: 10.48550/ARXIV.2112.09924 (cit. on p. 16).

Mohammad Taher Pilehvar and Nigel Collier (2016). "De-Conflated Semantic Representations". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 1680–1690. DOI: 10.18653/v1/D16-1174 (cit. on pp. 146, 160).

Rajesh Piryani, Vedika Gupta, and Vivek Kumar Singh (Apr. 2017). "Movie Prism: A Novel System for Aspect Level Sentiment Profiling of Movies". In: *Journal of Intelligent & Fuzzy Systems* 32.5, pp. 3297–3311. ISSN: 1064-1246. DOI: 10.3233/JIFS-169272 (cit. on pp. 44, 46).

Joseph Polifroni, Lynette Hirschman, Stephanie Seneff, and Victor Zue (1992). "Experiments in evaluating interactive spoken language systems". In: *Proceedings of the workshop on Speech and Natural Language - HLT '91*. Morristown, NJ, USA: Association for Computational Linguistics, p. 28. DOI: 10.3115/1075527.1075533 (cit. on pp. 141, 142).

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphee De Clercq, et al. (2016). "SemEval-2016 Task 5: Aspect Based Sentiment Analysis". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 19–30. DOI: 10.18653/v1/S16-1002 (cit. on pp. 44, 46, 165).

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos (2015). "SemEval-2015 Task 12: Aspect Based Sentiment Analysis". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, pp. 486–495. DOI: 10.18653/v1/S15-2082 (cit. on pp. 44, 46).

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar (2014). "SemEval-2014 Task 4: Aspect Based Sentiment Analysis". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. SemEval. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 27–35. DOI: 10.3115/v1/S14-2004 (cit. on pp. 44, 46).

Soujanya Poria, Erik Cambria, Alexander Gelbukh, Federica Bisio, and Amir Hussain (Nov. 2015). "Sentiment Data Flow Analysis by Means of Dynamic Linguistic Patterns". In: *IEEE Computational Intelligence Magazine* 10.4, pp. 26–36. ISSN: 1556-603X. DOI: 10.1109/MCI.2015.2471215 (cit. on pp. 44, 46).

Soujanya Poria, Iti Chaturvedi, Erik Cambria, and Federica Bisio (July 2016). "Sentic LDA: Improving on LDA with Semantic Similarity for Aspect-Based Sentiment Analysis". In: *2016 International Joint Conference on Neural Networks (IJCNN)*. Vancouver, BC, Canada: IEEE, pp. 4465–4473. DOI: 10.1109/IJCNN.2016.7727784 (cit. on pp. 44, 46).

Jason Portenoy, Marissa Radensky, Jevin West, Eric Horvitz, Daniel Weld, and Tom Hope (2021). *Bursting Scientific Filter Bubbles: Boosting Innovation via Novel Author Discovery*. Vol. 1. 1. Association for Computing Machinery. ISBN: 9781450391573. DOI: 10.1145/3491102.3501905 (cit. on p. 145).

Tribikram Pradhan and Sukomal Pal (June 2020). "A Multi-Level Fusion Based Decision Support System for Academic Collaborator Recommendation". In: *Knowledge-Based Systems* 197, p. 105784. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2020.105784 (cit. on pp. 44, 45).

Derek J. De Solla Price (1965). "Networks of Scientific Papers: The Pattern of Bibliographic References Indicates the Nature of the Scientific Research Front." In: *Science (New York, N.Y.)* 149.3683, pp. 510–515. DOI: 10.1126/science.149.3683.510 (cit. on p. 38).

Pearl Pu, Li Chen, and Rong Hu (2011). "A user-centric evaluation framework for recommender systems". In: *Proceedings of the fifth ACM conference on Recommender systems - RecSys '11*. New York, New York, USA: ACM Press, p. 157. DOI: 10.1145/2043932.2043962 (cit. on p. 61).

Juliano Rabelo, Randy Goebel, Mi-Young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh (Apr. 2022). "Overview and Discussion of the Competition on Legal Information Extraction/Entailment (COLIEE) 2021". In: *The Review of Socionetwork Strategies* 16.1, pp. 111–133. ISSN: 2523-3173, 1867-3236. DOI: 10.1007/s12626-022-00105-z (cit. on pp. 16, 17).

Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh (2020). "A Summary of the COLIEE 2019 Competition". In: *New Frontiers in Artificial Intelligence*. Vol. 12331. Cham: Springer International Publishing, pp. 34–49. ISBN: 978-3-030-58790-1. DOI: 10.1007/978-3-030-58790-1_3 (cit. on p. 16).

Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh (2021). "COLIEE 2020: Methods for Legal Document Retrieval and Entailment". In: *New Frontiers in Artificial Intelligence*. Vol. 12758. Cham: Springer International Publishing, pp. 196–210. ISBN: 978-3-030-79942-7. DOI: 10.1007/978-3-030-79942-7_13 (cit. on pp. 16, 17).

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). "Language Models are Unsupervised Multitask Learners". In: *OpenAI blog*. URL: https://www.techbooky.com/wp-content/uploads/2019/02/Better-Language-Models-and-Their-Implications.pdf (cit. on pp. 5, 35).

Md. Khaledur Rahman and Ariful Azad (Dec. 2021). *A Comprehensive Analytical Survey on Unsupervised and Semi-Supervised Graph Representation Learning Methods* (cit. on p. 43).

Radim Rehurek and Petr Sojka (May 2010). "Software Framework for Topic Modelling with Large Corpora". In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, pp. 45–50 (cit. on pp. 33, 82, 121, 124, 215).

Nils Reimers and Iryna Gurevych (2017). "Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 338–348. DOI: 10.18653/v1/D17-1035 (cit. on p. 138).

Nils Reimers and Iryna Gurevych (2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 3980–3990. DOI: 10.18653/v1/D19-1410 (cit. on pp. 83, 84, 105, 106, 123, 124, 127, 136, 137, 143, 147, 151, 159).

S. Renuka, G. S. S. Raj Kiran, and Palakodeti Rohit (2021). "An Unsupervised Content-Based Article Recommendation System Using Natural Language Processing". In: *Data Intelligence and Cognitive Informatics*. Singapore: Springer Singapore, pp. 165–180. ISBN: 9789811585302. DOI: 10.1007/978-981-15-8530-2_13 (cit. on pp. 15, 29).

Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl (1994). "GroupLens: An Open Architecture for Collaborative Filtering of Netnews". In: *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work - CSCW '94*. Chapel Hill, North Carolina, United States: ACM Press, pp. 175–186. DOI: 10.1145/192844.192905 (cit. on pp. 3, 18).

Philip Resnik (1995). "Using Information Content to Evaluate Semantic Similarity in a Taxonomy". In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 448–453. ISBN: 1-55860-363-8 (cit. on pp. 23, 25).

Nils Rethmeier and Isabelle Augenstein (2021a). "A Primer on Contrastive Pretraining in Language Processing: Methods, Lessons Learned and Perspectives". In: *arXiv:2102.12982* (cit. on pp. 97, 99, 111).

Nils Rethmeier and Isabelle Augenstein (2021b). "Data-Efficient Pretraining via Contrastive Self-Supervision". In: *arXiv:2102.12982* (cit. on p. 97).

Leonardo F.R. Ribeiro, Pedro H.P. Saverese, and Daniel R. Figueiredo (Aug. 2017). "Struc2vec: Learning Node Representations from Structural Identity". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Halifax NS Canada: ACM, pp. 385–394. DOI: 10.1145/3097983.3098061 (cit. on pp. 14, 43).

Francesco Ricci, Lior Rokach, and Bracha Shapira (2011). "Introduction to Recommender Systems Handbook". In: *Recommender Systems Handbook*. Boston, MA: Springer US, pp. 1–35. ISBN: 978-0-387-85820-3. DOI: 10.1007/978-0-387-85820-3_1 (cit. on pp. 13, 19).

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl (May 2020). "Adapt or Get Left behind: Domain Adaptation through BERT Language Model Finetuning for Aspect-Target Sentiment Classification". In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 4933–4941. ISBN: 979-10-95546-34-4 (cit. on pp. 44, 46).

Julian Risch, Philipp Hager, and Ralf Krestel (2021). "Multifaceted Domain-Specific Document Embeddings". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 78–83. DOI: 10.18653/v1/2021.naacl-demos.9 (cit. on pp. 8, 44, 48).

Peter Gordon Roetzel (2018). "Information overload in the information age: a review of the literature from business administration, business psychology, and related disciplines with a bibliometric approach and framework development". In: *Business Research*, pp. 1–44 (cit. on p. 3).

Anna Rogers, Olga Kovaleva, and Anna Rumshisky (2020). "A Primer in BERTology: What We Know About How BERT Works". In: *Trans. Association Comput. Linguistics* 8, pp. 842–866. URL: https://transacl.org/ojs/index.php/tacl/article/view/2257 (cit. on p. 97).

Peter H Rossi, Mark W Lipsey, and Gary T Henry (2018). *Evaluation: A systematic approach*. Sage publications (cit. on p. 19).

Susan Rothstein (June 2016). "Aspect". In: *The Cambridge Handbook of Formal Semantics*. First. Cambridge University Press, pp. 342–368. ISBN: 978-1-139-23615-7. DOI: 10.1017/CBO9781139236157.013 (cit. on p. 44).

Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar (2020). "An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs". In: (cit. on p. 84).

Terry Ruas, Charles Henrique Porto Ferreira, William Grosky, Fabrício Olivetti de França, and Débora Maria Rossi de Medeiros (2020). "Enhanced Word Embeddings Using Multi-Semantic Representation through Lexical Chains". In: *Information Sciences* 532, pp. 16–32. ISSN: 0020-0255. DOI: 10.1016/j.ins.2020.04.048 (cit. on p. 146).

Terry Ruas, William Gorsky, and Akiko Aizawa (2019). "Multi-sense embeddings through a word sense disambiguation process". In: *Expert Systems with Applications* 136, pp. 288–303. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2019.06.026 (cit. on p. 146).

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin (2016). "INSIGHT-1 at SemEval-2016 Task 5: Deep Learning for Multilingual Aspect-based Sentiment Analysis". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 330–336. DOI: 10.18653/v1/S16-1053 (cit. on pp. 44, 46).

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams (Oct. 1986). "Learning Representations by Back-Propagating Errors". In: *Nature* 323.6088, pp. 533–536. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/323533a0 (cit. on pp. 30, 33).

Belen Saldias and Deb Roy (2020). "Exploring Aspects of Similarity between Spoken Personal Narratives by Disentangling Them into Narrative Clause Types". In: *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 78–86. DOI: 10.18653/v1/2020.nuse-1.10 (cit. on pp. 44, 49).

Gerard Salton (Oct. 1963). "Associative Document Retrieval Techniques Using Bibliographic Information". In: *Journal of the ACM* 10.4, pp. 440–457. ISSN: 0004-5411 (cit. on p. 145).

Gerard Salton (1989). "Automatic text processing: The transformation, analysis, and retrieval of". In: *Reading: Addison-Wesley* 169 (cit. on p. 13).

Gerard Salton (1971). *The SMART Retrieval System—Experiments in Automatic Document Processing*. USA: Prentice-Hall, Inc. (cit. on pp. 28, 219).

Gerard Salton, Andrew Wong, and Chungshu Yang (Nov. 1975). "A vector space model for automatic indexing". In: *Communications of the ACM* 18.11, pp. 613–620. ISSN: 0001-0782. DOI: 10.1145/361219.361220 (cit. on p. 3).

Mark Sanderson, Monica Lestari Paramita, Paul D. Clough, and E. Kanoulas (2010). "Do user preferences and evaluation measures line up?" In: *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (cit. on p. 20).

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar (June 2019). "A Theoretical Analysis of Contrastive Unsupervised Representation Learning". In: *ICML*. Vol. 97. PMLR. URL: http://proceedings.mlr.press/v97/saunshi19a.html (cit. on pp. 100, 102, 108).

J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen (2007). "Collaborative Filtering Recommender Systems". In: *The Adaptive Web*. Vol. 4321. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 291–324. DOI: 10.1007/978-3-540-72079-9_9 (cit. on p. 18).

J. Ben Schafer, Joseph Konstan, and John Riedl (Nov. 1999). "Recommender Systems in E-Commerce". In: *Proceedings of the 1st ACM Conference on Electronic Commerce*, p. 9. DOI: 10.1145/336992.337035 (cit. on p. 13).

Kim Schouten and Flavius Frasincar (2016). "Survey on Aspect-Level Sentiment Analysis". In: *IEEE Transactions on Knowledge and Data Engineering* 28.3, pp. 813–830. DOI: 10.1109/TKDE.2015.2485209 (cit. on pp. 44, 46).

Florian Schroff, Dmitry Kalenichenko, and James Philbin (2015). "FaceNet: A unified embedding for face recognition and clustering". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682 (cit. on pp. 99, 101).

Robert Schwarzenberg, Lisa Raithel, and David Harbecke (2019). "Neural Vector Conceptualization for Word Vector Space Interpretation". In: *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations*. Stroudsburg, PA, USA: ACL, pp. 1–7 (cit. on pp. 44, 48, 161).

Abigail See, Peter J. Liu, and Christopher D. Manning (2017). "Get to the Point: Summarization with Pointer-Generator Networks". In: *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)* 1, pp. 1073–1083. DOI: 10.18653/v1/P17-1099 (cit. on pp. 44, 47).

Guy Shani and Asela Gunawardana (2011). "Evaluating Recommendation Systems". In: *Recommender Systems Handbook*. Boston, MA: Springer US, pp. 257–297. ISBN: 978-0-387-85820-3. DOI: 10.1007/978-0-387-85820-3_8 (cit. on p. 19).

Yunqiu Shao and Ziyi Ye (2019). "THUIRAILA 2019: Information Retrieval Approaches for Identifying Relevant Precedents and Statutes". In: *FIRE2019AILA*, p. 6 (cit. on p. 16).

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro (2019). "Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism". In: (cit. on p. 37).

Connor Shorten, Taghi M. Khoshgoftaar, and Borko Furht (2021). "Text Data Augmentation for Deep Learning". In: *J. Big Data* 8.1, p. 101. DOI: 10.1186/s40537-021-00492-0 (cit. on p. 97).

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Paul Hsu, Kuansan Wang (2015). "An Overview of Microsoft Academic Service (MAS) and Applications". In: *Proceedings of the 24th International Conference on World Wide Web*. DOI: 10.1145/2740908.2742839 (cit. on p. 103).

Rashmi Sinha and Kirsten Swearingen (2002). "The role of transparency in recommender systems". In: *CHI '02 extended abstracts on Human factors in computing systems - CHI '02*. New York, New York, USA: ACM Press, p. 830. ISBN: 1581134541 (cit. on p. 75).

Henry Small (July 1973). "Co-citation in the scientific literature: A new measure of the relationship between two documents". In: *Journal of the American Society for Information Science* 24.4, pp. 265–269. ISSN: 0002-8231. DOI: 10.1002/asi.4630240406 (cit. on pp. 4, 38, 40, 101).

Brent Smith and Greg Linden (May 2017). "Two Decades of Recommender Systems at Amazon.Com". In: *IEEE Internet Computing* 21.3, pp. 12–18. ISSN: 1089-7801. DOI: 10.1109/MIC.2017.72 (cit. on pp. 3, 8, 13).

Linda C Smith (1981). "Citation Analysis". In: *Library Trends* 30, pp. 83–106. ISSN: 1933-8244. DOI: 10.1080/19338244.2010.483622 (cit. on p. 38).

Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng (2013). "Zero-Shot Learning through Cross-Modal Transfer". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*. Lake Tahoe, Nevada: Curran Associates Inc., pp. 935–943 (cit. on p. 101).

Karen Sparck Jones, S. Walker, and S. E. Robertson (2000). "A probabilistic model of information retrieval: development and comparative experiments, Part 2". In: *Information Processing and Management* 36.6, pp. 809–840. DOI: 10.1016/S0306-4573(00)00015-7 (cit. on pp. 42, 58).

Xiaoyuan Su and Taghi M. Khoshgoftaar (Oct. 2009). "A Survey of Collaborative Filtering Techniques". In: *Advances in Artificial Intelligence* 2009, pp. 1–19. ISSN: 1687-7470, 1687-7489. DOI: 10.1155/2009/421425 (cit. on p. 18).

Chi Sun, Luyao Huang, and Xipeng Qiu (June 2019). "Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 380–385. DOI: 10.18653/v1/N19-1035 (cit. on pp. 44, 46, 50, 85).

Veronica Symons (1991). "A review of information systems evaluation: content, context and process". In: *European Journal of Information Systems* 1, pp. 205–212 (cit. on p. 19).

Bin Tan and Fuchun Peng (2008). "Unsupervised query segmentation using generative language models and wikipedia". In: *Proceeding of the 17th international conference on World Wide Web - WWW '08*. New York, New York, USA: ACM Press, p. 347. DOI: 10.1145/1367497.1367545 (cit. on p. 142).

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei (2015). "LINE: Large-scale Information Network Embedding". In: *Proceedings of the 24th International Conference on World Wide Web - WWW '15*. New York, New York, USA: ACM Press, pp. 1067–1077. DOI: 10.1145/2736277.2741093 (cit. on pp. 43, 145).

Wenbin Tang, Jie Tang, and Chenhao Tan (Aug. 2010). "Expertise Matching via Constraint-Based Optimization". In: *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. Toronto, AB, Canada: IEEE, pp. 34–41. DOI: 10.1109/WI-IAT.2010.133 (cit. on pp. 44, 45).

Min Tao, Xinmin Yang, Gao Gu, and Bohan Li (2020). "Paper Recommend Based on LDA and PageRank". In: *Artificial Intelligence and Security*. Vol. 1254. Singapore: Springer Singapore, pp. 571–584. ISBN: 9789811581014. DOI: 10.1007/978-981-15-8101-4_51 (cit. on pp. 15, 20).

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler (Mar. 2022a). *Efficient Transformers: A Survey* (cit. on p. 36).

Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. (Feb. 2022b). *Transformer Memory as a Differentiable Search Index* (cit. on p. 173).

Simone Teufel, Advaith Siddharthan, and Dan Tidhar (2006). "Automatic classification of citation function". In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing - EMNLP '06*. Morristown, NJ, USA: Association for Computational Linguistics, p. 103. DOI: 10.3115/1610075.1610091 (cit. on pp. 38, 39, 102).

Mike Thelwall and David Wilkinson (2004). "Finding Similar Academic Web Sites with Links, Bibliometric Couplings and Colinks". In: *Information Processing and Management* 40.3, pp. 515–526. ISSN: 0306-4573. DOI: 10.1016/S0306-4573(03)00042-6 (cit. on p. 38).

Yonglong Tian, Dilip Krishnan, and Phillip Isola (2020a). "Contrastive Representation Distillation". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=SkgpBJrtvS (cit. on p. 101).

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola (2020b). "What Makes for Good Views for Contrastive Learning?" In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Vancouver, BC, Canada: Curran Associates Inc. ISBN: 9781713829546 (cit. on pp. 97, 99).

Anastasios Tombros and C. J. Van Rijsbergen (2001). "Query-Sensitive similarity measures for the calculation of interdocument relationships". In: *International Conference on Information and Knowledge Management, Proceedings*, pp. 17–24. DOI: 10.1145/502586.502589 (cit. on p. 161).

Nam Tran, Pedro Alves, Shuangge Ma, and Michael Krauthammer (2009). "Enriching PubMed Related Article Search with Sentence Level". In: *AMIA Annu Symp Proceedings 2009 Nov 14*, pp. 650–654. URL: https://pubmed.ncbi.nlm.nih.gov/20351935/ (cit. on pp. 40, 53).

Vu Tran, Minh Le Nguyen, and Ken Satoh (June 2019). "Building Legal Case Retrieval Systems with Lexical Matching and Summarization Using A Pre-Trained Phrase Scoring Model". In: *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*. Montreal QC Canada: ACM, pp. 275–282. DOI: 10.1145/3322640.3326740 (cit. on p. 16).

Trieu H. Trinh and Quoc V. Le (2018). "A Simple Method for Commonsense Reasoning". In: *arXiv:1806.02847*. DOI: 10.48550/arXiv.1806.02847 (cit. on p. 137).

Keita Tsuji, Nobuya Takizawa, Sho Sato, Ui Ikeuchi, Atsushi Ikeuchi, Fuyuki Yoshikane, Hiroshi Itsumura (Aug. 2014). "Book Recommendation Based on Library Loan Records and Bibliographic Information". In: *Procedia - Social and Behavioral Sciences* 147, pp. 478–486. ISSN: 1877-0428. DOI: 10.1016/j.sbspro.2014.07.142 (cit. on pp. 17, 19, 29).

Andrew H. Turpin and William Hersh (2001). "Why Batch and User Evaluations Do Not Give the Same Results". In: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New Orleans, Louisiana, USA: Association for Computing Machinery, pp. 225–231. DOI: 10.1145/383952.383992 (cit. on p. 20).

Amos Tversky (1977). "Features of Similarity". In: *Psychological review* 84.4, p. 327. DOI: 10.1037/0033-295X.84.4.327 (cit. on pp. 24–26, 50, 166).

Cornelis J. Van Rijsbergen (1979). *Information Retrieval*. 2d ed. London ; Boston: Butterworths. ISBN: 978-0-408-70929-3 (cit. on pp. 3, 14).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin (June 2017). "Attention Is All You Need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000–6010. DOI: 10.1017/CBO9780511809071 (cit. on pp. 34, 35, 100, 122, 136, 145, 219).

Paula Cristina Vaz, David Martins de Matos, Bruno Martins, and Pavel Calado (Mar. 2012). "Improving an Hybrid Literary Book Recommendation System through Author Ranking". In: *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '12)* (cit. on p. 17).

Maksims Volkovs, Guangwei Yu, and Tomi Poutanen (2017). "DropoutNet: Addressing Cold Start in Recommender Systems". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach, California, USA: Curran Associates Inc., pp. 4964–4973. ISBN: 9781510860964 (cit. on p. 4).

Denny Vrandecic and Markus Krötzsch (2014). "Wikidata: a free collaborative knowledgebase". In: *Commun. ACM* 57, pp. 78–85 (cit. on p. 118).

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, Hannaneh Hajishirzi (2020). "Fact or Fiction: Verifying Scientific Claims". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 7534–7550. DOI: 10.18653/v1/2020.emnlp-main.609 (cit. on p. 100).

Rupali S. Wagh and Deepa Anand (2020). "Legal document similarity: A multicriteria decision-making perspective". In: *PeerJ Computer Science* 2020.3, pp. 1–20. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.262 (cit. on pp. 16, 29).

Robert A. Wagner and Michael J. Fischer (Jan. 1974). "The String-to-String Correction Problem". In: *Journal of The Acm* 21.1, pp. 168–173. ISSN: 0004-5411. DOI: 10.1145/321796.321811 (cit. on p. 27).

Jan Philip Wahle, Terry Ruas, Tomáš Foltýnek, Norman Meuschke, and Bela Gipp (2022). "Identifying Machine-Paraphrased Plagiarism". In: *Information for a Better World: Shaping the Global Future*. Cham: Springer International Publishing, pp. 393–413. ISBN: 978-3-030-96957-8 (cit. on p. 5).

Jan Philip Wahle, Terry Ruas, Norman Meuschke, and Bela Gipp (2021). "Incorporating Word Sense Disambiguation in Neural Language Models". In: *arXiv:2106.07967*. DOI: 10.48550/arXiv.2106.07967 (cit. on p. 146).

Lipeng Wan, Xuwei Song, Xuguang Lan, and Nanning Zheng (2020). "Multi-agent Policy Optimization with Approximatively Synchronous Advantage Estimation". In: *arXiv:2012.03488*. DOI: 10.48550/arXiv.2012.03488 (cit. on p. 158).

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (2019). "Glue: A multi-task benchmark and analysis platform for natural language understanding". In: *7th International Conference on Learning Representations, ICLR 2019*, pp. 353–355. DOI: 10.18653/v1/w18-5446 (cit. on pp. 119, 130).

Chi Wang, Kaushik Chakrabarti, Tao Cheng, and Surajit Chaudhuri (2012). "Targeted disambiguation of ad-hoc, homogeneous sets of named entities". In: *WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web*, pp. 719–728. DOI: 10.1145/2187836.2187934 (cit. on p. 142).

Chong Wang and David M. Blei (2011). "Collaborative Topic Modeling for Recommending Scientific Articles". In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '11*. San Diego, California, USA: ACM Press, p. 448. DOI: 10.1145/2020408.2020480 (cit. on p. 14).

Fan Wang, Ning Shi, and Ben Chen (July 2010). "A Comprehensive Survey of the Reviewer Assignment Problem". In: *International Journal of Information Technology & Decision Making* 09.04, pp. 645–668. ISSN: 0219-6220, 1793-6845. DOI: 10.1142/S0219622010003993 (cit. on p. 45).

Hao Wang, Naiyan Wang, and Dit-Yan Yeung (2015). "Collaborative Deep Learning for Recommender Systems". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Sydney, NSW, Australia: Association for Computing Machinery, pp. 1235–1244. DOI: 10.1145/2783258.2783273 (cit. on p. 18).

Jiapeng Wang and Yihong Dong (Aug. 2020). "Measurement of Text Similarity: A Survey". In: *Information* 11.9, p. 421. ISSN: 2078-2489. DOI: 10.3390/info11090421 (cit. on p. 27).

Lidan Wang, Ming Tan, and Jiawei Han (2016a). "FastHybrid: A hybrid model for efficient answer selection". In: *Proceedings of the 26th International Conference on Computational Linguistics*, pp. 2378–2388 (cit. on p. 84).

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, et al. (July 2020a). "CORD-19: The COVID-19 Open Research Dataset". In: *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. Online: Association for Computational Linguistics. URL: https://aclanthology.org/2020.nlpcovid19-acl.1 (cit. on pp. 134, 135).

Shirui Wang, Wenan Zhou, and Chao Jiang (Mar. 2020b). "A Survey of Word Embeddings Based on Deep Learning". In: *Computing* 102.3, pp. 717–740. ISSN: 0010-485X, 1436-5057. DOI: 10.1007/s00607-019-00768-7 (cit. on p. 30).

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma (Apr. 2021). "Entailment as Few-Shot Learner". In: *arXiv:2104.14690*. DOI: 10.48550/arXiv.2104.14690 (cit. on pp. 49, 174).

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma (June 2020c). "Linformer: Self-Attention with Linear Complexity". In: *arXiv:2006.04768*. DOI: 10.48550/arXiv.2006.04768 (cit. on p. 37).

Tongzhou Wang and Phillip Isola (July 2020). "Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere". In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. PMLR, pp. 9929–9939. URL: https://proceedings.mlr.press/v119/wang20k.html (cit. on pp. 100, 102, 108).

Xiaolong Wang, Yufei Ye, and Abhinav Gupta (2018). "Zero-Shot Recognition via Semantic Embeddings and Knowledge Graphs". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6857–6866. DOI: 10.48550/arXiv.1803.08035 (cit. on p. 101).

Yequan Wang, Minlie Huang, xiaoyan zhu, and Li Zhao (2016b). "Attention-Based LSTM for Aspect-level Sentiment Classification". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 606–615. DOI: 10.18653/v1/D16-1058 (cit. on pp. 44, 46).

Mark Ware and Michael Mabe (2015). "The STM Report: An overview of scientific and scholarly journal publishing". In: *The STM Report* 3. ISSN: 1098-6596. URL: https://digitalcommons.unl.edu/scholcom/9/ (cit. on p. 3).

Hannes Westermann, Jaromir Savelka, and Karim Benyekhlef (2021). "Paragraph Similarity Scoring and Fine-Tuned BERT for Legal Information Retrieval and Entailment". In: *New Frontiers in Artificial Intelligence*. Vol. 12758. Cham: Springer International Publishing, pp. 269–285. ISBN: 978-3-030-79942-7. DOI: 10.1007/978-3-030-79942-7_18 (cit. on pp. 17, 29).

Gineke Wiggers and Suzan Verberne (2019). "Citation Metrics for Legal Information Retrieval Systems". In: *BIR 2019 Workshop on Bibliometric-enhanced Information Retrieval ECIR*, pp. 39–50 (cit. on pp. 16, 94).

Wikisource (2020). *United States Supreme Court decisions by topic*. URL: https://en.wikisource.org/wiki/Category:United%5C%5FStates%5C%5FSupreme%5C%5FCourt%5C%5Fdecisions%5C%5Fby%5C%5Ftopic (visited on 06/30/2020) (cit. on p. 81).

Adina Williams, Nikita Nangia, and Samuel Bowman (2018). "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". In: *arXiv:1704.05426*, pp. 1112–1122. DOI: 10.18653/v1/n18-1101 (cit. on p. 83).

Radboud Winkels, Alexander Boer, Bart Vredebregt, and Alexander Van Someren (2014). "Towards a Legal Recommender System". In: *Frontiers in Artificial Intelligence and Applications*. Vol. 271, pp. 169–178. ISBN: 9781614994671 (cit. on pp. 16, 80).

Andi Winterboer and Johanna D. Moore (2007). "Evaluating information presentation strategies for spoken recommendations". In: *RecSys'07: Proceedings of the 2007 ACM Conference on Recommender Systems*, pp. 157–160. DOI: 10.1145/1297231.1297260 (cit. on pp. 141, 142).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. (Oct. 2020). "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6 (cit. on pp. 105, 124, 138).

Allison Woodruff, Rich Gossweiler, James Pitkow, Ed H. Chi, and Stuart K. Card (2000). "Enhancing a Digital Book with a Reading Recommender". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, pp. 153–160. DOI: 10.1145/332040.332419 (cit. on p. 40).

Dustin Wright and Isabelle Augenstein (Aug. 2021). "CiteWorth: Cite-Worthiness Detection for Improved Scientific Document Understanding". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pp. 1796–1807. DOI: 10.18653/v1/2021.findings-acl.157 (cit. on pp. 100, 105, 106).

Baoning Wu and Brian D Davison (2005). "Identifying Link Farm Spam Pages". In: *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, pp. 820–829 (cit. on p. 39).

Chao-yuan Wu, R. Manmatha, Alexander J Smola, and Philipp Krahenbuhl (Oct. 2017). "Sampling Matters in Deep Embedding Learning". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, pp. 2859–2867. DOI: 10.1109/ICCV.2017.309 (cit. on p. 102).

Felix Wu, Tianyi Zhang, Amauri Holanda de Souza, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger (2019). "Simplifying Graph Convolutional Networks". In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 6861-6871. PMLR, pp. 815–826. DOI: 10.48550/arXiv.1902.07153 (cit. on pp. 100, 105, 106).

Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui (May 2022). "Graph Neural Networks in Recommender Systems: A Survey". In: *ACM Computing Surveys*, p. 3535101. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3535101 (cit. on p. 43).

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu (2021). "Smoothed Contrastive Learning for Unsupervised Sentence Embedding". In: *arXiv:2109.04321*. DOI: 10.48550/arXiv.2109.04321 (cit. on p. 100).

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. (2016). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". In: *arXiv:1609.08144*. DOI: 10.48550/ARXIV.1609.08144 (cit. on p. 36).

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma (Dec. 2020). "CLEAR: Contrastive Learning for Sentence Representation". In: *arXiv:2012.15466*. DOI: 10.48550/arXiv.2012.15466 (cit. on pp. 97, 100).

Ellery Wulczyn and Dario Taraborelli (2015). "Wikipedia Clickstream". In: *figshare*. DOI: 10.6084/m9.figshare.1305770 (cit. on p. 57).

Ning Xi, Guangchao Tang, Xinyu Dai, Shujian Huang, and Jiajun Chen (July 2012). "Enhancing Statistical Machine Translation with Character Alignment". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Jeju Island, Korea: Association for Computational Linguistics, pp. 285–290. URL: https://aclanthology.org/P12-2056 (cit. on pp. 141, 142).

Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun (May 2021). *Lawformer: A Pre-trained Language Model for Chinese Legal Long Documents*. DOI: 10.48550/arXiv.2105.03887 (cit. on p. 17).

Xinyu Xing, Xiaosheng Fan, and Xiaojun Wan (2020). "Automatic Generation of Citation Texts in Scholarly Papers: A Pilot Study". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 6181–6190. DOI: 10.18653/v1/2020.acl-main.550 (cit. on p. 174).

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, Arnold Overwijk (2020). "Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval". In: *International Conference on Learning Representations*, pp. 1–16. DOI: 10.48550/arXiv.2007.00808 (cit. on p. 100).

Hu Xu, Bing Liu, Lei Shu, and Philip Yu (June 2019). "BERT Post-Training for Review Reading Comprehension and Aspect-Based Sentiment Analysis". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 2324–2335. DOI: 10.18653/v1/N19-1242 (cit. on pp. 44, 46).

Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang (Aug. 2021). "A Unified Generative Framework for Aspect-Based Sentiment Analysis". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 2416–2429. DOI: 10.18653/v1/2021.acl-long.188 (cit. on pp. 44, 46, 85).

Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang (2015). "Network Representation Learning with Rich Text Information". In: *Proceedings of the 24th International Conference on Artificial Intelligence*. Buenos Aires, Argentina: AAAI Press, pp. 2111–2117. DOI: 10.5555/2832415.2832542 (cit. on p. 100).

Chenxing Yang, Baogang Wei, Jiangqin Wu, Yin Zhang, and Liang Zhang (2009). "CARES: A Ranking-Oriented CADAL Recommender System". In: *Proceedings of the 2009 Joint International Conference on Digital Libraries - JCDL '09*. Austin, TX, USA: ACM Press, p. 203. DOI: 10.1145/1555400.1555432 (cit. on p. 18).

Xiwang Yang, Yang Guo, Yong Liu, and Harald Steck (Mar. 2014). "A Survey of Collaborative Filtering Based Social Recommender Systems". In: *Computer Communications* 41, pp. 1–10. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2013.06.009 (cit. on p. 18).

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le (2019). "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., pp. 5754–5764. DOI: 10.48550/arXiv.1906.08237 (cit. on pp. 36, 118, 122, 124, 130, 136, 138).

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun (2019). "DocRED: A Large-Scale Document-Level Relation Extraction Dataset". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 764–777. DOI: 10.18653/v1/P19-1074 (cit. on p. 119).

Michihiro Yasunaga, Jure Leskovec, and Percy Liang (2022). "LinkBERT: Pretraining Language Models with Document Links". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 8003–8016. DOI: 10.18653/v1/2022.acl-long.551 (cit. on p. 173).

Jun Yin and Xiaoming Li (2017). "Personalized Citation Recommendation via Convolutional Neural Networks". In: *Web and Big Data*. Vol. 10367. Cham: Springer International Publishing, pp. 285–293. ISBN: 978-3-319-63564-4. DOI: 10.1007/978-3-319-63564-4_23 (cit. on p. 15).

Cong Yu, Laks Lakshmanan, and Sihem Amer-Yahia (2009). "It Takes Variety to Make a World: Diversification in Recommender Systems". In: *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. Saint Petersburg, Russia: Association for Computing Machinery, pp. 368–378. DOI: 10.1145/1516360.1516404 (cit. on p. 54).

Chao Zhang, Nan Sun, Xia Hu, Tingzhu Huang, and Tat Seng Chua (2009). "Query segmentation based on eigenspace similarity". In: *ACL-IJCNLP 2009 - Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and 4th International Joint Conference on Natural Language Processing of the AFNLP, Proceedings of the Conference*, pp. 185–188. DOI: 10.3115/1667583.1667640 (cit. on p. 142).

Dong Zhang, Shu Zhao, Zhen Duan, Jie Chen, Yang-ping Zhang, and Jie Tang (2020a). "A Multi-Label Classification Method Using a Hierarchical and Transparent Representation for Paper-Reviewer Recommendation". In: *ACM Transactions on Information Systems (TOIS)* 38, pp. 1–20 (cit. on pp. 44, 45).

Shuai Zhang, Xi Rao, Yi Tay, and Ce Zhang (2021a). "Knowledge Router: Learning Disentangled Representations for Knowledge Graphs". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 1–10. DOI: 10.18653/v1/2021.naacl-main.1 (cit. on p. 49).

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay (Jan. 2020b). "Deep Learning Based Recommender System: A Survey and New Perspectives". In: *ACM Computing Surveys* 52.1, pp. 1–38. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3285029 (cit. on p. 18).

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam (Nov. 2021b). "Aspect Sentiment Quad Prediction as Paraphrase Generation". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 9209–9219. DOI: 10.18653/v1/2021.emnlp-main.726 (cit. on pp. 44, 46).

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam (Mar. 2022). "A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges". In: *arXiv:2203.01054*. DOI: 10.48550/arXiv.2203.01054 (cit. on pp. 44, 46).

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam (Aug. 2021c). "Towards Generative Aspect-Based Sentiment Analysis". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Online: Association for Computational Linguistics, pp. 504–510. DOI: 10.18653/v1/2021.acl-short.64 (cit. on pp. 44, 46).

Xiongyi Zhang, Jan-Willem van de Meent, and Byron Wallace (2021d). "Disentangling Representations of Text by Masking Transformers". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 778–791. DOI: 10.18653/v1/2021.emnlp-main.60 (cit. on pp. 44, 48).

Yi Zhang, Fen Zhao, and Jianguo Lu (Oct. 2019). "P2V: Large-Scale Academic Paper Embedding". In: *Scientometrics* 121.1, pp. 399–432. ISSN: 0138-9130, 1588-2861. DOI: 10.1007/s11192-019-03206-9 (cit. on pp. 14, 20).

Yongfeng Zhang and Xu Chen (2020). "Explainable Recommendation: A Survey and New Perspectives". In: *Foundations and Trends in Information Retrieval* 14.1, pp. 1–101. ISSN: 1554-0669. DOI: 10.1561/1500000066 (cit. on pp. 6, 77, 174).

Yue Zhang and Stephen Clark (2009). "Transition-based parsing of the Chinese treebank using a global discriminative model". In: *Proceedings of the 11th International Conference on Parsing Technologies - IWPT '09*. Morristown, NJ, USA: Association for Computational Linguistics, p. 162. DOI: 10.3115/1697236.1697267 (cit. on pp. 141, 142).

Qian Zhao, F. Maxwell Harper, Gediminas Adomavicius, and Joseph A. Konstan (2018). "Explicit or Implicit Feedback? Engagement or Satisfaction? A Field Experiment on Machine-Learning-Based Recommender Systems". In: *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. Pau, France: Association for Computing Machinery, pp. 1331–1340. DOI: 10.1145/3167132.3167275 (cit. on p. 54).

Zicheng Zhao, Hui Ning, Liang Liu, Chengzhe Huang, Leilei Kong, Yong Han, Zhongyuan Han (2019). "FIRE2019AILA: Legal Information Retrieval Using Improved BM25". In: *FIRE2019AILA*, p. 6 (cit. on p. 16).

Hao Zheng and Mirella Lapata (July 2019). "Sentence Centrality Revisited for Unsupervised Summarization". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 6236–6247. DOI: 10.18653/v1/P19-1628 (cit. on p. 47).

Hua Zheng, Dong Wang, Qi Zhang, Hang Li, and Tinghao Yang (2010). "Do Clicks Measure Recommendation Relevancy?" In: *Proceedings of the Fourth ACM Conference on Recommender Systems - RecSys '10*. New York, New York, USA: ACM Press, p. 249. DOI: 10.1145/1864708.1864759 (cit. on pp. 13, 19).

Ding Zhou, Shenghuo Zhu, Kai Yu, Xiaodan Song, Belle L. Tseng, Hongyuan Zha, C. Lee Giles (2008). "Learning Multiple Graphs for Document Recommendations". In: *Proceeding of the 17th International Conference on World Wide Web - WWW '08*. Beijing, China: ACM Press, p. 141. DOI: 10.1145/1367497.1367517 (cit. on pp. 14, 20).

Xinyue Zhu, Yifan Liu, Zengchang Qin, and Jiahong Li (2017). "Data Augmentation in Emotion Classification Using Generative Adversarial Networks". In: *arXiv:1711.00648*. DOI: 10.48550/arXiv.1711.00648 (cit. on pp. 157, 158).

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, Sanja Fidler (2015). "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books". In: *Proceedings of the IEEE International Conference on Computer Vision*. Vol. 2015 Inter, pp. 19–27. DOI: 10.1109/ICCV.2015.11 (cit. on pp. 37, 124, 131, 137, 138).

Glossary

A

ACL — Annual Meeting of the Association for Computational Linguistics
The ACL conference is one of the primary high impact conferences for natural language processing research (has a CORE-2021 rank of `A*`).

AILA — Shared task on Artificial Intelligence for Legal Assistance

Apache Lucene:
Apache Lucene is a free and open-source full-text search library written in Java.

API — Application Programming Interface
Computing interface for software-to-software communication; specifies the possible interactions, their workflow, and the data exchanged.

ArXiv:
ArXiv is an online repository of scientific publications and preprints that are freely available to the public.

Aspect:
The aspect defines the perspective from that a user looks at the document's content when assessing the document's similarity.

AVG — Average

B

B — Billion

BERT — Bidirectional Encoder Representations from Transformers
BERT is a encoder-only Transformer language model introduced by Devlin et al. (2019).

C

CAP — Caselaw Access Project

CBOW — Continuous Bag-of-Words

CF — Collaborative Filtering
Collaborative filtering is a method of generating recommendations or predictions about a user's interests based on the interests of other users. It involves finding similar users based on their past ratings or behavior, and using those users' preferences to make recommendations to the target user.

CNN — Convolutional Neural Network
A convolutional neural network is a type of artificial neural network specifically designed to learn spatial hierarchies of features from input data with the help of convolutional filters.

CoCit — Co-Citation
A co-citation occurs when two or more documents are being cited together in a third document. Co-citations are used as a measure of the relationship between different documents.

COLIEE — Competition on Legal Information Extraction and Entailment

COLING — International Conference on Computational Linguistics
COLING is one of the premier conferences for the natural language processing and computational linguistics (has a CORE-2021 rank of `A`).

Content-based method:
A method is considered content-based if it primarily relies on information extracted from the item it is applied to. In the case of documents, this information can include the textual content, such as the title and main body text, as well as graph information like citations or web links.

CORE — Computing Research and Education Association of Australasia
CORE is an association of university departments that provide assessments of major conferences in the computing disciplines. The main categories are `A*` (flagship), `A` (excellent), `B` good to very good, and `C` for other ranked conferences that meet minimum standards, see `http://portal.core.edu.au/conf-ranks/`.

Corpus:
A corpus is a collection of texts in machine-readable form.

CPA — Co-Citation Proximity Analysis
CPA is a similarity measures introduced by Gipp and Beel (2009) that uses co-citation and the position of citation markers to assess the similarity of documents.

CPI — Co-Citation Proximity Index
CPI is the central metric of CPA that quantifies the proximity of co-cited documents.

CPU — Central processing unit

CSV — Comma-separated values

CTR — Click-through-rate

CUP — Cambridge University Press
Cambridge University Press is a department of the University of Cambridge and is both an academic and educational publisher.

D

DBOW — Distributed Bag-of-Words

Doc2Vec (Paragraph Vectors):
A document embedding technique introduced by Le and Mikolov (2014).

DocEng — ACM Symposium on Document Engineering
The ACM Symposium on Document Engineering is an annual meeting of researchers active in document engineering (has a CORE-2021 rank of B).

DOI — Digital Object Identifier
Persistent identifier for digital data maintained and resolved by a registrar; frequently assigned to research publications.

E

Embedding:
An embedding is a way of representing discrete variables as low-dimensional, continuous vectors in the context of machine learning.

EMNLP — The Conference on Empirical Methods in Natural Language Processing
EMNLP is a leading conference in the area of natural language processing and artificial intelligence and organized by the Association for Computational Linguistics (has a CORE-2021 rank of A).

ER — Explicit Retrofitting
A word vector retrofitting method introduced by Glavaš and Vulić (2018).

ES — Elasticsearch
Elasticsearch is a distributed search engine based on the Apache Lucene framework.

F

F1-Score:
Harmonic mean of precision and recall

G

GenSim:
GenSim is an open-source Python library used for topic modeling, word vectors, and other NLP tasks (Rehurek and Sojka, 2010).

GPT — Generative Pretrained Transformer

GPU — Graphics processing unit

I

ICADL — International Conference on Asia-Pacific Digital Libraries
ICADL is a digital library conference and held annually for connecting digital library, computer science, and library and information science communities (has a CORE-2018 rank of A).

ICAIL — International Conference on Artificial Intelligence and Law
ICAIL is the primary international conference addressing research in Artificial Intelligence and Law, and has been organized biennially since 1987 under the auspices of the International Association for Artificial Intelligence and Law (has a CORE-2021 rank of C).

Information need:
An information need is the topic about which the user desires to know more.

IR — Information Retrieval
Information retrieval is the process of searching for and retrieving information from a collection of documents or data. It involves designing and implementing systems that can locate and extract relevant information from a large corpus based on a user's query.

J

JCDL — ACM/IEEE Joint Conference on Digital Libraries
JCDL is an annual international conference focusing on digital libraries and associated technical, practical, and social issues (has a CORE-2018 rank of A*).

JSON — JavaScript Object Notation
A a lightweight data interchange format based on the JavaScript programming language that is easy for humans to read and write and easy for machines to parse and generate.

K

k — Kilo (Thousand)

kNN — *k* Nearest Neighbors

KONVENS — Konferenz zur Verarbeitung natürlicher Sprache
KONVENS is an annually held conference covering diverse topics from computer linguistics and language technologies.

L

LDA — Latent Dirichlet Allocation
LDA is a statistical topic model introduced by Blei et al. (2003).

LM — Language Model
A language model is a type of artificial intelligence model that is trained on a large corpus of text data and is used to predict or generate new sequences of text.

LREC — International Conference on Language Resources and Evaluation
LREC is the major event on language resources and evaluation for language technologies (has a CORE-2021 rank of C).

LSTM — Long Short-Term Memory
An LSTM is a type of RNN that is specifically designed to model long-term dependencies in sequential data, introduced by Hochreiter and Schmidhuber (1997).

M

M — Million

MAG — Microsoft Academic Graph

MAP — Mean Average Precision
A performance measure representing the mean of the average precision scores over a set of queries.

MeSH — Medical Subject Headings

MLM — Masked Language Modeling
MLM is a pretraining objective in which the model is trained to predict masked (hidden) tokens in a given sequence.

MLP — Multilayer Perceptron
MLP is a type of artificial neural network that has at least two layers of nodes: an input layer and an output layer, with one or more hidden layers in between.

MLT — MoreLikeThis
MLT is a function of the Apache Lucence search framework that allows the retrieval of semantically similar documents, see `https://lucene.apache.org/core/7_2_0/queries/org/apache/lucene/queries/mlt/MoreLikeThis.html`.

MRR — Mean Reciprocal Rank
A performance measure representing the average of the reciprocal ranks at which the method retrieves the first relevant item for each query.

N

nDCG — Normalized Discounted Cumulative Gain
A performance measure computed as the sum the true scores ranked in the order induced by the predicted scores, after applying a logarithmic discount.

NLP — Natural Language Processing
Natural language processing is a field of artificial intelligence, computer science, and linguistics that involves developing algorithms and models that can understand, interpret, and generate human language, including speech and text. Applications of NLP include language translation, text and speech recognition, question answering, and text summarization, among others.

NSP — Next Sentence Prediction
: NSP is a pretraining objective used by BERT and other language models.

O

Offline evaluation:
: Offline evaluation is an evaluation of a system based on historical data.

Online evaluation:
: Online evaluation is an evaluation of a system based on measurement of real users' experiences of the system in a natural usage environment.

OOV — Out-of-vocabulary

P

P — Precision
: A performance measure that is defined as the fraction of relevant items among the retrieved items.

PDF — Portable Document Format

PV-DBOW — Distributed Bag-of-Words of Paragraph Vector

PV-DM — Distributed Memory Model of Paragraph Vectors

PyTorch:
: PyTorch is an open-source machine learning library for Python.

Q

QURATOR — Conference on Digital Curation Technologies
: The Qurator conference provides a forum on the use of digital curation technologies in application domains for, e.g., media, journalism, logistics, cultural heritage, health care and life sciences, energy, industry.

R

R — Recall
: A performance measure that is defined as the fraction of relevant items that were retrieved.

RAM — Random access memory

RDF — Resource Description Framework

Recommender System:
A recommender system is an application that recommends the most suitable item to a particular user given a collection of items.

RecSys — ACM Conference on Recommender Systems
The ACM Conference on Recommender Systems is the major international conference for new research results, systems and techniques in the broad field of recommender systems (has a CORE-2021 rank of A).

RNN — Recurrent Neural Network
A recurrent neural network is a type of artificial neural network that is designed to process sequential data by maintaining a state that depends on the past elements of the sequence.

S

S2ORC — Semantic Scholar Open Research Corpus

SciDocs — Scientific Document Representation Benchmark

SPARQL:
SPARQL (SPARQL Protocol and RDF Query Language) is a query language for accessing and manipulating data stored in the Resource Description Framework (RDF) format. RDF is a standardized data model for representing information as a set of interconnected triples, where each triple consists of a subject, predicate, and object.

T

T — Trillion

TF-IDF — Term Frequency - Inverse Document Frequency
TF-IDF evaluates how relevant or important a term is to a document in a collection of documents (Salton, 1971).

Transformer:
A Transformer is a type of deep learning model architecture introduced by (Vaswani et al., 2017).

U

UI — User Interface
A user interface is the point of interaction between a human user and a computer or a software application.

UMAP — Uniform Manifold Approximation and Projection

URL — Uniform Resource Locator

USE — Universal Sentence Encoder

User-based method:
A user-based method is one that primarily relies on user information, such as user profiles or interactions. Collaborative filtering is an example of a user-based method.

V

VSM — Vector Space Model
A representation of documents as numeric vectors by using raw or weighted term counts as the vector elements.

W

Wikidata:
Wikidata is a free and open knowledge graph that is used to support Wikipedia and other Wikimedia projects.

Wikipedia:
Wikipedia is a free online encyclopedia that is collaboratively written and consists of articles on a wide range of topics in many languages.

Word2Vec:
A word embedding technique introduced by Mikolov et al. (2013b).

X

XML — Extensible Markup Language
A standard for encoding documents in a format that is readable for machines and humans.