

© 2021 Saar Kuzi

ACCELERATING SCIENTIFIC RESEARCH IN THE DIGITAL ERA: INTELLIGENT
ASSESSMENT AND RETRIEVAL OF RESEARCH CONTENT

BY

SAAR KUZU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Professor ChengXiang Zhai, Chair
Professor Kevin Chenchuan Chang
Professor Heng Ji
Dr. Michael Bendersky, Google Research

Abstract

The efficient, effective, and timely access to the scientific literature by researchers is crucial for accelerating scientific research and discovery. Nowadays, research articles are almost exclusively published in a digital form and stored in digital libraries, accessible over the Web. Using digital libraries for storing scientific literature is advantageous as it enables access to articles at any time and place. Furthermore, digital libraries can leverage information management systems and artificial intelligence techniques to manage, retrieve, and analyze research content. Due to the large size of those libraries and their fast growth pace, the development of intelligent systems that can effectively retrieve and analyze research content is crucial for improving the productivity of researchers. In this thesis, we focus on improving literature search engines by addressing some of their limitations.

One of the limitations of the current literature search engines is that they mainly treat articles as the retrieval units and do not support the direct search for any of the article's elements such as figures, tables, and formulas. In this thesis, we study how to enable researchers to access research collections using figures of articles. Figures are entities in research articles that play an essential role in scientific communications. For this reason, research figures can be utilized directly by literature systems to facilitate and accelerate research. As the first step in this direction, we propose and study the novel task of figure retrieval from collections of research articles where the goal is to retrieve research article figures using keyword queries. We focus on the textual bag-of-words representation of search queries and figures and study the effectiveness of different retrieval models for the task and various ways to represent figures using text data. The empirical study shows the benefit of using multiple textual inputs for representing a figure and combining different retrieval models. The results also shed light on the different challenges in addressing this novel task.

Next, we address the limitations of the text-based bag-of-words representation of research figures by proposing and studying a new view of representation, namely deep neural network-based distributed representations. Specifically, we focus on using image data and text for learning figure representations with different model architectures and loss functions to understand how sensitive the embeddings are to the learning approach and the features used. We also develop a novel weak supervision technique for training neural networks for this task that leverages the citation network of articles to generate large quantities of training examples. The experimental results show that figure representations, learned using our weak supervision approach, are effective and outperform representations of the bag-of-words technique and pre-trained neural networks.

The current systems also have minimal support for addressing queries for which a search engine performs poorly due to ineffective formulation by the user. When conducting research, poor-performing search queries may occur when a researcher faces a new or fast-evolving research topic, resulting in a significant vocabulary gap between the user’s query and the relevant articles. In this thesis, we address this problem by developing a novel strategy for collaborative query construction. According to this strategy, the search engine would actively engage users in an iterative process to continuously revise a query. We propose a specific implementation of this strategy in which the search engine and the user work together to expand a search query. Specifically, the system generates expansion terms, utilizing the history of interactions of the user with it, that the user can add to the search query in every iteration to reach an “ideal query”. The experimental results attest to the effectiveness of using this approach in improving poor-performing search queries with minimal effort from the user.

The last limitation that we address in this thesis is that the current systems usually do not leverage any content analysis for the quality assessment of articles and instead rely on citation counts. In this thesis, we study the task of automatic quality assessment of research articles where the goal is to assess the quality of an article in different aspects such as clarity, originality, and soundness. Automating the quality assessment of articles could improve the current literature systems that can leverage the generated quality scores to support the search and analysis of research articles. Previous works have applied supervised machine learning to automate the assessment by learning from examples of reviewed articles by humans. In this thesis, we study the effectiveness of using topics for the task and propose a novel strategy for constructing multi-view topical features. Experimental results show that such features are effective for this task compared to deep neural network-based features and bag-of-words features.

Finally, to facilitate further evaluation of the different approaches suggested in this thesis using real users and realistic user tasks, we developed **AcademicExplorer**, a novel general system that supports the retrieval and exploration of research articles using several new functions enabled by the proposed algorithms in this thesis, such as exploring research collections using figure embeddings, sorting research articles based on automatically generated review scores, and interactive query formulation. As an open-source system, **AcademicExplorer** can help advance the research, evaluation, and development of applications in this area.

To my parents, for their love and support.

Acknowledgments

First and foremost, I would like to thank my advisor, Professor ChengXiang Zhai, for his guidance and support during my Ph.D. journey. Cheng would always encourage me to develop a research plan based on my interests which helped shape me as an independent researcher. With his optimistic approach to research, I learned to focus on the positive in times of failure and disappointment, which was crucial for completing this thesis. Research discussions and brainstorming with Cheng were always fruitful, inspired me to explore exciting research directions, and I am very grateful to him for that.

I would also like to thank my committee members. Dr. Michael Bendersky provided many practical ideas for improving the approaches studied in this thesis. I am also grateful to him for the internship opportunity at Google that exposed me to research in the industry setting. The feedback from Professor Kevin Chenchuan Chang helped me improve the coherency and clarity of my thesis work, and I am very grateful to him for that. Finally, I want to thank Professor Heng Ji for her support. Heng shared many ideas and proposed collaboration opportunities to increase the impact of this thesis.

I want to thank my collaborators in different parts of this thesis. Professor William Cope, Professor Duncan Ferguson, and Dr. Chase Geigle deserve special thanks for the fruitful collaboration on the automatic assessment project. Thanks to Abhishek Narwekar and Anusri Pampari for collaborating on the interactive query formulation project. Last, I would like to thank IBM, The China Railway Rolling Stock Corporation Academy, Yin Tian, Haichuan Tang, and Dr. Jinjun Xiong for their support in developing the AcademicExplorer system.

I want to thank the Computer Science Department at the University of Illinois Urbana-Champaign for allowing me to pursue this Ph.D. degree and for their support. The members of the Text Information Management and Analysis group also deserve special thanks. From informal talks in the office to formal meetings and collaborations, they shared helpful feedback that helped me improve my work.

Finally, I want to thank my family for their support throughout my academic path so far and for always believing in me.

TABLE OF CONTENTS

CHAPTER 1	INTRODUCTION	1
1.1	The Limitations of the Current Literature Search Systems	1
1.2	The Main Contributions of this Thesis	2
CHAPTER 2	RELATED WORK	6
CHAPTER 3	FIGURE RETRIEVAL FROM COLLECTIONS OF RESEARCH ARTICLES	10
3.1	Introduction	10
3.2	Related Work	11
3.3	Figure Retrieval	13
3.4	Evaluation	16
3.5	Conclusions	22
CHAPTER 4	A STUDY OF DISTRIBUTED REPRESENTATIONS FOR FIGURES OF RESEARCH ARTICLES	25
4.1	Introduction	25
4.2	Related Work	26
4.3	Figure Embeddings	27
4.4	Weak Supervision for Figure Embeddings	31
4.5	Empirical Study	32
4.6	Conclusions	39
CHAPTER 5	INTERACTIVE SUPPORT FOR QUERY CONSTRUCTION IN LITERATURE SEARCH ENGINES	40
5.1	Introduction	40
5.2	Collaborative Query Construction (CQC)	42
5.3	Related Work	46
5.4	Evaluation	48
5.5	Conclusions	58
CHAPTER 6	AUTOMATIC ASSESSMENT OF RESEARCH ARTICLES	59
6.1	Introduction	59
6.2	Related Work	62
6.3	Topic Discovery and Construction of Topical Features	65
6.4	Experimental Setup	68
6.5	Experimental Results	73
6.6	Result List Ranking Using Aspect Scores	87
6.7	Generating Explanations for Automatically Generated Quality Scores	94
6.8	Conclusions	95

CHAPTER 7	ACADEMIC-EXPLORER: A SYSTEM FOR RETRIEVAL AND EXPLORATION OF RESEARCH ARTICLE COLLECTIONS	97
7.1	Introduction	97
7.2	Related Systems	99
7.3	System Functions	100
7.4	Implementation	103
7.5	Data Sets	106
7.6	Application Scenarios	107
7.7	Conclusions	108
CHAPTER 8	CONCLUSIONS	109
8.1	Summary of Contributions	109
8.2	Deployment of Academic-Explorer	110
8.3	Future Work	112
REFERENCES	115

CHAPTER 1: INTRODUCTION

Scientific research is essential for increasing our knowledge about the world and improving many aspects of our lives. Conducting scientific research is usually a labor-intensive and complicated process. For this reason, researchers often use computer systems to support and facilitate different parts of it. The advances in artificial intelligence in the last years have opened up many opportunities for developing a new generation of research assistant systems to accelerate research and scientific discovery. In this thesis, the focus is on advancing the technology of intelligent data-driven systems to assist researchers and improve their research productivity.

An integral component of the research process, which is the focus of this thesis, is the consumption of knowledge from the existing scientific literature. For example, researchers often need to conduct extensive literature surveys to discover missing knowledge in a scientific field. Scientific literature is usually stored nowadays in a digital format using digital libraries, which are easily accessible over the Web. The number of scientific publications in the last years is growing at an exponential pace [1]. As a result, the collections of scientific literature contain a large number of articles and also evolve at a very high rate.

The main tools used by researchers nowadays to obtain knowledge from collections of research articles are literature search engines, such as Google Scholar [2], Microsoft Academic [3], and Semantic Scholar [4]. The main task performed in these tools is the ad-hoc retrieval task in which users input keyword queries to retrieve relevant articles. The main advantage of these tools is that they are general enough to support a great variety of research tasks, such as literature review, question answering, and known-item search. Yet, the current systems have several limitations, which we review in the following section.

1.1 THE LIMITATIONS OF THE CURRENT LITERATURE SEARCH SYSTEMS

The existing literature search engines have different limitations. In this thesis, we address three main drawbacks of current systems, which we overview in detail in this section.

First, the retrieval units in existing literature search engines are mostly research articles. Using only research articles as retrieval units ignores the different elements of a research article that contain different types of focused information that can be valuable for researchers. For example, research articles usually contain elements, such as tables, figures, formulas, algorithms, and data sets that researchers often pay special attention to when reading a paper. To obtain such information using the current systems, users need to search for

articles and then locate those elements within them.

Another limitation of the current systems is that they mainly support the single and standard mode of interaction prevalent in search engines. According to this interaction paradigm, users input individual keyword queries to obtain result lists. Thus, to complete a research task using a literature search engine, users often need to issue several queries. This can be the case, for example, when researchers study new problems that they are not very familiar with, which may result in poorly performing queries due to a vocabulary gap. In other cases, the user's information need is too complex to be satisfied by a single query. Finally, the information need in some tasks may not be well defined at the beginning of the process (e.g., in the case of a literature review or exploratory search). When using a standard search engine to complete such tasks, each query is usually considered as an independent unit from the system perspective. For this reason, users would often end up formulating many queries to complete the task with minimal support from the system [5, 6].

Finally, to obtain relevant and high-quality articles in the result list, literature search engines need to automatically assess the quality of research articles. This problem is similar to the case of Web search where different algorithms, such as PageRank, are used to determine the quality of a Web page. The current literature search engines mainly use the number of citations of an article to this end and do not leverage almost any content analysis techniques. Using only the paper citations has some limitations. For example, citation information is not very useful for recently published works and emerging topics. Furthermore, the quality of a paper has different aspects, such as clarity, originality, and novelty, which do not always fully correlate with the number of citations.

1.2 THE MAIN CONTRIBUTIONS OF THIS THESIS

In this thesis, we address the limitations of existing literature search systems in three ways: (1) we study how to enable researchers to explore research article collections using figures, (2) we propose a novel approach for query construction that optimizes the collaboration between the user and the system, and (3) we study how to improve the automated assessment of research articles using textual features. Finally, to illustrate the different approaches proposed in this thesis, we developed a novel system for exploring collections of research articles. In the remainder of this chapter, we review the main contributions of this thesis.

1.2.1 Using Research Figures to Explore Collections of Research Articles

Figures of research articles are important elements that researchers often pay special

attention to. For example, research figures are often used in a research article to depict its main technical contributions and empirical findings. For this reason, many application systems can directly use research article figures to assist researchers.

In this thesis, we first introduce and study a new task of *figure retrieval* in which the retrieval units are figures of research articles, and the goal is to rank research figures given a query [7]. As a first step toward addressing this task, we focus on textual queries and represent a figure using text extracted from its article. Then, using this type of representation, we study the effectiveness of several retrieval methods for the task. The results of this study help to gain a preliminary understanding of the relative effectiveness of different representations of a figure and retrieval methods. The results also shed light on the possible types of information need that can be satisfied by this task and the potential challenges in figure retrieval.

Secondly, we study the effectiveness of *distributed representations*, learned using deep neural networks, for research figures [8]. We learn representations using both text and image data and compare different model architectures and loss functions for the task. Furthermore, to overcome the lack of training data for the task, we propose and study a novel weak supervision approach for learning embedding vectors and show that it is more effective than using some of the pre-trained neural models as suggested by recent works. Experimental results show that distributed representations for research figures can be more effective than the previously studied bag-of-words representations. Yet, combining the two approaches can further improve performance. Finally, the results also show that these representations, while effective in general, can be sensitive to the learning approach used and that using both image data and text and a simple model architecture is the most effective approach.

1.2.2 Interactive Support for Query Construction

A second research direction of this thesis is improving the accuracy of poor-performing queries in literature search engines whose result list does not contain much relevant information. An example of a scenario in which this can happen is when researchers study a research topic that is new to them or an emerging and fast-evolving research topic in general. In such a case, there might be a large vocabulary gap between the user’s query and the relevant documents, resulting in poor-performing search queries.

To address this problem, we propose a novel strategy of collaborative query construction where the search engine would actively engage users in an iterative process to revise a search query [9]. This approach can be implemented in any search engine to provide search support for users via a “Help Me Search” button that users can click on as needed. We focus on

studying a specific collaboration strategy where the search engine and the user work together to expand a query iteratively. We propose a possible implementation for this strategy in which the system generates candidate terms by utilizing the history of interactions of the user with the system. Our evaluation, using a simulated user and a data set of research articles, shows the great promise of the proposed approach. Specifically, the results show that by using this approach, a substantial number of queries, which initially resulted in no relevant articles, can be improved with minimal effort of the user.

1.2.3 Automatic Assessment of Research Articles

The automated assessment of research articles has many advantages. First, it has the potential of improving the current literature search system by including quality signals rather than just citation counts. Second, it can help to scale up the review process by assisting reviewers, thus minimizing the effort required for the reviewing process.

To address this task, some previous works have applied supervised machine learning to automate the assessment by learning from examples of articles that were reviewed and scored in different aspects [10]. In the previous work, textual features that were learned using deep neural networks have mainly been used for the task. In this thesis, we propose to use *topic model-based features* for this task [11]. Using topics is advantageous since the topics can be learned in an unsupervised manner, e.g., using a probabilistic topic model. For this reason, topic models do not require massive amounts of training data that is hard to obtain in this domain. Second, topic model-based features are more interpretable than the previously studied features. We propose and study multiple approaches to construct topical features and to combine topical features with bag-of-words features.

Experiments were performed using two data sets of research articles in the domains of computer science and veterinary medicine. The experimental results show that topical features are generally very effective and can substantially outperform the baseline features. However, their effectiveness is highly sensitive to how the topics are constructed and a combination of topics constructed using multiple views of the text data works the best. Finally, we also conducted an empirical analysis that demonstrated how the predicted scores can be used to improve literature search engines.

1.2.4 A Novel System for Exploring Research Article Collections

We developed `AcademicExplorer` to integrate the different approaches studied in this thesis [12]. `AcademicExplorer` is a novel general system that supports the retrieval and

exploration of research articles. Specifically, the system can support (1) figure and article retrieval using keyword queries, (2) various functions that can be used to explore the research collection using figure embeddings, (3) interactive query construction support, and (4) result list ranking using review aspect scores.

We designed this system to facilitate the collection of user data for training and test purposes. Furthermore, the system is flexible enough to be extended to include new functions and algorithms. As an open-source system, **AcademicExplorer** can help advance the research, evaluation, and development of applications in this area.

1.2.5 Summary

To conclude, this thesis tackles three limitations of the current literature search engines. Specifically, we study the exploration of research article collections using figures, interactive support for query construction, and automated assessment of research articles. Finally, to illustrate the contributions of this work, we developed a novel literature search system that implements those different ideas.

The rest of this dissertation is structured as follows. In Chapter 2, we provide a high-level literature review of works that are related to all contributions; we give an in-depth literature review for each of the thesis topics in the relevant chapters. Chapters 3 and 4 focus on our approach to figure retrieval and embedding, respectively. Next, we discuss our approach for interactive query construction in Chapter 5 and for the automated assessment of articles in Chapter 6. The implementation details of our novel literature search system are provided in Chapter 7. Finally, we conclude this work and discuss directions for future work in Chapter 8.

CHAPTER 2: RELATED WORK

In this chapter, we give a high-level review of the related work on approaches and systems for the search and mining of scientific literature. A more extensive literature review of the areas of the different thesis contributions is provided in the relevant chapters later.

We begin by surveying the existing commercial research literature systems and some of the main problems that were addressed in this domain in previous work. We then give an overview of the work done in the lines of the three main contributions of this thesis. First, we review the past work on research article representation and analysis, which relates to our study of research figures. Then, we review the previously studied interactive approaches to literature search, which relate to our collaborative query construction approach. Finally, we provide some related work in the area of automated assessment of research articles.

Commercial systems for literature search and mining Research articles are stored nowadays in digital libraries, which contain articles on a specific topic, such as computer science (e.g., the ACM Digital Library [13]) and the bio-medical domain (e.g., PubMed [14]). While these libraries have search engines to facilitate the retrieval of relevant information, the most common practice taken by researchers is to use general literature search systems that crawl the information from the different digital libraries over the Web [15]. One major search engine for research literature is Google Scholar [2] which mainly supports the search of articles using keyword queries. Another popular system is Microsoft Academic [3] in which users can also perform a semantic search using a knowledge graph [16]. Other literature search engines also provide AI-powered tools for further analysis of the research literature collection. One example for such a system is AMiner [17], which uses a topic modeling approach and social network analysis to find experts in different domains and for collaboration recommendation. In another system, Semantic Scholar [4], information extraction techniques are used to present the figures, tables, and research topics of a paper, which facilitate the consumption of research articles by researchers.

Other literature systems, not necessarily search engines, were also developed to assist researchers. For example, ResearchGate [18] is a social network platform in which researchers can share their papers and discuss different topics through posts and question answering. Other examples of literature systems are Google Dataset Search [19] for searching data sets for research over the Web, and Mendeley [20] for bibliographic/citation management.

All of those systems have the limitations that we address in this thesis. Specifically, in all of them, figure search is not supported, content-based quality assessment of articles is

limited, and there is minimal support for the construction of difficult search queries.

Research literature applications While most of the research literature systems focus mainly on search, there has been a large body of work on developing approaches to address other application scenarios. For example, the problem of citation recommendation was extensively studied in previous work [21, 22, 23, 24, 25, 26]. In this task, the goal is to generate a set of articles to be cited when writing a new paper. In a related research direction, some works have studied the problem of automatic generation of an entire “related work” section using an article as an input [27, 28, 29]. Other works have studied techniques to generate research summaries in a topic using a set of articles [30, 31, 32]. Finally, there has been some work on the task of research article recommendation where the goal is to recommend articles of interest to a researcher [33, 34, 35].

In this thesis, while the main focus is on search, the approaches we developed have the potential of benefiting a wider range of applications. For example, research figures can be used to generate a visual summary for a research topic to help researchers understand it better. Another example is citation recommendation where user-system collaboration can be leveraged to improve the outcome and speed of the process.

Research article representation and analysis Our work on figures of research articles is related to the large body of work on article representation and analysis. The general problem of text representation was studied in previous work and is still an active research direction [36]. Scientific research articles can be considered as a special category of documents with unique characteristics which led to a research direction of studying their representation in particular. Developing effective representations for scientific articles is an important direction to study since it can benefit virtually all research literature applications. In one line of work, the goal was to extract scientific concepts from articles that describe, for example, methods, algorithms, and processes [37, 38]. Similarly, other works focused on the annotation of sentences in articles for better visualization and indexing [39, 40]. The extraction of entities from articles and the relationships between them is a subject that was studied extensively in the past [16, 41, 42, 43, 44, 45]. The goal of the different works on this topic is to construct research knowledge graphs that can be used to improve the performance of different tasks. For example, using a knowledge graph was shown to improve the performance of article retrieval [46, 47] and assessment [48].

In a different line of work, the focus was on representing research articles using research elements, such as data sets [49, 50] and figures [51, 52]. The motivation behind these works is that using these elements can help to address unique information needs that are otherwise

hard to satisfy using a general-purpose search engine. Finally, semantic representations using dense vectors were also shown to be useful for article representation. For example, probabilistic topic models [53] and neural network-based language models (e.g., Sci-BERT [54]) were shown to be effective for article representation.

In this thesis, we study the problem of representing research articles using figures. Using figures provides an additional view of a research article that is complementary to the various types of representation that were previously studied.

Interactive approaches for literature search Researchers often use literature systems to complete complex tasks. For example, researchers often perform a literature review of a new topic, which requires an exploratory search process with multiple queries. For this reason, novel modes of effective user interaction and the optimization of user-system collaboration can potentially improve the performance of many downstream research applications. Despite the importance of optimizing user-system interaction, the main mode of interaction to date is of a standard search engine where users issue individual search queries to satisfy an information need (e.g., as in Google Scholar [2]).

While there have been many works on the topic of interactive information retrieval in general [55, 56, 57], the focus was rarely on literature search systems specifically. One work proposed to use reinforcement learning to balance the exploration/exploitation trade-off through the process of exploratory search [58]. In another work, a novel browsing tool was developed to assist researchers in answering complex questions by using interactive clouds of scientific concepts [59]. A topic model-based interface for improving exploratory search was proposed in another work [60]. The effectiveness of using relevance feedback in a literature search was also studied in the bio-medical domain [61]. Finally, a system that supports question-answering in the bio-medical domain was developed (BioMed Explorer [62]). The system allows the users to ask follow-up questions (taking into account the original question) and expand queries with terms extracted from articles.

In this thesis, we propose a novel mode of interaction between the user and the system to assist the user in the case of a difficult search query. Different from the previous works, we propose an approach that optimizes the user-system collaboration throughout the process.

Automatic assessment of research articles Several previous works have studied the problem of automatic quality assessment of different types of documents such as Wikipedia pages [63], news articles [64], and student assignments [65]. In this thesis, our focus is on research articles that have different characteristics than the previously studied documents.

Automating the assessment of research articles is crucial for accelerating scientific research.

For example, the growing volumes of pre-print articles that are published using online platforms, such as Arxiv, may benefit from such an approach [66]. Yet, there has not been much literature on addressing this problem due to the lack of appropriate data sets. One of the reasons for this is that the full text of research articles, article reviews, and accept/reject decisions are hard to obtain due to copyright and privacy considerations. Recently, a data set of research articles and peer-reviews was released that boosted the research on the topic [10]. Using this data, various novel tasks were proposed and studied. Several works have studied the task of review aspect score prediction using either the textual reviews [67, 67, 68, 69] or the article’s text [70, 71]. Other works focused on predicting an accept/reject decision for an article [67, 72, 73]. Finally, the task of automatic generation of a textual review for a research article was also studied [48].

In this thesis, we focus on the task of predicting review aspect scores using the article’s text. We propose to use various features that are generated using probabilistic topic models to complement the existing approaches. Furthermore, using topics has the advantage of being more interpretable than the previously studied features, which are mainly based on deep neural networks.

CHAPTER 3: FIGURE RETRIEVAL FROM COLLECTIONS OF RESEARCH ARTICLES

Research figures are important elements in research articles. For this reason, developing systems and approaches that use them directly can be useful for researchers. In this chapter, we introduce and study the novel problem of figure retrieval. As a first step of studying this problem, we focus on the textual representation of queries and figures and study the effectiveness of different approaches for figure representation and several retrieval models for the task. We perform experiments using figures from the ACL Anthology of papers in the natural language processing domain. The results shed light on the effectiveness and challenges of using the different approaches for the task and motivate the further study of this problem.

3.1 INTRODUCTION

Devising intelligent systems to assist researchers and improve their productivity is crucial for accelerating research and scientific discovery. Tools for literature search such as Google Scholar and many digital library systems are essential for researchers; their effectiveness directly affects the productivity of researchers. Conventional literature search systems often treat a literature article as a retrieval unit (i.e., a document) and the retrieval task is to rank articles in response to a query. In this chapter, we introduce and study a novel retrieval task where we would treat a figure in a literature article as a retrieval unit and the retrieval task is to return a ranked list of figures from all the literature articles in a collection in response to a query.

An effective figure retrieval system is useful in many ways. First, major scientific research results (e.g., precision-recall curves in information retrieval research) are often summarized in figures and key ideas of technical approaches (e.g., neural networks and graphical models in machine learning research) are often illustrated with figures, making figures important “information objects” in research articles that researchers often want to locate and pay special attention to. While one can also navigate into relevant figures after finding a relevant article, it would be much more efficient if a researcher can directly retrieve relevant figures by using a figure retrieval system. Second, a figure search system may supply useful features for improving the ranking of literature articles in a conventional literature search system by rewarding an article whose figures also match well with a query. Third, a figure search system can be very useful for finding examples of illustrations of a concept, thus potentially having broad applications beyond supporting researchers to also generate benefit in education. For

example, a figure search engine operating on a collection of research articles in the natural language processing domain can conveniently allow anyone to find some examples of parse trees, which would be useful for learning about a parse tree or just citing an example in a tutorial of natural language processing.

As a retrieval problem, figure retrieval is different from conventional retrieval tasks in many ways, making it an interesting new problem for research. First, the types of information need of users in figure retrieval are expected to be different than in document retrieval, thus potentially requiring the development of novel approaches to satisfy those needs. Another challenge in figure retrieval is how to effectively represent a figure in the collection. One way to represent figures is to treat them as independent units (i.e., image files). However, such a representation does not benefit from the rich context of a figure in the research article that contains the figure. For example, text in the article that explicitly describes the figure as well as other related parts of the article can be used to represent a figure. Finally, it would be important to study models for measuring the relevance between a figure and a query.

In this thesis, as a first step, we focus on textual queries (i.e., keywords) and represent figures using text extracted from their articles. We propose multiple ways to represent figures and study their effectiveness when using different retrieval methods. Specifically, we propose to represent a figure using multiple textual fields, generated using text in the article that explicitly mentions the figure and also other text in the article that might be related. We then use existing retrieval models, based on lexical similarity and semantic similarity, to measure the relevance between a figure field and a query. Finally, a learning-to-rank approach is used in order to combine different figure fields and retrieval models.

We perform experiments using research articles from the natural language processing domain (ACL Anthology). Since no data sets of queries for figure retrieval are publicly available, we created an initial test collection for evaluation in which figure captions are used to simulate queries (thus, the task is to retrieve a single figure using its caption). While having some limitations, using this data set we were able to obtain some interesting preliminary results. Specifically, our experimental results show that it is beneficial to use a rich textual representation for a figure and to combine different retrieval models. We also gain some initial understanding of the figure retrieval problem, including some illustration of potential types of information need and possible difficulties and challenges. We conclude this chapter by suggesting a road map for future research on the task.

3.2 RELATED WORK

In most retrieval tasks, the retrieval units are documents, though the retrieval of other

units, notably entities (e.g., [74, 75, 76, 77]) and passages (e.g., [78, 79, 80]) has also been studied. The motivation is that these units can serve as a better response to some queries than an entire document. In the research domain, the retrieval of some article elements was also studied, including, for example, formulas [81] and data sets [82]. Our work adds to this line of research a new retrieval task where the retrieval units are figures in scientific research articles and increases our understanding of how to develop effective retrieval models for this new task.

As an effective way to communicate research results, figures are especially useful in domains such as the biomedical domain. As a result, how to support biologists to search for figures has attracted a significant amount of attention, and multiple systems were developed [52, 83, 84]. These previous works have focused on the development of a figure search engine system from the application perspective, but none of those systems or algorithms used in those systems has been evaluated in terms of retrieval accuracy.

Some works [51, 85] studied the ranking of figures within a given article based on the assumption that figures in an article have different levels of importance. These works suggested a set of features for ranking so as to measure the centrality of a figure in the article. The suggested features, however, have not been used for figure retrieval. In this thesis, we analyze the performance of our approach as a function of the figure centrality in the article, which serves as a first step toward utilizing such features for figure retrieval in the future.

In another line of work, methods for extraction of text from figures in the biomedical domain were studied (e.g., [86, 87, 88]). Using the text inside a figure can potentially improve retrieval effectiveness by enriching the figure representation. Yet, these works focused mainly on testing the text extraction accuracy, and not the retrieval effectiveness. In the work described in this thesis, we focus on studying the effectiveness of general figure retrieval models, which we believe is required in order to establish a solid foundation for research in figure retrieval; naturally, the general retrieval models can be enhanced by using many additional techniques to enrich figure representation to further improve accuracy as happens in many other applications such as Web search, which we leave as an interesting direction for future work.

Finally, our work is also related to the large body of work on image search. As an effort for improving image search, the ImageCLEF Track was established. In one task, for example, participants were asked to devise approaches for ranking images in the medical domain using visual and textual data [89]. Content-based Image Retrieval (CBIR) was also explored in some works [90, 91]. In CBIR, the idea is to extract visual features from the image (e.g., color, texture, and shape) and use them for ranking with respect to an image query. Other works focused on combining visual and textual data for image representation and retrieval

(e.g., [92, 93, 94]). Figures in research articles can also be viewed as images, but we study the problem from the perspective of the textual representation of figures. An interesting future work would be to try to incorporate some of the approaches for image search in figure retrieval.

3.3 FIGURE RETRIEVAL

In this section, we introduce and define the new problem of figure retrieval from collections of research articles, discuss strategies for solving this problem, and present specific retrieval methods that we will later experiment with.

3.3.1 Problem Formulation

As a retrieval problem, figure retrieval treats each figure in a research article as a retrieval unit. As those figures do not naturally exist as well separated units, the notion of a collection in figure retrieval is defined based on a collection of research articles D , which can be used to build a collection of figures F_D as follows. For every article $d \in D$, k_d figures are extracted; each figure can be uniquely identified in its article by a number $i \in \{1, \dots, k_d\}$. Then, all figures, extracted from all articles in D , constitute the figure collection F_D .

The goal of the figure retrieval task is to rank figures in F_D according to their relevance to a user query q , where q can be a set of keywords (i.e., textual), an image, or a combination of the two. In general, users may use keywords to describe what kind of figures they want to find and may also (optionally) use one or multiple example images to define what kind of figures should be retrieved. As a first step in studying this problem, we only consider keyword queries, though we should note that a full treatment of the figure retrieval problem should also include matching any user-provided examples of images with the figure collection, which would be a very interesting direction for future work.

With a keyword query, the figure retrieval problem is quite challenging because it requires matching a keyword query with a figure, which does not necessarily have any readily available text description. Fortunately, we can extract relevant text information from the article with a figure to represent the figure; indeed, all figures have captions, which we can conveniently use to represent them. We can also extract any sentences discussing a figure in an article as an additional text description of the figure. This way, we would obtain a pseudo text document to represent each figure, which we refer to as a *figure document*. Thus, our figure collection contains a set of figures where each figure is associated with a figure document, and the main task for retrieval now is to match a query with those figure documents. This

transformation of problem formulation allows us to leverage existing text retrieval models to solve the problem. As in many other retrieval tasks, in order to develop effective approaches to figure retrieval (and any other retrieval tasks in general), there are two key components that should be studied:

1. How to derive effective text representations of the figures?
2. How to measure the relevance between a figure and a query?

We discuss each of these components next in detail.

3.3.2 Figure Representation

While figures can be treated just as independent images (i.e., sets of pixels), they appear in the context of research articles, which offers opportunities to build a rich representation for them. For example, text in the article that explicitly mentions the figure can be utilized. Such text can be the figure caption or other parts of the article that describe or discuss the figure. Other text in the article may not explicitly mention the figure but can still be useful. The abstract of the article, for instance, may serve as a textual representation of the figure since both are in the topic of the article. Finally, other information can be derived from the context of the article which is not necessarily textual. The “authority” of the article (e.g., the number of citations) can serve as a prior for the figure relevance. Our approach to the computation of figure representation is to generate a set of textual fields for each figure, using text that explicitly mentions the figure, as well as other parts of the article.

Explicit figure mentions We generate textual fields using text in the article that explicitly mentions the figure. The caption of the figure, for example, can be regarded as such text. Nevertheless, since figure captions serve as queries in our experiments, we were not able to use them for figure representation at this point. Thus, we only utilize text in the article that discusses or describes the figure (e.g., “The results for the experiment are depicted in Figure 1 ...”). While the general location of such text can be detected easily (since the figure number is explicitly mentioned), it might be challenging to determine its boundaries. That is, automatically detecting at what point in the text the discussion about the figure begins, and at what point the subject changes. A similar problem has been studied in the context of identifying the text that describes a cited article [95]. Yet, it was not studied, to the best of our knowledge, for figure retrieval. In this thesis, we take the following approach for extracting this type of text. Given an explicit mention of a figure (i.e., the string “Figure

i ”), we include w words that precede the figure mention and w words that follow it; w is a free parameter. We denote these textual fields as **FigText** fields and generate three such fields for $w \in \{10, 20, 50\}$. In the case where a figure is mentioned several times in the text, we concatenate all of the text segments that correspond to the different mentions to form a single textual field for a given value of w ; overlapping texts are merged so as to avoid textual redundancy.

General article text Other parts of the article that do not explicitly mention the figure can also be useful for figure representation. This might be the case since a figure is usually related to some of the topics of the article, and these topics may also be discussed in some other parts of the article. Using this type of text can be potentially advantageous when the text that explicitly mentions the figure is very short or not highly informative. In such a case, other parts of the article can help to bridge the lexical gap between the query and the figure when measuring the relevance between them. We denote this type of fields **FigArticle** fields. We use the title, abstract, and introduction of the article to generate three separate fields, denoted **Title**, **Abs**, and **Intro**, respectively. By using these sections of the article we can obtain textual fields with different levels of length and generality. We do not use other parts of the article as these may be too general (e.g., using the entire text), or too narrow (e.g., using sections that describe the model). Furthermore, these three sections appear in almost every research article and are easy to detect automatically.

An alternative approach for using the text of an entire article section would be to select only parts of it that are presumably more related to the figure. Motivated by a previous work [96], we select a single sentence from the abstract to represent a figure. This sentence serves as an additional field and is denoted **Abs-sen**. We select a single sentence from the abstract in the following way. We measure the similarity between a sentence in the abstract and the figure using the cosine similarity between their *tf.idf* representations; a figure is represented using the FigText field ($w = 50$). Then, we choose a single sentence with the highest similarity. If the scores for all abstract sentences with respect to a figure are zeros, we do not represent the figure with a sentence from the abstract. In that sense, using this field we can somehow measure the centrality of the figure in the article (i.e., if the similarity with all abstract sentences is zero then the figure is not likely to be central). The importance in considering the figure centrality was discussed in previous works [51, 85].

3.3.3 Retrieval Models

As each figure is represented by a figure document which consists of multiple text segments,

inconventional retrieval models are applicable to measure relevance. Our study thus focuses on understanding how effective the basic standard retrieval models are for this new retrieval task, and what kind of representation of figures is the most effective. Specifically, we generate a set of features for each figure where each feature corresponds to a combination of a textual field and a retrieval model and use these features to learn a ranking function using a learning-to-rank (LTR) algorithm [97]. We use LTR so as to effectively combine the different retrieval models and textual fields. Furthermore, LTR offers a flexible framework for adding more features in the future that are not necessarily generated using text data.

In our experiments, we considered two retrieval models in order to measure the relevance between a query and a textual field. The first model we use is **BM25** [98]. This model can also be viewed as a model that measures the lexical similarity between the query and some text as it heavily relies on exact keyword matching. The second model that we use is based on word embeddings (e.g., Word2Vec [99]). Specifically, word embeddings can be used to measure the semantic similarity between the query and a textual field, thus this approach is expected to be complementary to BM25. We learn an embedding model using the entire collection of research articles. Then, we represent the query and a textual field using the *idf* weighted average of their term vectors. Finally, the similarity between them is measured using the cosine function. This retrieval approach is denoted **W2V** in our analysis of experimental results.

3.4 EVALUATION

Our main goal is to study the effectiveness of the various approaches we proposed for computing figure representation and ranking figures. Unfortunately, as figure retrieval is a new task, there does not exist any test collection that we can use for evaluation. Thus, we first need to address the challenge of creating a test collection.

3.4.1 Test Collection Creation

A test collection for figure retrieval generally consists of three components: (1) a collection of figures, (2) a set of queries, and (3) a set of relevance judgments. We now discuss how we construct each of them and create the very first test collection for figure retrieval.¹

Figure collection To construct a figure collection, we leveraged the ACL Anthology reference corpus [100]. This is one of the very few publicly available full-text article collections.

¹Available at <https://figuredata.web.illinois.edu> (accessed August 25, 2021).

This corpus consists of 22,878 articles whose copyright belongs to ACL. Figures and their captions were extracted from all articles in the corpus using the PdfFigures toolkit [101], resulting in a collection of 42,530 figures; figures that were not mentioned in the text of the article at least one time were excluded from the collection. In order to extract the full text from the PDF files of the articles, we used the Grobid toolkit.²

Queries data set and relevance judgments Ideally, we should create our query set based on real queries from users. Unfortunately, there are no such queries available to us. To address this challenge, we opt to use figure captions as queries with the assumption that if a user would like to search for figures, it is conceivable that the user would use a sentence similar to a caption sentence of a figure. One additional benefit of this is that we can then assume that the figure whose caption has been taken as the query is relevant to the query and thus should be ranked on the top of other figures by an effective figure retrieval algorithm. Of course, we have to exclude the caption sentences from the representation of the figure, or otherwise, the relevant figure would be trivially ranked on the top of other figures by every ranking method. The other figures are assumed to be non-relevant. We note that this assumption is clearly invalid as some of those figures may also be relevant. However, it is still quite reasonable to assume that the figure whose caption has been used as a query should be regarded as more relevant than any other figures, thus measuring to what extent a method can rank this target figure on top of all others is still quite meaningful and can be used to make relative comparisons of different methods. To further improve the quality of the queries, we use only captions that have between 2 and 5 words (not including stopwords), resulting in 16,829 queries; 17%, 33%, 30%, and 20% of the queries in the data set are of length 2, 3, 4, and 5, respectively. The data set of queries was split at random such that one half was used for training the LTR algorithm and the other half was used for evaluation.

3.4.2 Implementation Details

The Lucene toolkit was used for experiments.³ Krovetz stemming and stopword removal were applied to both queries and figure fields. For our word embeddings-based retrieval model, we trained a Continuous Bag of Words (CBOW) Word2Vec model [99] with a window size of 5 and 100 dimensions.⁴ We used the LambdaMart algorithm [102] in order to learn an

²<https://github.com/kermitt2/grobid> (accessed August 25, 2021)

³<https://lucene.apache.org> (accessed August 25, 2021)

⁴<https://radimrehurek.com/gensim/models/word2vec> (accessed August 25, 2021)

LTR model.⁵ Using the LTR model for ranking the entire collection of figures is not practical as several features are quite expensive to compute for all figures (e.g., word embeddings). We address this issue by adopting a 2-phase retrieval paradigm as follows. We perform an initial retrieval of 100 figures using the FigText field with $w = 50$ (and the BM25 retrieval model). Then, we re-rank the result list using the LTR model with the entire set of features.⁶ We use the Mean Reciprocal Rank ($MRR@100$) as an evaluation measure that is appropriate for our scenario in which there is a single relevant figure for a query. We also measure the success at the top k ($\in \{1, 3, 5, 10\}$) documents, denotes $succ@k$; $succ@k$ is the fraction of queries for which the relevant figure is among the top k results.

3.4.3 Experimental Results

Main result The performance of our suggested approach for the figure retrieval task is presented in Table 3.1. We compare the effectiveness of the initial retrieval with that of the re-ranking approach in which LTR was used. In the case of LTR, we report the performance of using different figure fields and different retrieval models. The LTR performance when the BM25 retrieval model is used is reported in the upper block of the table. According to the results, this approach outperforms the initial retrieval by a very large margin when FigText fields are used. This result attests to the benefit of using different sizes of window for the FigText fields (recall that only a single window size of 50 was used for the initial retrieval). Using the FigArticle fields, on the other hand, results in an ineffective LTR model compared to the initial retrieval. Yet, according to the results, there is clear merit in combining FigText and FigArticle fields. When W2V is used as a retrieval model, we can see that it is not effective with respect to the initial retrieval. Furthermore, as in the case of BM25, FigText fields are more effective than FigArticle fields when W2V is used. Finally, when all figure fields and all retrieval models are combined, the highest performance is achieved for all evaluation measures. We conclude, based on Table 3.1, that the most useful figure fields are the FigText fields and the most effective retrieval model is BM25. The W2V retrieval model and the FigArticle fields, on the other hand, are not very effective when used alone and only improve performance when added on top of the other features.

Analysis of individual fields The performance of using individual FigText fields and FigArticle fields for re-ranking the initial result list is reported in Figure 3.1(a) and 3.1(b), respectively. In each graph, the performance (MRR) when a single field is used is reported

⁵<https://sourceforge.net/p/lemur/wiki/RankLib> (accessed August 25, 2021)

⁶We made the training/test data of the LTR algorithm publicly available as part of the data set.

Table 3.1: Main result. Figure retrieval performance when different figure fields and different retrieval models are used. The differences in MRR between all LTR models and the initial retrieval are statistically significant (two-tailed paired t-test, $p < 1.0e - 7$).

		MRR	$succ@1$	$succ@3$	$succ@5$	$succ@10$
Initial Retrieval		.443	.353	.497	.547	.607
LTR						
BM25	FigText	.478	.391	.531	.577	.639
	FigArticle	.126	.079	.142	.172	.218
	FigText+FigArticle	.483	.394	.538	.586	.648
W2V	FigText	.212	.129	.233	.291	.377
	FigArticle	.070	.026	.064	.096	.154
	FigText+FigArticle	.212	.127	.230	.289	.380
BM25+W2V	FigText+FigArticle	.487	.398	.541	.592	.649

(blue bar) as well as when a single field is used together with all the fields presented to its left (i.e., accumulative performance; orange bar); BM25 was used as a retrieval model. According to Figure 3.1(a), all FigText fields are quite effective and the re-ranking performance increases with the size of the window. Moreover, there is a clear benefit in combining different sizes of the window as the accumulative performance also increases as a function of the window size. A possible reason for this is that the length of the text which describes a figure can often vary. In this thesis, we address this issue by using different values for the text length. In future work, we plan to explore automatic approaches for setting this value dynamically on a per-figure basis. As for the FigArticle fields, the performance increases as a function of the average field length. That is, the lowest performance is achieved for the title and the highest performance is achieved for the introduction. As in the case of the FigText fields, we can see that there is always an added value when using multiple fields.

Figure centrality analysis A figure in a research article can be mentioned in the text several times. We define the number of figure mentions as the number of times the figure number was explicitly mentioned in the article (i.e., the number of mentions of figure i is the number of appearances of the string “Figure i ” in the text). We examine the performance of using different figure fields (using BM25) for re-ranking the initial result list as a function of the number of figure mentions in Figure 3.2. Figures with 1, 2, 3, 4, and 5 (or more) mentions constitute 65%, 23%, 7%, 3%, and 2% of the entire figures in the test set, respectively. The performance of using the FigText fields is depicted in Figure 3.2(a). According to the graph, the poorest performance is achieved when the figure has only one mention and the highest performance is achieved for two mentions. Furthermore, increasing the number of mentions

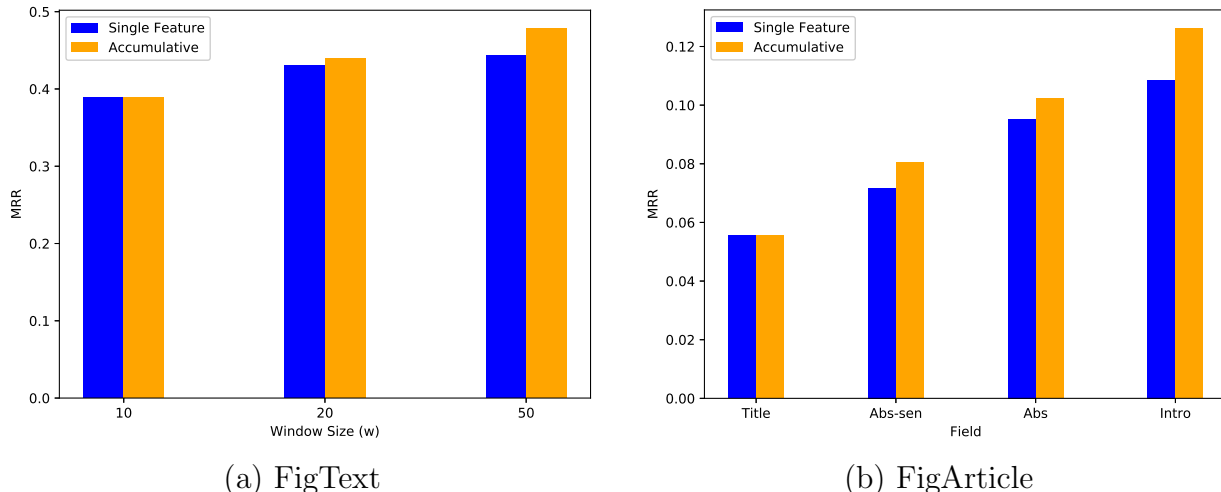


Figure 3.1: Performance of using individual figure fields. The performance of the FigText and FigArticle fields is depicted in Figure (a) and (b), respectively.

to more than two almost always results in a performance decrease. A possible explanation for this can be that when the figure is mentioned many times, there are high chances for the window of text to include irrelevant text. The results for the FigArticle fields are presented in Figure 3.2(b). According to the graph, the performance almost always increases with the number of mentions for all fields. An explanation for this can be that once the figure is mentioned many times in the article, there are high chances that it describes a central topic in the article. Consequently, the text that does not explicitly describes the figure is expected to serve as a more reliable representation of the figure. Further exploration revealed that adding the number of mentions as an additional feature in the LTR algorithm does not result in further performance gains. An interesting future work would be to explore the effectiveness of more features that capture the centrality of a figure in an article as suggested in previous works [51, 85].

Table 3.2: Representative queries and the rank of the relevant figure.

Query	Rank	Query	Rank
(1) dialog strategy architecture	6	(6) word gloss algorithm	2
(2) dependency tree english sentence	2	(7) precision recall graph query	32
(3) performance official runs	1	(8) example graphic tree	1
(4) full simulation naive bayes f1	9	(9) graphical model sdtm	1
(5) hierarchical recurrent neural network	1	(10) example dependency tree	0

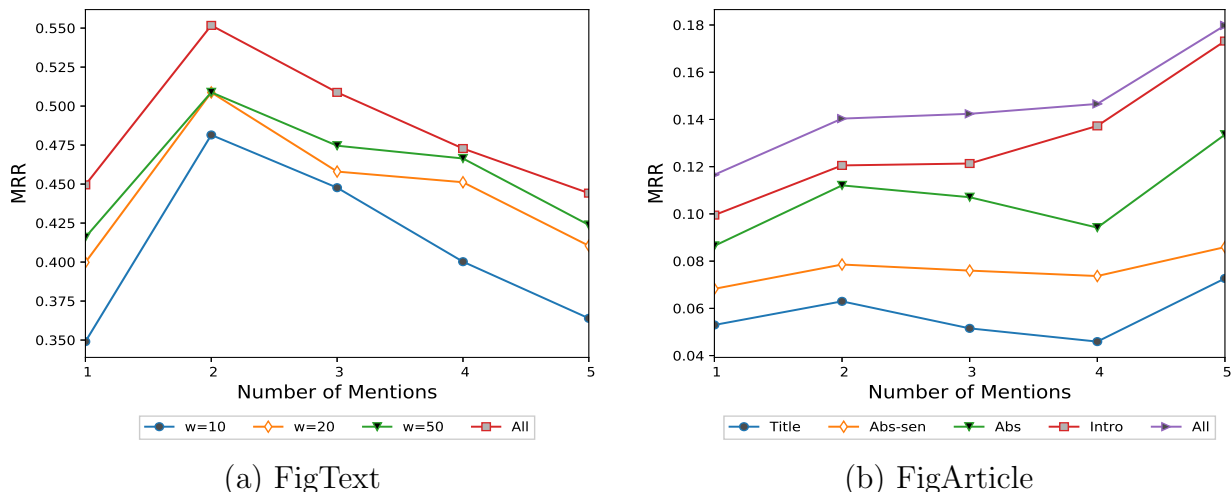


Figure 3.2: Performance of using different figure fields as a function of the number of mentions of the figure in the article. “All” refers to using all fields. The value of ‘5’ in the x-axis refers to figures with *at least* five mentions.

Query analysis In Table 3.2, we provide ten representative examples of queries with variable performance and information needs and the corresponding rank of the relevant figure when all features are used for re-ranking the initial result list. (Rank=0 means that the relevant figure did not appear in the top 100 results.) The queries in Table 3.2 help to illustrate the different information needs that can be addressed by figure retrieval. For example, queries 4 and 7 describe a need for experimental results, while queries 5 and 9 describe a need for some model. Table 3.2 also helps to illustrate the variance in performance of different queries. For example, query 10 fails to retrieve the relevant figure presumably since this query is very general, resulting in many other figures that match those keywords. Other queries are well specified (e.g., query 4) and thus result in a much better performance. As we already mentioned, one limitation of our experiments is that only one figure is considered relevant for a query. Thus, it is plausible that in a more realistic scenario we would be able to see much better performance for these queries. Nevertheless, these examples help illustrate the potential information needs in figure retrieval and the difficulty of some queries.

We perform an analysis of the query topics in order to gain further understanding about the types of information need in figure retrieval and the effectiveness of their corresponding queries. Specifically, we learn an LDA topic model [103] using all queries in both training and test set. (We use the MeTA toolkit to learn the topic model [104].) Ten words with the highest probabilities in five topics are presented in Table 3.3. We also present the performance of each topic, which is calculated as follows. We first assign a topic for each query. This topic is the one with the highest probability in the multinomial distribution over

topics for this query. Then, we report the average *MRR* of the queries in each topic. (Each topic ended up containing about 20% of the queries.) The results in Table 3.3 illustrate potentially five types of information need. For example, Topic 1 contains words that are frequently used in figures that describe examples in the ACL corpus (e.g., “example”, “tree”, and “parse”). Words that describe a model or an algorithm, on the other hand, can be seen in Topic 2. Finally, Topic 3 contains words that are related to the description of experimental results (e.g., “accuracy” and “performance”). Examining the performance of the different topics, we can see that it can be very different. For example, the worst performance is achieved for Topic 1 (potentially queries for retrieving examples), and the best performance is achieved for Topic 4 which presumably describes an information need for an experimental setup (e.g., “corpus”, “annotation”, and “text”).

Table 3.3: Query topics (LDA). The average performance of the queries in each topic in terms of *MRR* is reported in the parenthesis.

Topic 1 (.417)	Topic 2 (.506)	Topic 3 (.501)	Topic 4 (.541)	Topic 5 (.471)
example	example	result	example	system
tree	algorithm	distribution	sample	architecture
sentence	model	accuracy	annotation	overview
parse	rule	different	model	result
structure	learning	set	corpus	process
dependency	word	score	dialogue	question
derive	alignment	data	interface	framework
sample	base	performance	entry	evaluate
graph	process	comparison	structure	flow
rule	graph	training	text	example

3.5 CONCLUSIONS

In this chapter, a novel task of figure retrieval from collections of research articles was suggested and studied. According to the new retrieval task, we treat figures of research articles as retrieval units, and the goal is to rank them with response to a query. We proposed and studied different approaches for building a representation for a figure using the article text and various retrieval methods. Our empirical evaluation demonstrated the benefit of using a rich textual representation for a figure and of combining different retrieval models. Furthermore, an analysis of the queries in the data set has shed some light on the potential information needs in figure retrieval and their relative difficulty. The focus of this chapter was on textual representations of figures. In the next chapter, we will explore

a different way to represent figures based on embedding vectors that can be learned from unlabelled data.

Figure retrieval is a very promising novel retrieval task; an effective figure search engine would enable researchers to increase productivity, thus accelerating scientific discovery. Our work in this thesis is only a small initial step; there are many interesting novel research directions that can be further studied in the future which we briefly discuss below.

First, as there does not exist any test collection for figure retrieval, evaluation of figure retrieval is quite challenging. Although we created a test collection, which allowed us to make some interesting relative comparisons of different methods, the test collection we constructed has two limitations: (1) captions do not necessarily represent information needs of real users; (2) captions have only one relevant figure. This data set allowed us to gain some initial understanding of the problem and study the relative effectiveness of different approaches, but those findings have to be further verified with additional experiments. Thus, a very important future work is to build a more realistic data set using a query log and verify our findings. For this reason, we developed a system to facilitate the collection of data sets for the task that we describe in detail in Chapter 7.

Second, related to the challenge of constructing a test collection is a better understanding of the information needs in figure retrieval. To that end, it is necessary to conduct a user study in order to obtain some realistic queries. It would also be interesting to study what kind of queries are harder to answer. Another interesting question would be whether there are some common types of information need shared among different research disciplines. A thorough understanding of the users' information needs is also crucial for devising effective retrieval methods that are optimized with respect to user needs.

Third, in this chapter, we assumed that the user query is textual. However, in the most general case, the query can involve both textual and visual information. For example, the user would describe an information need using text and also provide figure examples. This raises the question of how to create an effective representation of the user query. To that end, it would make sense to leverage ideas from the area of computer vision, creating an interesting opportunity for interdisciplinary research of IR and computer vision. Furthermore, different representations of the query may also necessitate the development of new ranking models that have to combine multiple ranking criteria.

Figure representation is another subject worth exploring in future work. In this chapter, we used only textual information for figure representation. In the general case, however, it might be useful to combine different types of information. For example: text data, visual information, article citation information, and figure centrality information. One line of work in this direction would be to identify useful sources of information. Another direction would

be to combine heterogeneous information into an effective figure representation. In Chapter 4, we study distributed representations of figures using both textual and image data.

Finally, devising approaches for the extraction of relevant information for representing a figure is also important. For example, devising methods for automatically identifying the text in the article that discusses a figure to enhance retrieval accuracy is an interesting direction for future work.

CHAPTER 4: A STUDY OF DISTRIBUTED REPRESENTATIONS FOR FIGURES OF RESEARCH ARTICLES

In Chapter 3, we introduced the problem of figure retrieval and studied the effectiveness of different figure representations for the task. The main approach taken for figure representation in the previous chapter was based on textual information using the bag-of-words approach.¹ In this chapter, we study the effectiveness of distributed representations (embeddings), built using text and image data, for research figures. Using figure embeddings is advantageous compared with the bag-of-words representation. For example, embeddings can measure the semantic similarity between figures more effectively. We implement distributed representations using different model architectures and loss functions and compare them with the bag-of-words baseline. We also propose a novel weak-supervision approach to obtain training data for the task of learning the representations. Our results, using the ACL Anthology data set, show that distributed representations are more effective than bag-of-words. Yet, the combination of the two approaches can further improve performance.

4.1 INTRODUCTION

Figures are entities in research articles that play an essential role in scientific communications. For example, figures often summarize the main empirical results of an article and visualize algorithms and models. As such, figures can be potentially used in many tasks to facilitate and accelerate research. Thus, while data mining and information retrieval techniques can be applied to articles in general to improve literature systems' performance, it would be especially beneficial to apply these techniques directly to figures.

Figure representation is a fundamental problem in all applications involving figures. Different from general images, figures are complex research entities that are associated with various sources of data of various modalities (e.g., text data of different types and sources, numeric, and image data), posing unique novel challenges for representation learning. Thus, the study of how to optimize representation specifically for research figures is crucial. Despite that, this problem has not been well studied in previous works. The dominant approach, as discussed in Chapter 3, is to represent a figure by its companion text data in an article using the bag-of-words model. Using this representation of figures has some limitations. First, it does not consider any other types of non-textual features, such as image features. Second, it has limited capability in accommodating the inexact matching of semantically related words.

¹We also used the Word2Vec approach but the focus was on a simple application of this approach to measure the similarity between two bag-of-words representations of the query and the figure.

In this chapter, we address the limitations of our work in the previous chapter and study a new view of representation for figures, namely deep neural network-based distributed representations. Learning distributed representations for many real-world entities is very successful in recent years [105, 106]. The main idea behind the different approaches in this scope is to learn embeddings of those entities using large amounts of data where the goal of learning is to capture the complex relations between the entities. For example, learning an embedding vector representation of words [106, 107] has proven to be useful for many text applications. Specifically, word embeddings can effectively address some of the limitations of bag-of-words representations, such as measuring the semantic similarity between words.

In this chapter, our goal is to study the effectiveness of distributed representations for figure representation, exploring the learning of such a representation from multiple views. Specifically, we focus on using both image data and text for learning representations with different model architectures and loss functions to understand how sensitive the embeddings are to the learning approach and the features used.

One technical challenge in learning deep neural network-based representations is that it requires massive amounts of data that is not available for this domain. While word embeddings can be easily learned by leveraging the co-occurrences of words in large amounts of text data, the amount of figure data is quite limited. To overcome this problem, we propose and study two strategies. The first is to leverage massively pre-trained models on general data (e.g., BERT [107]). The second is a novel weak supervision approach that can generate a large amount of training data by leveraging the already existing citation relations between research articles.

We use a collection of figures from the ACL anthology to empirically study the effectiveness of different representations by their ability to measure the semantic similarity between research figures. We also study the effectiveness of embeddings on the downstream application of recommending figures of interest based on an input (query) figure. The results show that the embeddings approach is generally more effective than bag-of-words, yet combining the two approaches provides the best performance. We also show that pre-trained image/text embeddings have limited effectiveness compared with the weak supervision approach and even the bag-of-words approach. Finally, the results show that embeddings for figures can be somewhat sensitive to the learning techniques. Specifically, the relatively simple model architectures are the most effective ones, text features are more effective than image features, and the combination of image and text features is the most successful approach.

4.2 RELATED WORK

There has been growing interest recently in learning vector representations of real-world

entities using deep neural networks. This led to the development and study of various embedding models for representing different entities such as words [106, 107], sentences [108], and images [109]. Our work in this chapter can be regarded as the first one to study the effectiveness of embedding-based representations for figures of research articles.

Learning embeddings using neural networks often requires massive amounts of data. To address this, there has been an active research direction exploring the use of weak supervision for learning [110, 111]. Our work in this chapter adds to the existing work a new line of application of weak supervision for learning figure embeddings.

There have been several previous works that studied various figure retrieval and mining tasks [40, 51, 83, 112]. One work focused on the prediction of figure type [40]. In another work [112], a model for linking figures to sentences in the abstract of the article was studied. Finally, there has been one work that studied the task of figure retrieval [7]. These previous works mostly relied on the bag-of-words representation of figures. In this thesis, we explore distributed representations of figures that can benefit a variety of tasks that involve figures.

Previous works have studied the joint embedding of images and text, focusing mostly on images that contain different objects and text that identifies the objects and the interactions between them (e.g., “An apple on a table”) [93, 94, 105, 113, 114, 115]. The main idea in many of these works was to embed image and text to the same space. Learning joint embeddings for image and text aims to find a common representation that can explain both and is thus less appropriate for research figures in which image and text are often two types of complementary information. Thus, in this thesis, we learn text and image features separately and then combine them using a third model. Using this strategy is sufficient for studying the different aspects of the problem that we are interested in, such as the effectiveness of various architectures for image/text modeling, the effectiveness of image and text feature combination, and the effectiveness of pre-training vs. weak supervision. We thus leave the study on finding the optimal integration of image and text features for figures for future work.

4.3 FIGURE EMBEDDINGS

4.3.1 Problem Definition

A collection of figures F_D can be generated using a collection of research articles D by extracting the figures from all articles. Each figure can be associated with different types of data of different modalities. For example, a figure can be associated with a caption, the abstract section of its article, an image, and a set of numbers. In our study, as a first step,

we focus on learning figure embeddings using only text and image data. Given two figures in the collection, f_i and f_j , the goal is to learn corresponding vectors in a continuous space, \vec{f}_i and \vec{f}_j , such that the distance between them in that space is inversely proportional to their semantic similarity. In this thesis, we use neural networks to learn these representations of figures.

4.3.2 Textual Representation of Figures

In this thesis, we associate figures with only text and image data. While the image data of a figure is well defined, the textual data for a figure is mostly not readily available. In the general case, the article that contains the figure can be used to extract text that directly describes it (e.g., the figure caption) and text that does not directly describe it, but is related to its topic (e.g., parts of the abstract section). In Chapter 3, we explored the effectiveness of using different types of textual data for figure representation to be used for the figure retrieval task. Based on the findings of that chapter, we generate a textual representation for a figure as follows. We use the caption of the figure, concatenated together with the text in the article that directly describes the figure, for the figure representation. To extract this text, first, the locations in the article where the figure is mentioned are identified. Then, the sentence that directly mentions the figure, one sentence before it, and one sentence after it, are extracted. (In the case of several mentions for the figure, all the text which was extracted is merged.) In the work described in this chapter, we use this text as a single textual input which resulted in a good enough performance. In future work, we plan to take into account the sources of those different texts (e.g., treating differently text that comes from the caption compared to text that comes from other parts of the article).

4.3.3 Model Architecture

To learn figure embeddings using neural networks, we use the Siamese architecture [116]. According to this architecture, given two figures, the same model is used to generate embeddings for both of them. Then, the dot product between the figure vectors is used as a semantic similarity score. The Siamese model is appropriate for our scenario since the two figures are entities of the same type and we also assume the relationship between them is symmetric. We note that the symmetry assumption may not always hold but it is still useful to learn meaningful representations; we thus leave the treatment of asymmetric relationships for future work. The model for our figure embedding approach is composed of three sub-models:

1. An image model that generates visual features.
2. A text model that generates textual features.
3. A fusion model that combines the image and the text features.

While the image and the text model are both Siamese models, the fusion model is a simple feed-forward neural network model. In the remainder of this section, we describe each of these model components.

Text models To generate textual features, we experimented with three models to explore varying levels of complexity, compare auto-regressive to non-auto-regressive models, and compare pre-training to weak supervision training. The first model we used is **LSTM** [117] that generates features for a text using a recurrent neural network to capture long and short-term dependencies. Specifically, our LSTM-based model contains a word embedding layer (learned from scratch) which is followed by a single LSTM layer, where the weights of the last hidden state of the LSTM layer are used as the textual features. The second model we use is **Bi-LSTM** [118]. This model is similar to LSTM but has a higher level of complexity since it models dependencies using both directions of the text. As in the case of the LSTM model, we use a word embedding layer which is followed by the Bi-LSTM layer. Additionally, the Bi-LSTM layer generates two sets of features (backward and forward). The two sets of features are concatenated, a dropout layer is added on top of this concatenation, and a final dense layer is added to obtain the textual features. The last model we use is **BERT** [107] which uses transformers and a self-attention mechanism to learn dependencies in text. This model was shown to achieve state-of-the-art performance in many natural language processing tasks, where the main approach that was taken is to pre-train the model using a very large amount of text data and then fine-tune the output of the model for the specific task. In this thesis, we experiment with three versions of this model. In the first one, we use a pre-trained model and treat the pooled output as the textual features. In the second version, we add a dropout layer, a dense layer with a Relu activation, and a final dense layer on top of the pooled output. Then, we learn the weights of those dense layers using the Siamese architecture; the output of the final dense layer serves as the textual features. In the third version, we use the same model as in the second one but also fine-tune the last layer of BERT.

Image models Previous works on using neural networks for computer vision leveraged massive amounts of data which enabled the learning of complex models with remarkable

performance. Another technique that is highly effective for computer vision is transfer learning in which a model is trained using large amounts of data and then is fine-tuned for a specific task. In this dissertation, our goal is to generate effective image features for figures. This is challenging, however, since we do not have available massive amounts of image training data. Furthermore, since images of figures are quite different than images in the massive training data sets (e.g., ImageNet), it is not clear how pre-training will be useful for our scenario. To answer these questions, we experiment with two models as follows. The first model that we use is a simple Convolutional Neural Network (**CNN**) which is fully trained using the figure image data. The model is composed of two convolutional layers, a max-pooling layer, a dropout layer, a dense layer with Relu activation, and a final dense layer. The second model we use is **DenseNet** [119] which uses densely connected convolutional networks. This model has higher complexity than the simple CNN and we use it since it was previously shown to be very effective for image representation. We use three versions of this model. In the first one, we use a pre-trained model with ImageNet to generate image features (no fine-tuning). In the second version, we add layers on top of the DenseNet model including a dropout layer, a dense layer with Relu activation, and a final dense layer. We then learn the parameters of the dense layers using the Siamese model. In the third one, we use the same architecture as in the second version but additionally fine-tune the last dense block of the DenseNet model.

Fusion model To combine the image and text features, we concatenate them and use a batch normalization layer on top of that. The batch normalization is crucial since the two sets of feature values are often not on the same scale. Finally, we use a single dense layer to generate the figure embedding. We take this approach since we are interested in obtaining a single embedding vector for a figure using different types of features.

4.3.4 Loss Function

We assume that each pair of figures, f_i and f_j , is associated with a numeric semantic similarity score $R_{i,j} \in \mathbb{R}$ (larger values of $R_{i,j}$ correspond to greater similarity). A semantic similarity label $L_{i,j} \in \{0, 1\}$ can be generated using $R_{i,j}$ by setting $L_{i,j}$ to 1 when $R_{i,j} > 0$ and setting $L_{i,j}$ to 0 otherwise. In this chapter, we experiment with three loss functions. The first one is the Cross-Entropy loss, computed using the Sigmoid of the dot product between the two vectors and the semantic similarity label, $CE(\vec{f}_i \cdot \vec{f}_j, L_{i,j})$. Secondly, we use the Mean Squared Error loss, computed using the dot product between the two vectors and the semantic similarity score, $MSE(\vec{f}_i \cdot \vec{f}_j, R_{i,j})$. Finally, we use the triplet hinge loss [109]. The

triplet hinge loss is defined for a triplet of figures, comprised of a query figure \vec{f}_q , a positive figure \vec{f}_+ (related figure), and a negative figure \vec{f}_- (unrelated figure). This loss is defined as: $\max(0, 1 + \vec{f}_q \cdot \vec{f}_- - \vec{f}_q \cdot \vec{f}_+)$. The main idea is that we want a figure to be closer to a related figure compared to an unrelated figure.

4.4 WEAK SUPERVISION FOR FIGURE EMBEDDINGS

In this thesis, since we are dealing with a novel problem, an important issue that needs to be addressed is how to collect training data. Furthermore, since we are interested in learning representations using deep neural networks, there is a need for a large set of training examples. To address this challenge, since log data was not available to us, we propose a novel approach for collecting data for the task using weak supervision. This approach allows us to leverage large amounts of training data that already exist. Specifically, to generate training data, we leverage existing relations between research articles. The approach is depicted in Figure 4.1. First, since we know that two articles are related if one is cited by the other, we assume that two figures are semantically similar if they appear in two articles with a citation relation. Second, we assume that figures that are in the same article are also semantically similar. Although both kinds of relations may be noisy, we expect that most relations are meaningful semantic associations and the learned embedding vectors to be meaningful as in the case of word embeddings where there are also noisy word associations, but they do not significantly affect the results. Comparing the two types of relations, it is reasonable to assume that two figures in the same article are more likely to be more semantically similar than two figures in citing articles and that the latter should be more similar than a random pair of figures. Based on this intuition, we set the semantic similarity score of two figures in citing articles to be lower than the score of two figures in the same article. Finally, we randomly sample pairs of figures from the collection to generate negative examples.

Formally, given two figures f_i and f_j , extracted from the articles $d(f_i)$ and $d(f_j)$, respectively, and given that $C(d(f_i))$ is the list of articles that cite $d(f_i)$ or cited by it, the semantic similarity score $R_{i,j}$ is set to:

$$R_{i,j} = \begin{cases} 1, & \text{if } d(f_i) = d(f_j) \\ 0.6, & \text{if } d(f_j) \in C(d(f_i)) \\ 0, & \text{otherwise} \end{cases} \quad (4.1)$$

We set the values of $R_{i,j}$ this way assuming a range of $[0, 1]$ where $R_{i,j} = 0$ corresponds to figures that are completely unrelated and $R_{i,j} = 1$ corresponds to figures that are highly

related to each other. We thus set the value for figures in citing articles to 0.6 to be greater than 0.5 but lower than 1. These values resulted in a very good performance in our experiments and we leave the investigation of the effectiveness of other values for future work.

When using this data for training the image model, some modifications are required. This is the case since semantically similar figures, as defined by our approach, may have images that are not visually similar. Our goal for the image model is to generate features that can help measure the visual similarity of figures. For this reason, we filter out pairs of figures which are not visually similar enough based on an unsupervised similarity function.² Finally, we do not make a differentiation between figures in the same article and figures in citing articles since the relationship type may not be indicative of different levels of visual similarity. Based on this approach, a pair of figures will be assigned only with a binary relevance label in the case of the image model (consequently, we do not use the MSE loss).

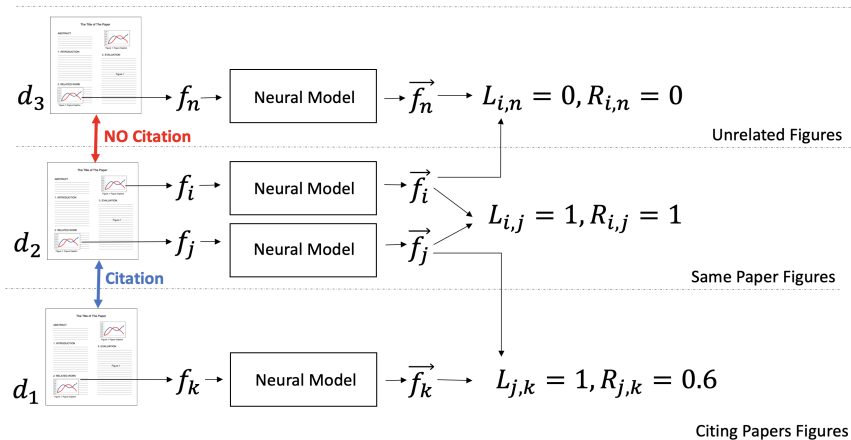


Figure 4.1: A weak supervision approach for learning figure embedding representations.

4.5 EMPIRICAL STUDY

4.5.1 Experimental Setup

Collection of figures We built a collection of figures using the ACL anthology.³ The collection includes 40,367 articles whose copyright belongs to ACL, published until October 2018. Using those articles, we created a collection of 84,340 figures. We used the PdfFigures⁴

²We use the Structural Similarity Index Measure (SSIM) [120] with a threshold of 0.5 for filtering out pairs and a threshold of 0.3 for negative sampling.

³<https://aclweb.org/anthology> (accessed August 25, 2021)

⁴<https://pdffigures2.allenai.org> (accessed August 25, 2021)

toolkit to extract the figure images and the Grobid⁵ toolkit to extract the full text from the PDF files of the articles.

Data pre-processing Text data was Porter stemmed, and stopwords were removed (using the INQUERY list). The collection does not include figures with an associated text (after pre-processing) of less than five words. We resized the images of figures to fit a $224 \times 224 \times 3$ matrix and normalized the features by a factor of 255.

Training data We used 947,335 pairs of figures in citing articles and 202,944 pairs of figures in the same article as related figures. After adding random pairs as negative examples, the data set for training the text network included about 2M pairs of figures. For the training of the image network, we used about 300K figure pairs after removing pairs that were not visually similar enough. For training the fusion network, since we are interested in figures with both text and image data, we used about 1M pairs after filtering out figures with no image data. In this thesis, we train all three components of the model separately (the image model, text model, and fusion model) due to our limited data. To evaluate the different approaches, we only used figures for which both image and text data were available to make it as realistic as possible (57K figures).

Neural network implementation We implemented the neural network models using the TensorFlow library. The values of the different parameters were set based on findings in recent works in the text and image domain. All models were trained for 3 epochs using the Adam optimizer with a batch size of 64 and a learning rate of 0.01. The vocabulary size was set to the 1000 most frequent words in the training data. We used only the first 100 words in the text data of a figure (the figure caption was concatenated first) due to BERT’s limitation on the input size and the limited effectiveness of LSTM for long sequences. The embedding size was set to 50 for all models, which means that the number of hidden layers in LSTM/Bi-LSTM was set to 50 as well as the size of the final dense layer in the other models.⁶ The size of the dense layer on top of BERT, DenseNet, and CNN was set to 100. The dropout rate was set to 0.5. The word embedding layer dimension for the LSTM/Bi-LSTM model was set to 100. For BERT, we used a model with 12 layers, 768 hidden units, and 12 attention heads. For DenseNet, we used a 121-layer model. For the CNN model, we used convolutional layers with 32 filters and a kernel size of 3×3 .

Baselines One of the major research questions we study is whether embedding-based representations can improve over the currently used bag-of-words representations. For this

⁵<https://github.com/kermitt2/grobid> (accessed August 25, 2021)

⁶Our preliminary experiments showed that a larger size of 100 is less effective.

reason, we compare our model with two representative baseline methods: **tf.idf** and **LDA**. For the LDA baseline [103], we learn a model with 50 topics and use the figure distribution over topics as its representation. The vocabulary used for both models was also restricted to the 1000 most frequent words.

4.5.2 Experimental Results

Semantic similarity prediction To evaluate how effective are the different representations in measuring the semantic similarity between figures, we used a binary classification task. Specifically, given two figure vectors, we used the cosine similarity function to get a similarity score, which we then transformed into a binary label using a threshold. Since the threshold value can vary depending on the representation type, a validation set was used to set it (selected from $\{0.1, 0.2, \dots, 0.9\}$). The evaluation is based on three test sets. In the first one, denoted “Same”, we used 500 pairs of figures in the same article (related figures) and 500 randomly sampled (unrelated) pairs. In the second one, denoted “Citing”, we used 500 pairs of figures that appear in citing articles (related figures) and 500 unrelated pairs. Finally, in the third set, denoted “Accuracy”, we combined the first two sets. (The training set did not include those selected pairs.) The sets were balanced such that the accuracy of a random baseline is 0.5. The results are presented in Table 4.1 for using text and image features separately and in Table 4.3 for the fusion model. For the pre-trained models that we fine-tuned (BERT and DenseNet), we added the term “(tuned)” when only the dense layers on top of the model were tuned and “(tuned+)” when the dense layers and also part of the model were fine-tuned.

According to the results in Table 4.1, most text-based and image-based representations perform much better than a random baseline. Focusing on the embedding models that use only textual features, we can see that the LSTM/Bi-LSTM model performs the best with the MSE loss. In the case of the tuned BERT models, on the other hand, there are no substantial differences between the different loss functions. Overall, based on the results, the best text-based embedding model is LSTM. A possible reason for this might be its relatively small number of parameters and the size of the training data set. Also, it is interesting to see that it outperforms the pre-trained BERT model, which might be due to the unique vocabulary used in ACL research articles. Comparing the embedding models with the baselines, we can see that LSTM/Bi-LSTM substantially outperforms all baselines (tf.idf, LDA, and the pre-trained BERT model). We can also see from the results that tf.idf is the best performing baseline. For this reason, we compare the embedding approaches only with this baseline in the rest of the evaluation section. Focusing on BERT, we can see that fine-tuning can

Table 4.1: Semantic similarity prediction: comparing text with image features.

		Accuracy	Same	Citing	
Text Features	tf.idf	.720	.818	.622	
	LDA	.688	.766	.609	
	BERT	.525	.522	.527	
	CE	LSTM	.740	.776	.704
		Bi-LSTM	.732	.743	.720
		BERT(tuned)	.533	.535	.530
		BERT(tuned+)	.534	.534	.533
	MSE	LSTM	.802	.831	.772
		Bi-LSTM	.791	.811	.770
		BERT(tuned)	.527	.527	.527
		BERT(tuned+)	.527	.528	.525
	Hinge	LSTM	.505	.508	.501
		Bi-LSTM	.500	.500	.500
		BERT(tuned)	.522	.525	.518
		BERT(tuned+)	.534	.537	.530
	Image Features	DenseNet	.620	.623	.616
CNN		.500	.500	.500	
CE		DenseNet(tuned)	.635	.641	.629
		DenseNet(tuned+)	.518	.510	.526
Hinge		CNN	.662	.663	.661
		DenseNet(tuned)	.630	.655	.605
		DenseNet(tuned+)	.500	.500	.499

improve its performance, but its overall effectiveness is still low. Another finding from the table is that embeddings outperform the bag-of-words baseline to a greater extent for citing figures than figures in the same article. A possible reason for this might be the soft matching nature of distributed (dense) representations and their ability to identify more loosely related figures. Moving on to the image features, we can see that most of them perform better than a random approach and that the best performing model is CNN. Finally, we can see that using fine-tuning for the DenseNet model can improve its performance. Still, the fine-tuned DenseNet model is not as well-performing as CNN. Comparing the image with text features, we can see that the text features are much more effective.

In light of the results in Table 4.1, an important question that comes up is whether embeddings can replace tf.idf for the textual representation of figures. To answer this, we examine the effectiveness of combining the predictions of tf.idf and embeddings using an “Oracle” in Table 4.2, which serves as an upper bound for the performance of such combination. We focus on effective models according to Table 4.1: LSTM trained with MSE

Table 4.2: Combining tf.idf with text-based embeddings using an ‘Oracle’.

	Accuracy	Same	Citing
tf.idf	.720	.818	.622
LSTM	.802	.831	.772
BERT(tuned+)	.534	.534	.533
LSTM&tf.idf	.914	.941	.886
BERT(tuned+)&tf.idf	.864	.913	.815

and BERT(tuned+) trained with CE. The results show that this combination is of merit as it outperforms the individual models in all cases. Even in the case of BERT, which is not very effective according to Table 4.1, the combination can improve tf.idf substantially. In this dissertation, we are mainly interested in studying distributed representations and thus leave the study of combining the two approaches for future work.

Table 4.3: Semantic similarity prediction: combining text and image features (fusion model).

	Accuracy	Same	Citing	
tf.idf	.720	.818	.622	
LSTM	.802	.831	.772	
BERT(tuned+)	.534	.534	.533	
CNN	.662	.663	.661	
DenseNet(tuned)	.635	.641	.629	
CE	LSTM&CNN	.805	.834	.775
	BERT(tuned+)&CNN	.684	.689	.678
	LSTM&DenseNet(tuned)	.643	.681	.604
	BERT(tuned+)&DenseNet(tuned)	.678	.680	.675
MSE	LSTM&CNN	.838	.866	.809
	BERT(tuned+)&CNN	.699	.704	.693
	LSTM&DenseNet(tuned)	.726	.760	.691
	BERT(tuned+)&DenseNet(tuned)	.693	.698	.687

Next, we analyze the performance of representations that combine both image and text data in Table 4.3. We focus on studying the combination of the most effective image and text features, based on the results in Table 4.1. Specifically, we use LSTM trained with MSE, BERT(tuned+) with CE, CNN with Hinge loss, and DenseNet(tuned) with CE. We also focus only on MSE and CE due to the poor performance of the Hinge loss for the textual features. The results show that for most model combinations, using both features substantially outperforms the individual components. This finding supports the idea that image and text features are complementary and represent different aspects of the figure. Finally, we can see that the MSE loss is the best performing for all models and that the best

Table 4.4: Retrieval performance of the recommendation task. All differences with tf.idf are statistically significant.

	$p@3$	$p@5$	Same		Citing	
			$p@3$	$p@5$	$p@3$	$p@5$
tf.idf	.298	.228	.354	.276	.057	.048
LSTM	.044	.032	.058	.047	.014	.016
CNN	.000	.001	.001	.003	.001	.002
LSTM&CNN	.051	.039	.066	.054	.015	.014

Table 4.5: Figure recommendation performance. Statistically significant differences with tf.idf are marked with an asterisk.

	$p@3$	$p@5$	Same		Citing	
			$p@3$	$p@5$	$p@3$	$p@5$
tf.idf	.298	.228	.354	.276	.057	.048
Cross Entropy (CE)						
LSTM&CNN	.308	.241*	.368	.296*	.060	.056*
BERT(tuned+)&CNN	.296	.227	.352	.277	.056	.050
LSTM&DenseNet(tuned)	.303	.233	.355	.289*	.053	.056*
BERT(tuned+)&DenseNet(tuned)	.303	.229	.357	.279	.054	.050
Mean Squared Error (MSE)						
LSTM&CNN	.320*	.240*	.380*	.299*	.060	.059*
BERT(tuned+)&CNN	.296	.226	.353	.277	.057	.052
LSTM&DenseNet(tuned)	.313*	.235*	.371*	.287*	.058	.053
BERT(tuned+)&DenseNet(tuned)	.300	.229	.356	.278	.056	.049

performing model is the LSTM&CNN model.

Figure recommendation The goal of this task is to recommend figures to the user that are related to a target figure. To address this problem, we used a standard two-phase approach. First, using the target figure, an initial retrieval is performed to get an initial figure set. Then, a re-ranking model is used to obtain the recommended figures. To build a test set of target figures for testing, we first collected all figures whose article has at least five more figures and that have at least five figures in citing/cited papers (to result in $p@5 = 1$ at the best case). From this set, 500 figures were selected randomly (400 for testing and 100 for validation); all pairs of figures that contained at least one of the target figures were removed from the training set. The performance of the different models is measured using $p@3$ and $p@5$. Since there are no human relevance judgments available for the task, we assume that a figure is relevant if it appears in the same article as the target figure (“Same”), a citing

Table 4.6: Figure recommendation example.

LDA graphical representation	
tf.idf	Embeddings
1. The graphical representation of LDA	1. The graphical representation of LDA
2. Graphical models of LDA and DMM	2. Topic Model
3. Topic Model	3. Graphical Representation of strTM
4. Plate notation of our model: MATM	4. Plate notation of our model: MATM
5. LDA plate diagram	5. Graphical representation of (a) BTM, (b) Twitter-BTM

article (“Citing”), or in either (the main performance measure). We note that while this evaluation is not fully realistic, it can still help us make meaningful comparisons between the different approaches. Statistically significant differences between approaches were measured using the two-tailed paired t-test at a 95% confidence level.

First, we study the effectiveness of the retrieval step in Table 4.4. The performance of three embedding methods (which use text data, image data, and both), trained using the MSE loss, is compared with that of tf.idf. We can see that the tf.idf approach is the most successful. A possible reason for this is that tf.idf relies mainly on exact keyword matching between two texts, while embedding-based methods rely more on soft matching. Since we are searching over the entire collection, the embedding model may not be discriminative enough.

We report the performance of the recommendation task in Table 4.5. To obtain these results, we first perform retrieval using tf.idf and then rank the first 100 figures using the cosine similarity between the figure embeddings. We calculate the final score for a recommended figure using a linear interpolation between the tf.idf score and the embedding score. The weight for the tf.idf component and the embedding component in the interpolation is determined using a validation set (selected from $\{0.1, 0.2, \dots, 0.9\}$; we set the weights to sum up to 1). We experiment with embedding approaches that use both text and image data with the same setting as in Table 4.3. According to the results in Table 4.5, we can see that using embeddings on top of the initial retrieval results (tf.idf based) can substantially improve the recommendation performance. Specifically, the embedding approaches outperform the baseline in terms of the overall $p@3$ and $p@5$ for most relevant comparisons. Comparing the LSTM model with BERT, we can see that the former is better in most cases. The best embedding model, according to the results, is the LSTM&CNN model with the MSE loss.

An example target figure with its recommendation list is presented in Table 4.6. In the table, the captions of the target figures are presented together with the captions of five

recommended figures when using either tf.idf or embeddings (LSTM&CNN with MSE); in both cases, tf.idf was used for the initial retrieval. The subject of the first figure is the graphical representation of the LDA topic model. Using the tf.idf approach, we get figures that are either equivalent (e.g., “LDA plate diagram”), or diagrams of related models (e.g., “MATM” and “DMM”). When using the embedding approach, we can see that we get more diverse recommendations. This difference can be because using embeddings results in softer matching compared with tf.idf.

4.6 CONCLUSIONS

In this chapter, we studied the effectiveness of neural network-based figure embeddings. The experimental results showed that figure embeddings outperform the bag-of-words approach in the tasks of semantic similarity prediction and figure recommendation. We also observed that embeddings cannot replace the bag-of-words approach and that combining the two is the best practice. Finally, the results also showed that some learning approaches can be more effective than others. Specifically, using a simple model architecture and combining both image and text features performs the best.

In future work, different methods for combining the different figure features can be studied. Collecting user data to learn more effective representations and to improve the evaluation is another possible future direction.

The techniques proposed in Chapter 3 and Chapter 4 enabled us to obtain more effective representations of figures. They are general and can be immediately implemented in any application system as we will show later in Chapter 7.

CHAPTER 5: INTERACTIVE SUPPORT FOR QUERY CONSTRUCTION IN LITERATURE SEARCH ENGINES

In this chapter, we address the problem of query construction in literature search engines. Constructing effective queries in literature search can be a challenging task in some scenarios. For instance, when researchers perform an exploratory search process to learn about a new topic, they may not construct effective queries due to their lack of knowledge about it. To address this problem, we propose and study the effectiveness of an approach that leverages the collaboration between the user and the system for query construction. One of our main premises is that researchers will be willing to collaborate with the search engine to improve the search result. We perform experiments using a data set of publications on the topic of COVID-19 with a simulated user. The empirical results show that this approach can substantially improve poor-performing queries that would otherwise return no relevant articles and that those improvements are achievable with minimal user effort.

5.1 INTRODUCTION

Search engines generally work very well for popular queries. The reason for this is that the system can leverage large amounts of click-through information to train machine learning algorithms to optimize search results for those search queries [121, 122]. Such a strategy may fail for long-tail queries issued by only a small number of users. In those cases, search engines generally would have to rely mainly on matching the keywords in the search queries with those in documents. Unfortunately, such a method would not work well when the users' queries do not include the "right" keywords. The users in such cases would often end up repeatedly reformulating their search queries, yet they still could not find the relevant articles [5, 6, 123, 124]. Unfortunately, there are many such queries in literature search engines [124]. For example, when researchers perform an exploratory search process to learn about a new topic, they may find it challenging to construct effective search queries. Thus, how to improve the performance of those queries is a pressing challenge for literature search engines.

In this chapter, we address this problem and propose a general strategy for collaborative query construction. The main idea is to actively engage users in an iterative process to revise a query. The strategy attempts to optimize the collaboration between the user and the search engine based on the following assumptions:

1. *Ideal query*: For any poor-performing query, there exists an ideal query that, if con-

structured, would work well. This assumption generally holds because if we gradually increase the number of discriminative terms in the query, we would eventually push a relevant document to the very top in the ranked list of retrieval results. This assumption allows us to re-frame our problem as the problem of how to construct an ideal query.

2. *User-system collaboration*: User-system collaboration can be optimized by leveraging the strength of a search engine in “knowing” all the content in the collection and the strength of a user in recognizing a useful modification for the query among a set of candidates.
3. *User effort*: When the query is poor-performing, users would be willing to make some extra effort to collaborate with the search engine. This assumption is reasonable for literature search engines since retrieving all relevant articles for a query is crucial for successful research.

Our main idea is to optimize the user-system collaboration to perform a sequence of modifications to the query to reach an ideal one. While the proposed strategy includes multiple ways to edit the query, we initially focus on studying a specific editing operator where the system suggests terms to the user to add to the search query in each iteration based on the history of interactions of the user with the system.

To illustrate our proposed approach, we use a query example in Figure 5.1. In the example, we can see that the initial user’s query is poor performing, resulting in a result page with no relevant documents ($p@10 = 10$). Then, according to our approach, the user performs two modifications for the query (following two result lists) that substantially improve the effectiveness of the result list, resulting in $p@10 = 0.3$. The main idea of our work in this chapter is to support the user in the complete query construction process starting with an initial query and ending with a query that satisfies the user’s information need.

We evaluate our approach using a data set of research articles with a simulated user. The results demonstrate the great promise of this novel collaborative search support strategy for improving the accuracy of poor-performing queries with minimum effort from the user. The results also show that suggesting terms based on user interaction history improves effectiveness without additional user effort. We also perform experiments using a data set of news articles that demonstrate the potential applicability of this approach to other domains. Finally, we conduct a case study with three real users that show the potential effectiveness of the collaboration approach when real users are involved.

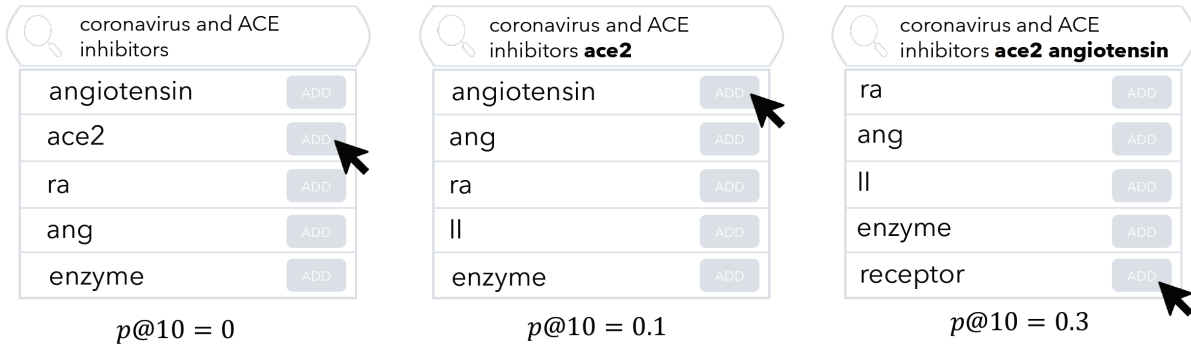


Figure 5.1: Illustration of our collaborative query construction approach.

5.2 COLLABORATIVE QUERY CONSTRUCTION (CQC)

5.2.1 General Framework

Our general idea for addressing the problem of poor-performing queries is to have a search engine collaborate with a user in constructing an ideal query based on the following hypothesis:

Ideal Query Hypothesis (IQH): For any information need of a user, there exists an ideal query that would allow a retrieval system to rank all the relevant documents above the non-relevant ones.

The IQH implies that if a user has perfect knowledge about the document collection, then the user would be able to formulate an ideal query. The IQH is reasonable because it is generally possible to uniquely identify a document by just using a few terms that occur together in it but not in others. This point was also referred to in previous work as the *perfect query paradox* [125]. We note that the IQH may not always hold (e.g., when there are duplicate documents). Nevertheless, it provides a sound conceptual basis for designing algorithms for supporting users in interactive search. Based on the IQH, the problem of optimizing retrieval accuracy can be reduced to the problem of finding the ideal query. Thus, based on this formulation, the main reason why a search task is difficult is that the user does not have enough knowledge to formulate the ideal query. In this thesis, we address this problem by helping a user to construct an ideal query.

Query construction approach Our collaborative query construction process is represented by a sequence of queries, Q_1, Q_2, \dots, Q_n , where Q_1 is the user's initial query, Q_n is

an ideal query, and Q_{i+1} is closer to Q_n than Q_i and the gap between Q_i and Q_{i+1} is small enough for the user to recognize the improvement of Q_{i+1} over Q_i .

From the system’s perspective, at any point in this process, the task is to suggest a set of candidate query terms. Given those candidates, the user’s task is to choose a set of query terms and possibly remove terms from the existing query. A single collaborative iteration of revising a query Q_i would be as follows:

1. Present the user a list of m candidate terms T_i (not in Q_i).
2. The user selects expansion terms: $E_i \subseteq T_i$.
3. The user selects terms to be removed from the query: $R_i \subseteq Q_i$.
4. $Q_{i+1} = \{Q_i \setminus R_i\} \cup E_i$.
5. Q_{i+1} is used to retrieve a result list D_{i+1} .

Query term suggestion framework Following the game-theoretic framework for interactive IR [126], our approach can be framed as the following Bayesian decision problem where the goal is to decide a candidate set of terms T_i to suggest to the user in response to the current query Q_i :

$$T_i = \arg \min_{T \subseteq V - Q_i} \int_{\Theta_Q} L(T, H_i, \Theta_Q, U) p(\Theta_Q | H_i, U) d\Theta_Q; \quad (5.1)$$

where:

- T_i is a candidate set of terms to be presented to the user from the vocabulary V .
- H_i is all the information from the history of interactions of the user with the system.
- Θ_Q is a unigram language model representing a potential ideal query.
- U denotes any relevant information about the user.
- $L(T, H_i, \Theta_Q, U)$ is a loss function assessing how good is T for H_i , U , and Θ_Q .
- $p(\Theta_Q | H_i, U)$ encodes the current belief about the ideal query.
- The integral indicates the uncertainty about the ideal query, which can be expected to be reduced as we collect more information from the user.

While we need to assess the loss of an entire candidate set T in the general case, in the much-simplified method that we will actually explore, we choose T by scoring each term and

then applying a threshold to control the number of terms. That is, we assume that the loss function on a term set T can be written as an aggregation of the loss on each individual term. As an additional simplification, we approximate the integral with the mode of the posterior probability about the ideal query, $\hat{\Theta}_Q$. Thus, our decision problem would become to compute the score of each term t , not already selected by the user, as follows: $s(t) = -L(t, H_i, \hat{\Theta}_Q, U)$; where $\hat{\Theta}_Q = \arg \max_{\Theta_Q} p(\Theta_Q | H_i, U)$. Computationally, the algorithm boils down to the following two steps: (1) Given all of the observed information H_i and U , compute $\hat{\Theta}_Q$. (2) Use $\hat{\Theta}_Q$ along with H_i and U to score each term in the vocabulary but not already in Q_i .

Framework implementation In this dissertation, as a first step in studying the proposed framework, we focus on a specific approach in which the query refinement is restricted to only adding one extra term to the query at each step, and terms are not allowed to be removed from the query by the user.

One advantage of using such an approach is that the gap between two adjacent queries is expected to be small enough for the user to recognize the correct choice. Furthermore, although this implementation strategy is very simple, theoretically speaking, the process can guarantee the construction of any ideal query that contains all the original query terms if the system can suggest additional terms in the ideal query but not in the original query and the user can recognize the terms to be included in the ideal query. We assume that the original query terms are all “essential” and should all be included in the ideal query. While true in general, in some cases this assumption may not hold, which would require the removal or substitution of terms in the initial query. In this thesis, however, we focus on term addition as our first strategy and leave the incorporation of other operations for future work.

5.2.2 Term Scoring Approach

According to the previous section, the optimal scoring function $s(t)$ is based on the negative loss $-L(t, H_i, \hat{\Theta}_Q, U)$. Intuitively, the loss of word t is negatively correlated with its probability according to $\hat{\Theta}_Q$. We thus simply define our scoring function as $s(t) = p(t | \hat{\Theta}_Q)$. That is, our problem is now reduced to infer $\hat{\Theta}_Q$ given all of the observed information H_i and U .

Next, we suggest a model for inferring $\hat{\Theta}_Q$, which is based on Pseudo-Relevance Feedback (PRF). This model is an extension of the relevance model RM1 [127] to incorporate H_i and is defined as follows:¹

$$p(t | \hat{\Theta}_Q) = \sum_{d \in D_i} p(t | d) \cdot p(d | Q_1, H_i). \quad (5.2)$$

¹We leave the incorporation of U for future work as such data is not available to us.

$p(t|d)$ is estimated using the maximum likelihood approach. We approximate $p(d|Q_1, H_i)$ using a linear interpolation:

$$p(d|Q_1, H_i) = (1 - \alpha) \cdot p(d|Q_1) + \alpha \cdot p(d|H_i); \quad (5.3)$$

$p(d|Q_1)$ is proportional to the reciprocal rank of d with respect to Q_1 ; $\alpha \in [0, 1]$. In order to estimate $p(d|H_i)$, two types of historical information are considered: (1) The terms selected by the user previously (H_i^T). (2) The result lists presented to the user previously (H_i^D). We combine these two components as follows:

$$p(d|H_i) = p(d|H_i^D) \cdot p(H_i^D|H_i) + p(d|H_i^T) \cdot p(H_i^T|H_i);^2 \quad (5.4)$$

To estimate $p(d|H_i^D)$, we assume that documents that appear in the result list presented to the user in the current iteration, and that were absent in the previous result list, represent aspects of the information need that are more important to the user. We thus estimate $p(d|H_i^D)$ as follows:

$$p(d|H_i^D) = \frac{1}{\text{rank}_{D_i}(d) \cdot Z_D} \quad \forall d \in D_i \setminus D_{i-1}; \quad (5.5)$$

$p(d|H_i^D) = 0$ for all other documents; $\text{rank}_{D_i}(d)$ is the rank of document d in the result list D_i ; Z_D is a normalization factor.

We estimate $p(d|H_i^T)$ such that high importance is attributed to documents in which terms that were previously selected by the user are prevalent.

$$p(d|H_i^T) = \sum_{j=1}^{i-1} p(d|t_j, H_i^T) \cdot p(t_j|H_i^T); \quad (5.6)$$

t_j is the term selected by the user in the j 'th iteration. $p(d|t_j, H_i^T)$ is set to be proportional to the score of d with respect to t_j as calculated by the system's ranking method. Assuming that terms selected in more recent iterations are more important than older ones, we estimate $p(t_j|H_i^T)$ as: $p(t_j|H_i^T) = \frac{\exp(-\mu \cdot (i-j))}{Z_T}$; Z_T is a normalization factor; μ is a free parameter that we set to 0.5 in our experiments.

To conclude, we assign a probability to each term which is a linear interpolation of its probabilities in the documents in the result list, where the interpolation weights are influenced by (1) the rank of the document, (2) the presence of the document in the previous list, and (3) the frequency of terms that were previously selected.

²We assume $p(H_i^D|H_i) = p(H_i^T|H_i)$ in the experiments.

5.2.3 Final Query Representation

According to our approach, the query Q_i is composed of the original query Q_1 and the terms selected by the user. The terms in Q_i are weighted based on a probability distribution such that the probability of a term t in V is:

$$p(t|Q_i) = \lambda_i \cdot p_{mle}(t|Q_1) + (1 - \lambda_i) \cdot p(t|H); \quad (5.7)$$

$p(t|H)$ is proportional to the weight that was assigned to the term by the scoring method if this term was previously selected, and is set to 0 otherwise; $p_{mle}(t|Q_1)$ is the maximum likelihood estimate of t in Q_1 ; $\lambda_i \in [0, 1]$.

5.3 RELATED WORK

The main novelty of the work reported in this chapter is the idea of collaborative construction of an ideal query, specific algorithms for iterative query expansion, and the study of their effectiveness for addressing poor-performing queries in literature search engines. In this section, we review the main research directions related to these contributions.

Interactive query expansion Previous works have studied approaches for interactive query expansion (e.g., [128, 129, 130, 131]). In one line of work [130, 132, 133], user studies were conducted to understand the extent to which users can effectively select expansion terms. The results of these studies showed that some user experience is required for selecting query terms effectively. An interactive approach for query expansion and reduction in the case of long queries was also studied [134]. Finally, other works have focused on the development of systems for interactive query expansion that leverage either explicit user feedback on documents [128, 135], or pseudo-relevance feedback [129].

According to these different works, the user needs to select terms to be added to each query independently. Our framework is more general both in performing a sequence of query modifications to optimize the user-system collaboration and in allowing potentially other query modifications than simply adding terms. Furthermore, we propose methods that suggest terms to the user based on the history of interactions of the user with the system.

Query suggestion On the surface, our approach is similar to query suggestion already studied in previous works [136]. However, there are two important differences: (1) The suggested queries in our approach are expected to form a sequence of queries incrementally converging to an ideal query whereas query suggestion is done for each query independently.

(2) The suggested queries in our method are composed of new terms extracted from the text collection, but the current methods for query suggestion tend to be able to only suggest queries taken from a search log.

Some works have focused on developing query suggestion approaches for long-tail queries [137, 138, 139, 140, 141, 142, 143]. For example, some works suggested to address the lack of click-through data for long-tail queries by considering skipped documents [137], leveraging anchor text [138], taking into account term associations [140], and using learning-to-rank algorithms [139]. In general, ideas from past works on query suggestion can be used in our approach for generating the set of query modifications that are suggested to the user in each revision step. Furthermore, these approaches can benefit from our framework by using information about user interactions history.

Automatic query reformulation There is a large body of work on devising approaches for automatic query reformulation (e.g., [121, 144, 145, 146]). According to these approaches, the query is automatically modified with no involvement of the user. One common method is to automatically add terms to the original query (a.k.a. query expansion) [144]. Other approaches also include, for example, substitution or deletion of terms (e.g., [121, 147]).

Automatic query reformulation approaches can differ in the sources of information that are taken into account (e.g., relevance feedback [148], external resources [149], and query logs [136]), as well as in the assumptions that are made regarding the effectiveness of terms. For example, while many works expand the query with terms that are semantically and topically related to the query (e.g., [127, 145, 150]), other works consider different criteria such as text coherency [121]. As in the case of query suggestion, ideas from automatic query reformulation can also be integrated into our collaborative approach.

Some works have used explicit user feedback on documents to effectively modify the query (e.g., [148, 151, 152]). In our work, we also utilize explicit user feedback, but on queries, thus requiring much less effort from the user.

Query Performance Prediction (QPP) Our approach is also related to the large body of work on query performance prediction [153, 154, 155, 156]. The goal of QPP is to predict the performance of a query in a given system using properties of the query, the result list, and the corpus. Thus, QPP can be used to improve the robustness of a system by identifying difficult queries and addressing them properly. Furthermore, QPP techniques can potentially be used to propose terms to the user to minimize the number of steps required to reach an ideal query according to our approach. For example, the information gain of the result list with respect to the corpus can be used to measure the effectiveness of terms [154]. Estimating

the deviation of the result list from the ideal one can be another possible approach worth exploring [153]. We plan to investigate the implementation of those ideas in our framework in future work.

5.4 EVALUATION

The evaluation of the proposed strategy has two main challenges: (1) The proposed approach is of interactive nature. (2) We are interested in focusing on difficult queries. We address these challenges by constructing new test collections based on existing collections that would focus on difficult queries and experimenting with simulated users.

5.4.1 Experimental Setup

Data sets To demonstrate the effectiveness of the collaborative query constriction approach in assisting researchers, we use the CORD-19 collection. This collection consists of 192,459 COVID-19 related research articles (we used the version of July 16, 2020). The 50 TREC-COVID track topics were used as queries (we used the “query” field). For the relevance judgments, we used the merged judgments of all five rounds of the competition.

While the approach we propose in this thesis can benefit researchers, it is general enough to be useful in any domain for addressing difficult queries. To show the generality of our approach, we also use a secondary collection of news-wire documents. Specifically, we use the ROBUST document collection (TREC discs 4 and 5- $\{CR\}$). The collection is composed of 528,155 news-wire documents, along with 249 TREC topics whose titles serve as queries (301-450, 601-700).

Building a test set of difficult queries In this dissertation, we define difficult queries as queries for which the first result page does not have any relevant documents. More specifically, we are interested in queries for which $p@10 = 0$ when using the BM25 model [98].

We use the following strategy to construct our test set. We perform retrieval for all queries and remove relevant documents at the top of the list from the collection. In the case of CORD-19, we consider the top 100 documents. In the case of ROBUST, we consider only the top 10 documents. We use more documents in the case of CORD-19 since it has substantially more relevant documents per query than ROBUST (around 500 vs. 70 relevant documents per query on average). Finally, we use for the evaluation only queries for which $p@10 = 0$ in the modified index. We also remove queries for which there are less than 10 relevant documents remaining in the modified collection (to obtain an upper-bound value

of $p@10 = 1$ for all queries). After this process, we remain with 45 queries in the case of **CORD-19** and 71 queries in the case of **ROBUST**.

Evaluation measures We report the performance in terms of precision ($p@ \{5, 10\}$) and Normalized Discounted Cumulative Gain ($ndcg@ \{5, 10\}$). We also report the fraction of queries for which a method resulted in $p@10 > 0$, denoted $success@10$. The two-tailed paired t-test at 95% confidence level is used to determine significant differences in performance between different approaches.

Implementation details Stopword removal and Porter stemming were applied to both documents and queries. The Anserini toolkit was used for experiments.³ The BM25 model was used for ranking [98]. Our approach involves various free parameters that were set using leave-one-out cross-validation; we use this approach due to the small number of test queries. We select the number of document used for pseudo-relevance feedback from $\{10, 50\}$. The interpolation parameter in Equation 5.3, α , which controls the importance of user history in the model, was selected from $\{0.0, 0.2, 0.5, 0.8, 1.0\}$. The value of λ_i , the weight that is given to the original query, is set to $\max(\gamma, \frac{|Q_1|}{|Q_i|})$; we chose this weighting function as to attribute high importance to the original query when a small amount of expansion is used; γ was selected from $\{0.2, 0.5\}$. The number of terms suggested to the user, m , was set to 5.

Baselines We compare the performance of our approach with that of using the original query (denoted **BM25**), and of using an automatic query expansion approach in which a set of terms is automatically added to the original query once. We use the **RM3** [157] expansion model and set its hyper-parameters using leave-one-out cross-validation. The number of pseudo-relevant documents was selected from $\{10, 50\}$, the weight for the original query was selected from $\{0.2, 0.5\}$, and the number of expansion terms was set to 10.

Simulated user To perform a controlled study of our approach, we experiment with a simulated user. Given a list of term suggestions, the simulated user chooses a term with the highest *tf.idf* score in the relevant documents for the query. Specifically, for each query, we concatenate all relevant documents and compute *tf.idf* based on the single concatenated “relevant document”.

5.4.2 Experimental Results

Main result The main results of the experiments are summarized in Table 5.1. In the table, the performance of the collaborative query construction approach is reported as a

³<https://github.com/castorini/anserini> (accessed August 25, 2021)

function of the number of terms added to the query (in $\{1, 2, \dots, 5\}$). According to the results, for both data sets, after adding a *single* term to the query, users are able to see a noticeable improvement on the first page of search results in more than 50% of these difficult queries that did not return any relevant document initially (*success@10*). The results also show that it is almost always beneficial to add more than one term to the query. Specifically, for the majority of evaluation measures, adding more terms always improves the approach performance. Focusing on RM3, we can see that it is outperformed by CQC in both data sets for the majority of relevant comparisons. Specifically, using at least two terms in CQC outperforms RM3 significantly. This result demonstrates the effectiveness of CQC when even a very small number of terms is used, thus requiring minimal effort from the user.

In Figure 5.2, we further analyze the performance of CQC compared with that of the automatic query expansion approach RM3. Specifically, at each iteration of query revision, we simply use the query generated by RM3, without any user involvement. The results further demonstrate the advantage of leveraging user interactions, both in the term suggestion algorithm and by involving the user in the revision process, compared with an automatic query expansion approach. Specifically, for all revision iterations, CQC outperforms RM3 in terms of *p@10* and *success@10*. Furthermore, we can see that the performance of RM3 for ROBUST is much lower than in the case of CORD-19. This result illustrates the sensitivity of RM3 (which only uses the top retrieved documents to extract expansion terms) to the data set at hand. Our CQC approach, on the other hand, is less sensitive to that since it leverages user interactions.

Model components analysis Our term scoring method utilizes both the original query and the user interaction history. In the following analysis, we are interested in examining the relative importance of these individual components. Setting $\alpha = 0$ in Equation 5.3 results in a model that uses only the original query (CQC-Q). Setting $\alpha = 1$, on the other hand, results in a model that uses only user history (CQC-H). The results of this analysis are presented in Figure 5.3.

Focusing on *p@10*, we can see that all components are very effective for both data sets. Comparing the different components, we can see that, in the general case, incorporating user history in the model is mostly beneficial in later iterations. Specifically, in the case of CORD-19, CQC-H outperforms CQC-Q for the fourth and fifth terms. In the case of ROBUST, we can see that while CQC-H does not improve over CQC-Q, combining the two outperforms the individual components in the later iterations. Focusing on *success@10*, we can see that by combining the two components of the model, the resultant approach is of higher robustness. Specifically, we can see that while the individual components can be

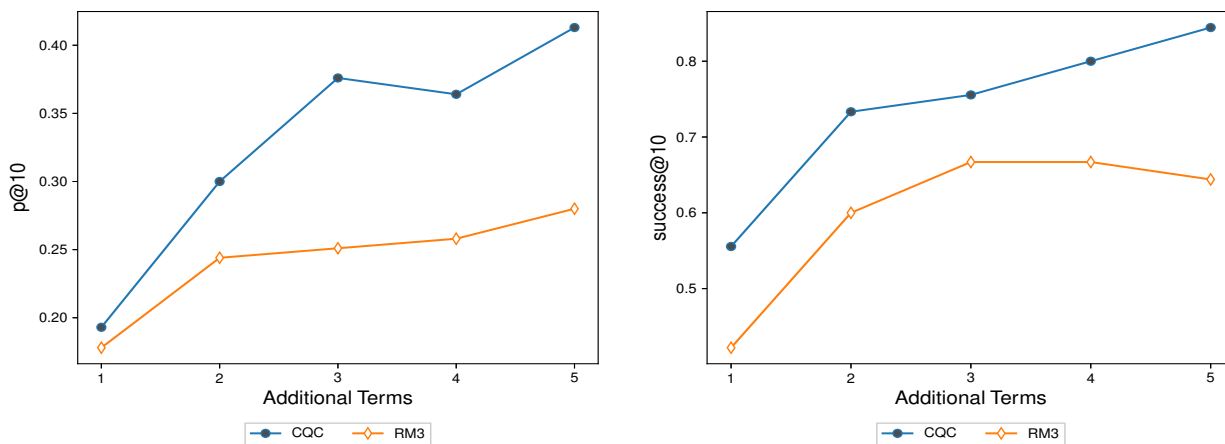
Table 5.1: Main Result. The performance of the collaborative query construction approach for a simulated user. Statistically significant differences with RM3 (*ndcg* and *p*) are marked with ‘*’.

CORD-19					
	<i>p</i> @5	<i>p</i> @10	<i>ndcg</i> @5	<i>ndcg</i> @10	<i>success</i> @10
BM25	.000	.000	.000	.000	.000
RM3	.196	.178	.166	.155	.422
Collaborative Query Construction (CQC)					
1 Term	.173	.193	.138	.151	.556
2 Terms	.302*	.300*	.239	.240*	.733
3 Terms	.356*	.376*	.275*	.297*	.756
4 Terms	.347*	.364*	.275*	.289*	.800
5 Terms	.404*	.413*	.328*	.338*	.844
ROBUST					
	<i>p</i> @5	<i>p</i> @10	<i>ndcg</i> @5	<i>ndcg</i> @10	<i>success</i> @10
BM25	.000	.000	.000	.000	.000
RM3	.051	.051	.039	.045	.296
Collaborative Query Construction (CQC)					
1 Term	.107*	.123*	.092*	.108*	.563
2 Terms	.172*	.187*	.156*	.170*	.606
3 Terms	.186*	.192*	.167*	.174*	.606
4 Terms	.237*	.227*	.218*	.214*	.606
5 Terms	.262*	.246*	.244*	.235*	.648

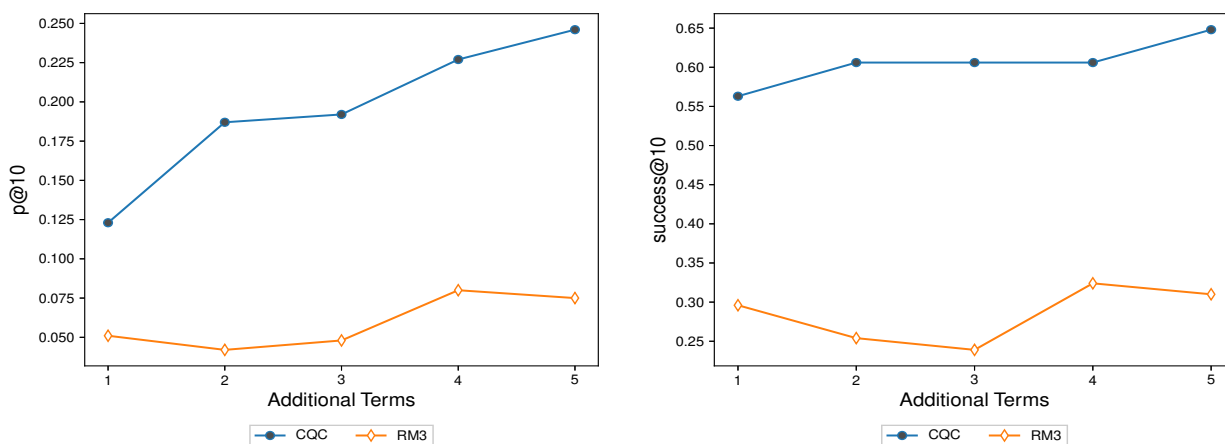
sensitive to the number of terms, their combination (CQC) is monotonically increasing.

User model analysis In the previous experiments, we assumed that an “ideal” user is interacting with the system. That is, given a list of terms, the user would select an expansion term that has the highest *tf.idf* score in the relevant documents of the query. In the following analysis, we would release this assumption and study the system performance when noisy users are involved in the interactive process. It is interesting to perform this analysis since users might deviate from the “ideal” term selection strategy in realistic application scenarios. For instance, this could be the case when users are not very familiar with the query topic. Thus, we are interested in examining the sensitivity of our approach to such changes.

To study that, we assume a probabilistic user model as follows. Given m terms $\{t_1, t_2, \dots, t_m\}$ that are shown to the user, the probability that a user would select a term t_i is set to $\frac{\exp(\gamma \cdot tf.idf_i)}{\sum_{j=1}^m \exp(\gamma \cdot tf.idf_j)}$; $tf.idf_i$ is the *tf.idf* score of the term t_i in the relevant documents. We set the value of $\gamma \in \{0.0, 0.0001, 0.001, 0.01, 0.1\}$ to control the level of user noise. Specifically,



(a) CORD-19

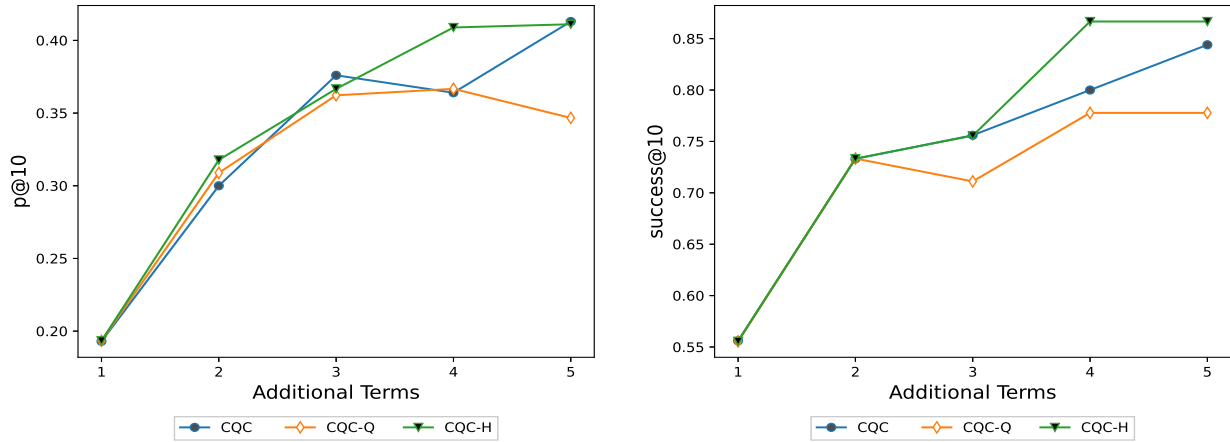


(b) ROBUST

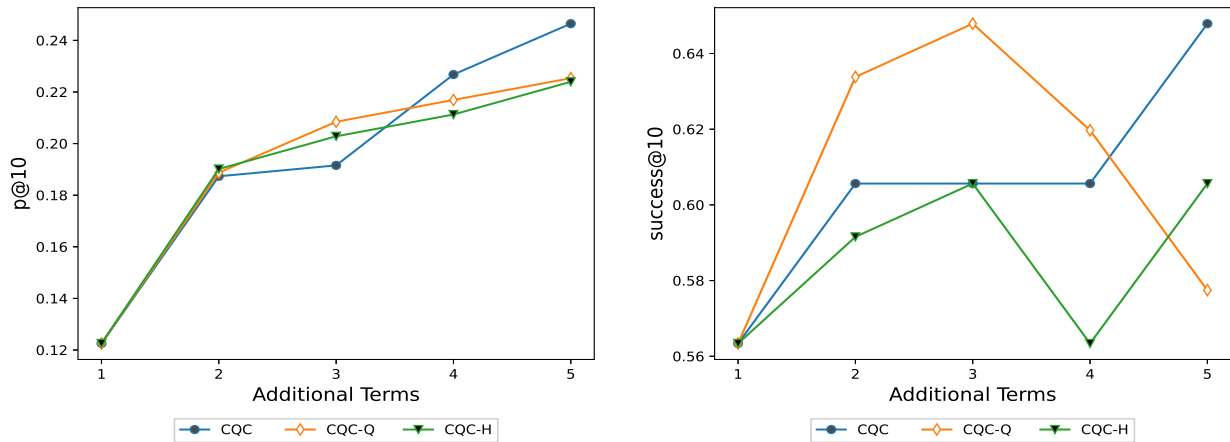
Figure 5.2: The performance of using RM3 at each iteration.

setting $\gamma = 0$ results in uniform sampling and a sufficiently high value of γ would result in the “ideal” term selection strategy; according to our experiments, this value is 0.1. The results of this analysis are presented in Figure 5.4.

The results show that the collaborative approach can be somewhat sensitive to user noise when selecting expansion terms. Specifically, it is often the case where decreasing the value of γ would result in lower performance for any number of words. Still, the results show that for all levels of user noise, there is a substantial performance benefit (compared with the initial list) when using the collaborative approach. According to the graphs (*success@10*), for all levels of user noise, there is a noticeable improvement (w.r.t. the initial result list) for at least 30% of the queries when adding a single term and for at least 50% of the queries when adding five terms. The results also show that the optimal number of terms that should



(a) CORD-19

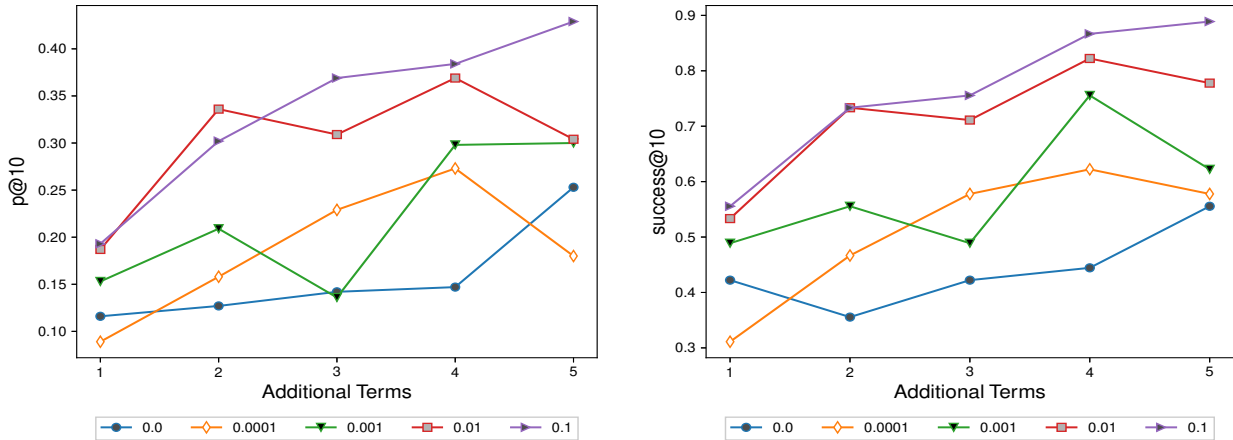


(b) ROBUST

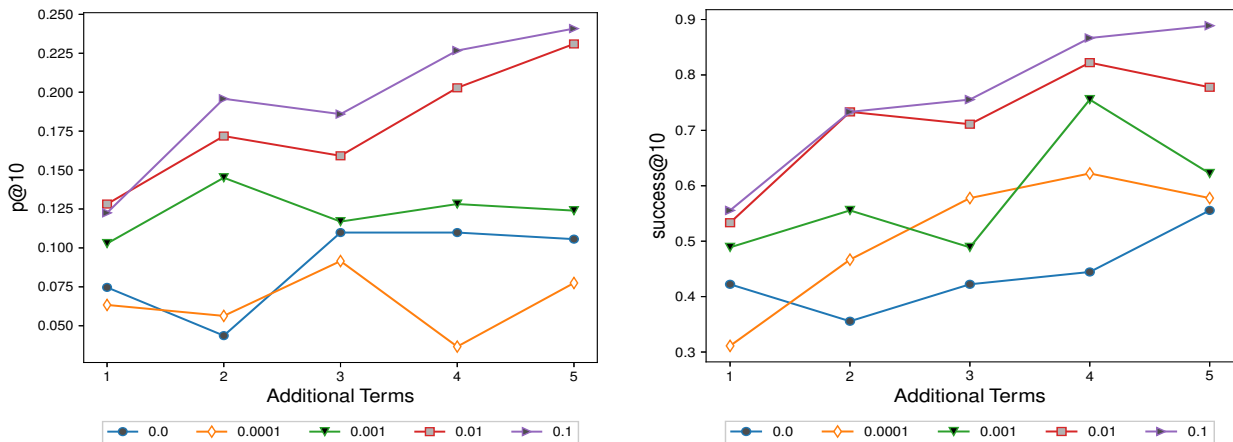
Figure 5.3: The performance of different model components.

be added to the query depends on user noise. While for low levels of noise (e.g., 0.1, and 0.01) the curves are usually increasing as a function of the number of terms, adding extra terms can decrease the performance when the level of noise is high in some cases. A possible way to address this issue is by having the option to remove terms from the query which is a possible extension of our framework that we plan to investigate in future work.

Query difficulty analysis In the following, we examine the performance of two groups of queries with similar initial performance. To perform the analysis, we use the original collection of documents to reach different levels of initial performance (in the modified collection, the initial performance for all queries in terms of $p@10$ is 0). To create the two groups, we sort the queries based on $p@10$ of the initial result list; if two queries have the same value,



(a) CORD-19



(b) ROBUST

Figure 5.4: The performance of a simulated user with different levels of noise.

we determine their order based on a numeric identifier. We then split the queries into two sets based on that sorting: (1) LOW: queries with low initial performance, and (2) HIGH: queries with high initial performance. The results are presented in Figure 5.5 for both data sets.

According to the results, we can see that queries with a relatively low performance benefit more from the collaborative approach than queries with higher performance. Focusing on ROBUST, we can see that adding more terms almost always helps to improve the performance in both query groups. Still, when comparing HIGH with LOW, we observe that the percentage of improvement (over not using term expansions) is higher in the case of LOW. In the case of CORD-19, on the other hand, we can see that in some cases adding terms can decrease the performance in both query groups. A possible explanation for this can be

the initial high performance of queries in this data set compared with ROBUST. Still, in both data sets and query groups, there exist a number of terms for which the collaborative approach can improve over the initial list.

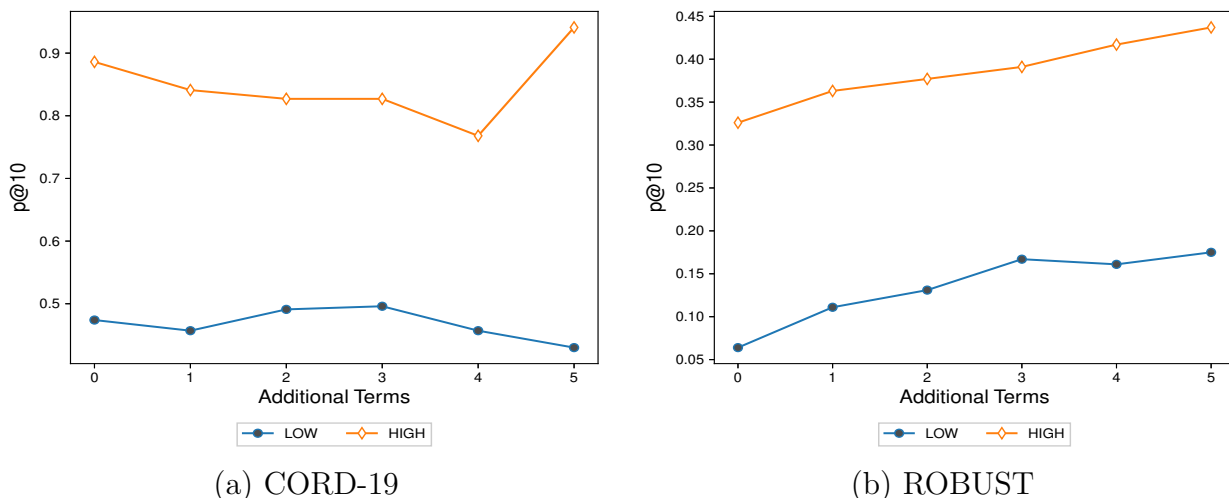


Figure 5.5: The performance of two groups of queries with similar $p@10$ of their initial result list. The queries with initial low and high performance are denoted LOW and HIGH, respectively.

Query characteristics In Table 5.2, we report different characteristics of queries for which the collaborative approach either failed or succeeded. To obtain those characteristics, we experiment with an Oracle-based collaborative approach that suggests to the user, at each iteration, a list of terms from the relevant documents (based on $tf.idf$). We chose this approach since we are interested in examining the characteristics of queries that can succeed/fail when using our framework regardless of the specific method used for term selection. To create the failing (successful) queries group, we select queries for which $p@10 = 0$ ($p@10 > 0.3$) after adding a single expansion term; we report the number of queries in each group in Table 5.2. For each query group, we use the top-10 documents based on the original query to calculate the following statistics: (1) Query Coverage: the portion of documents from the result list that contain all query terms. (2) Single Term Coverage: the portion of documents from the result list that contain a single query term (an average over query terms). (3) Number of Terms: the number of query terms. (4) idf : the average idf of the query terms. (5) tf : the average of query terms frequency in the result list.

The results in Table 5.2 shed some light on the differences between queries that succeed and fail when using the collaboration-based strategy. First, the coverage of query terms in

the result list is generally higher for successful queries as attested by Query Coverage, Single Term Coverage, and tf . This is an interesting finding since the term suggestions displayed to the simulated user in those experiments were not selected from the top-ranked documents but from the relevant ones only (which are not top-ranked since we ensured that $p@10 = 0$). This finding can be explained by the potential risk of adding a single term to the query. Good coverage of the query in the result list might suggest that the query is formulated relatively well thus reducing the risk of adding a single term. This is further supported by the difference in idf between the query groups which is known to be an indicator for query performance [156]. Finally, we can see that the number of terms in successful queries is lower than in failing queries. This finding motivates using our approach for short queries that can potentially benefit from some additional terms.

Table 5.2: Different characteristics of queries for which the collaborative approach resulted in either success or failure (in terms of $p@10$ w.r.t. the initial result list). In parenthesis: the number of queries in each group.

	CORD-19		ROBUST	
	Failure (18)	Success (16)	Failure (14)	Success (19)
Query Coverage	.617	.744	.736	.837
Single Term Coverage	.875	.889	.886	.938
Number of Terms	3.4	2.8	2.7	2.6
idf	101	130	352	381
tf	30	36	87	96

5.4.3 Case Study with Real Users

We are interested in examining whether real users can recognize the “good” terms suggested by the system. To gain some initial understanding regarding this issue, we conducted a case study with three real users. We note that the conclusions that can be drawn from this study are limited due to the small number of users. Yet, this study is still useful for getting some intuition regarding the utility of the approach. We use the ROBUST data set for this experiment. The reason for this is that this data set is of general news-wire documents which makes it easier to find actual users who could make sense of it. This is in contrast to the CORD-19 data set that requires expertise in the topic.

Each participant performed three iterations of the collaborative process for 30 queries. Specifically, we selected queries that achieved the highest performance in terms of $p@10$ after adding a single term by the simulated user. We chose these queries as we are interested

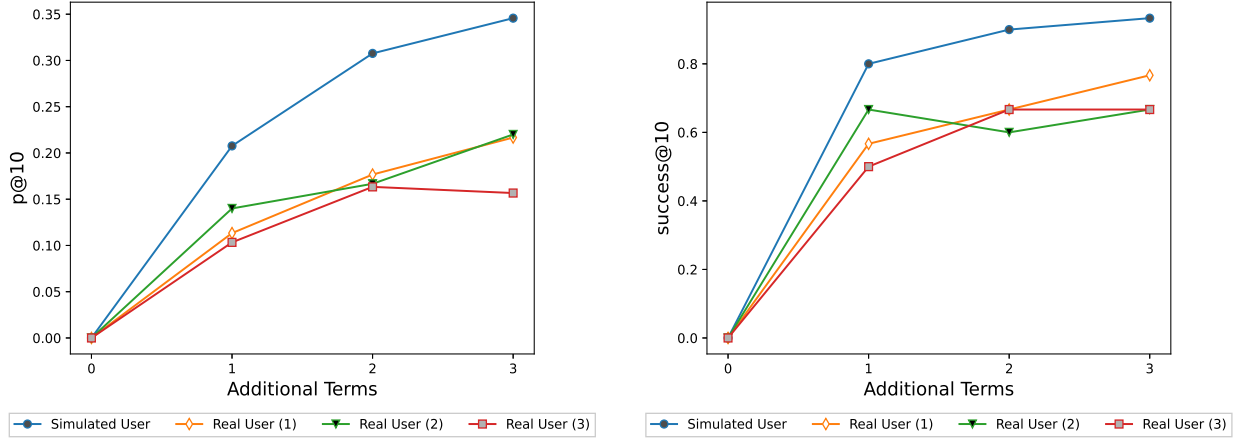


Figure 5.6: The performance of real users vs. a simulated user.

Table 5.3: Query Examples. The performance of the query ($p@10$) after adding a term is reported in the brackets.

curbing population growth	Real User	plan (0.0)	family (0.2)	birth (0.6)
	Simulated	china (0.1)	economic (0.1)	rate (0.2)
Stirling engine	Real User	company (0.0)	financial (0.0)	group (0.0)
	Simulated	cfc (0.9)	hcfc (1.0)	hyph (1.0)
antibiotics ineffectiveness	Real User	infection (0.2)	research (0.2)	study (0.2)
	Simulated	drug (0.1)	pharmaceutical (0.2)	product (0.1)

to study the following research question: given a term scoring method that can provide effective terms, can the user identify them? For each query, the user was presented with the initial query, a text describing the topic, and the guidelines regarding how a relevant document should look (all are part of the TREC topics). After issuing a query, the users are presented with a result list of 10 documents (a title and a short summary of 5 sentences are presented).

In Figure 5.6, we compare the performance of the real users with that of the simulated user. According to the results, retrieval performance can be very good when terms are selected by real users. Specifically, all users reach $success@10$ of around 0.5. That is, after adding a single term, at least one relevant result is obtained for about 50% of the queries.

In Table 5.3, we present examples of queries along with the terms that were selected by a single real user and a simulated user. We also report the performance that resulted from adding a term. The first query serves as an example where the real user outperforms the simulated user by a better choice of terms. The second query is an example where the simulated user outperforms the real user presumably by recognizing the correct technical

terms. Finally, the last query is an example where both users achieve similar performance but using different terms.

5.5 CONCLUSIONS

How to improve retrieval accuracy for poor-performing long-tail queries is a pressing challenge in the optimization of literature search engines. We proposed and studied a novel strategy for improving the accuracy of difficult queries by having the search engine and the user collaboratively expand the original query to incrementally approach an ideal query.

Evaluation with simulated users on two test collections with difficult queries showed great promise of this strategy. Furthermore, the results showed that leveraging the history of interactions of the user with the system can improve the effectiveness of the term suggestion algorithm. Specifically, the results showed that for more than 50% of the queries, adding just a single expansion term using our collaborative approach improves the initial result page that did not contain any relevant information initially. The results also showed that the method is quite robust to user noise when selecting expansion terms. Finally, a case study with three real users showed the system’s potential effectiveness in a realistic setting.

The strategy proposed in this chapter is general and can be implemented in any search engine as an option to improve the accuracy of poor-performing queries in the form of a “Help Me Search” button, which the users can click on as needed. To demonstrate that, in Chapter 7, we propose an implementation of this approach in an actual system for literature search.

There are several possible directions for future work. One possible research direction is to develop more approaches for term scoring which can potentially leverage semantic information (such as word embeddings) or techniques from query performance prediction. Another research direction for future work can be to conduct a user study to examine the benefit of this approach with users. Finally, we plan to study the framework’s effectiveness when more than one term can be added to the query and when term subtraction is allowed.

CHAPTER 6: AUTOMATIC ASSESSMENT OF RESEARCH ARTICLES

In this chapter, we study the problem of automatic quality assessment of research articles. Specifically, we focus on the task of predicting the performance of an article in different research aspects such as originality, clarity, and impact. Automating this task can be useful for researchers in many ways. First, it can speed up and improve the quality of the academic review process through automatic labeling of articles and provide early feedback for the article’s authors. Second, it can help improve the current literature systems by enriching articles with additional side information. This information can then be used to develop novel functions to facilitate the analysis of collections of research articles.

We explore the use of topic model-based features for the task. Specifically, we propose and study different approaches to generate topic models using the collection of research articles. We also study the effectiveness of different approaches to combine different types of topic model features as well as topic model features with bag-of-words features. Our experiments, using two data sets of research articles in two domains, demonstrate the effectiveness of our proposed approach. We conclude this chapter with an empirical study of a possible application of using the predicted scores to improve the ranking of literature search engines.

6.1 INTRODUCTION

Assessment is an essential part of scientific research as it is needed to make sure that published articles are of high quality and adhere to strict scientific principles. Traditionally, assessment has been done manually by several peer-reviewers who are researchers with experience in the research area of the paper. The reviewers would judge the quality of the work according to different criteria to come with a recommendation on whether the article is ready for publication. For instance, reviewers would assess the impact of findings, the soundness of the methods, and the clarity of presentation. Thus, a review for an article would usually be composed of scores for different aspects and text explaining the reviewer’s decision of those scores. The manual assessment of research articles is thus a labor-intensive task. Furthermore, the reviews (scores and text) are often not publicly available where one of the reasons for this is concerns regarding the anonymity of the reviewers.

In this chapter, we study the automatic assessment of research articles and are specifically interested in predicting the scores of an article in the different aspects of an academic review. The automatic generation of review scores for articles can benefit scientific research in many ways. First, it can accelerate and improve the quality of the publication process by auto-

matically generating review scores to support the reviewing process. By doing that, it can also help to address the biases of the different reviewers, which can make the outcomes of a review process unreliable in some cases [158, 159]. In addition to reducing human effort, the automated assessment also has the potential to leverage data mining and machine learning to provide detailed feedback to authors at an early stage to help them improve their work.

Automatically generated review scores also have the potential of improving the current literature search systems. Specifically, most systems nowadays usually do not use content analysis for assessing the quality of articles and rely on citation counts instead. Review scores can thus enrich the article’s representation, which would help to improve those systems. Specifically, the review scores in different aspects can be used to enhance the system’s ranking by adding more features to the representation of articles. Furthermore, sorting and filtering based on a review dimension can reveal works with strength in specific aspects to improve the exploration of collections of research articles. For example, ranking articles based on their clarity, users can find papers on a topic that are easier to read than others for researchers that are new to a specific field. The scores can also assist the user to assess the quality of newly published articles with low citation counts and papers in pre-print repositories that are gaining popularity in recent years [66]. For example, ranking based on the aspect of originality, the user can find newly published works with the potential to be the new state-of-the-art or that define new interesting problems to study. Finally, since reviews are usually not publicly available, presenting the user automatically generated reviews can help them understand the strengths and weaknesses of a paper.

Although automatic assessment of research articles is an important problem to study, there have not been many works on the topic. The main reason for this is that due to privacy and copyright considerations, until recently, there were no publicly available data sets for the task [10]. Furthermore, the typical data sets of research articles are usually not very large, which poses challenges for using machine learning algorithms for the task. For those reasons, the current reviewing process and literature systems do not include almost any form of automated assessment. Finally, to accurately assess an article based on different aspects, such as novelty, clarity, and impact, it is usually required to have substantial expertise in the specific research area. Thus, using the article’s text to generate review aspect scores automatically is a challenging task.

We focus on a specific assessment problem in which the goal is to predict the scores of different aspects of a review for a scientific article using only its text. Some previous studies have proposed to apply machine learning to automate the assessment of research articles [10, 48, 70, 71, 72], where the authors have demonstrated the feasibility of leveraging supervised learning to automate the assessment of research articles. These works have mainly

used deep neural networks to learn textual features based on the article’s text. In this thesis, we explore using topics as features for the automated assessment of research articles. Our motivation in using topic models is based on the premise that well-performing features for the task should be able to capture the complex semantics of a research article. For example, a word can have different meanings in different articles (polysemy), and multiple words within an article can share the same meaning (synonymy). Furthermore, there can be complex ideas in an article, which may require many words to describe. Topic models can effectively address those issues. These models represent a topic as a distribution over the words in a vocabulary with the high probabilities assigned to the most important ones in characterizing it. Topical features are potentially advantageous because of their ability to capture semantics via the clustering of words. For instance, a topic represented as a word distribution would address the problem of polysemy by allowing a word to have non-zero probabilities for multiple topics and the problem of synonymy by involving all the synonyms of a word in the same topic representation with non-zero probabilities. Moreover, because of the use of potentially all the words in the vocabulary and the flexibility in assigning different weights (i.e., probabilities) to them, the topic representation can help distinguish subtle differences between articles.

Compared with techniques such as bag-of-words and neural networks, using topic models for the task has two main advantages. First, topic model features can be more interpretable than other approaches. For example, topical features can potentially explain a given score better than bag-of-words or neural network features, which are less interpretable. Furthermore, to learn features only from the training data at hand using deep neural networks, large amounts of data are needed that are not available for some tasks in the research domain. An alternative approach studied in recent years is to pre-train neural network models using a general corpus of research articles and then fine-tune the model using the specific target task [54]. The process, however, might be challenging when the training set for the downstream task is very small as in the case of the data set we experimented with since we deal with a relatively new task.

Topic models, on the other hand, can be learned using the text data with unsupervised probabilistic models such as the Probabilistic Latent Semantic Analysis model (PLSA) [160] and the Latent Dirichlet Allocation model (LDA) [103]. Topic models have already been used successfully in various text mining applications. For example, they were shown to be effective for prediction of time series variables [161], information retrieval [162], and text analysis [163]. This work can be considered a novel application of topic models for the automated assessment of research articles.

We propose and study multiple ways to extract topics and apply topical features to au-

tomate the assessment of research articles. Specifically, we propose to generate different kinds of topics using multiple views of the text data, including: (1) Topics learned in an *unsupervised* manner, using all available data. (2) Topics learned using the *guidance of review aspect scores* in a training set: using only the high scoring articles may result in topics that capture good practices, while using low scoring ones may result in topics that capture common drawbacks. (3) Topics learned using *different granularities* of text segments which can capture different levels of semantic meaning.

We also study the combination of topical features with bag-of-words features as these provide complementary perspectives of the research articles. For this combination, and the combination of different topical features, we study two approaches as follows: (1) Combining the predictions of the models learned for each group of features separately. (2) Pooling all features together to form a larger set and learning a single prediction model.

To evaluate the proposed methods, we used two data sets in different domains. One data set is in the veterinary medicine domain, and the other data set is in the computer science domain. For both data sets, the goal is to predict the scores for each article in multiple review aspects. Our experimental results show that topical features are interpretable and highly effective for the task, outperforming the bag-of-words approach and neural networks in most review aspects. Further analysis also demonstrates the effectiveness of using multiple views of the text data to learn the different models and combining them using the prediction scores. The results also show that combining topical features with bag-of-words features can substantially outperform the individual components but only for some of the review aspects. Furthermore, combining the prediction results of using bag-of-words features and topical features is more effective than pooling the features. Finally, we perform an empirical study that demonstrates the effectiveness of using the predicted scores to improve literature search systems.

6.2 RELATED WORK

The main novelty of the work described in this chapter is the development and study of topic model features for the automated assessment of research articles. This thesis work is also the first to study the effectiveness of using review aspect scores for the rankings of scholarly search engines. In this section, we review three lines of work that are most related to our contributions.

6.2.1 Review Aspect Score Prediction

The task of review aspect score prediction was introduced first by Kang et al. that also

released a data set for the task [10]. In that work, they also implemented different baselines based on neural networks. Specifically, the different approaches implemented in that paper used a pre-trained word embedding layer (e.g., Word2Vec [99]) whose output served as an input to a neural network with the CNN or LSTM architecture. Finally, the output of the model served as a score for a specific aspect. In this thesis, we study the effectiveness of using probabilistic topic model-based features that, similar to neural network-based features, serve as a semantic representation of articles. Using topics is advantageous since they can be learned in an unsupervised manner without access to massive amounts of data. Furthermore, topic model features are more interpretable than neural network-based features. In the experimental section, we compare the performance of topic models with that of some representative neural network architectures.

Qiao et al. studied an approach that takes into account the structure of a research article by combining the different article sections using an LSTM architecture with an attention mechanism [70]. Their results, however, were inconclusive and the improvements in various review aspects were only marginal. In our work in this chapter, we study the effectiveness of different text representations for the task which can be leveraged in the future in models that rely on the article’s structure. Another related work [48] used knowledge graphs to predict review aspect scores and to generate textual reviews. In this thesis, we focus on studying the effectiveness of using the article’s text as the source of information without relying on external resources such as knowledge graphs. We thus leave the combination of knowledge graph features for future work.

6.2.2 Related Tasks

Some previous works have focused on the related task of predicting acceptance of research articles using the article’s text. One work [72] proposed to use different hand-crafted features to address the problem. Another work [73] used a hierarchical attention network by taking into account the structure of the article. Finally, another work used only visual features for the task [164]. In this thesis, we focus on the task of multi-aspect review score prediction that is more challenging than the problem of predicting article acceptance, which is essentially a binary classification problem.

In another line of work related to ours, the text of reviews was used to predict the review outcomes. In several works in this direction [67, 69, 165], sentiment analysis was used for the task. Other works used mostly textual features [68, 166]. In this thesis, we focus on using only the article’s text for the task.

A few works have studied the task of automatically generating the text of reviews, which

is also related to our work in this thesis. For example, techniques such text summarization [167] and knowledge graphs [48] were studied for the task. In this thesis, we focus on quality score prediction and leave the generation of review text for future work.

Finally, there has been some previous work on automatic quality assessment in other domains as well. For example, one paper focused on the quality assessment of Wikipedia documents using multi-modal analysis [63]. The task of automatic assessment was also studied for the tasks of code reviewing [168] and simple essay scoring [169]. Finally, one work addressed the problem of automatic assessment of complex assignments, which can be considered as similar to research articles [65].

6.2.3 Topic Models

Topic models have frequently been used for classification tasks since their inception [103]. The most traditional approach is to use a topic model to infer topics on a set of training documents and then at classification time use the model to infer topic proportion vectors for the unseen documents, which are then used as an input to a classifier [170]. Such approach, for example, was used for forecasting of time series variables [161], image ranking [51], and citation recommendation [22].

Another approach is to integrate the topic modeling with the classification task into one unified framework [171, 172] where both the topics and the labels are modeled directly through an augmented LDA/PLSA topic model. Then, topics are learned across the entire corpus at once, including documents from all labels to be predicted. By contrast, the approach we propose in this thesis generates an independent set of topics for each label, which can then be combined into a larger feature vector. Furthermore, compared to previous works in which specialized topic models with supervision were developed, our approach is completely general and can thus be combined with any existing topic models to achieve the effect of supervision.

Topic models have also been modified to more directly support other specific tasks. One example is the author-topic model [173], which attempts to model topical preferences within documents as well as among individual authors of documents. Topic models have also been modified to predict urban activity patterns [174] by adapting LDA to geo-tagged activity data. In these cases, the underlying graphical model itself is adapted to address the new task, which often necessitates the derivation of a new sampling algorithm. In this thesis, we instead focus on approaches that can leverage *existing* topic models more optimally without directly changing the underlying graphical model itself. In this sense, our work is completely orthogonal to the existing work on topic models.

6.3 TOPIC DISCOVERY AND CONSTRUCTION OF TOPICAL FEATURES

The approach we explore in this thesis is to use topic models to learn topics and construct features based on these, which we would use in a supervised machine learning framework to predict review aspect scores. In this section, we describe the technical approaches in detail, starting with an introduction to topic models.

6.3.1 Topic Models Background

A topic model is a probabilistic generative model for text data. The underlying premise is that text in a document originates from a mixture of several topics, which represent different themes. In this thesis, we use the LDA model [103] to learn topics. LDA can be applied to any set of documents to learn k topics, where each topic is a multinomial distribution over the vocabulary words, $\theta_j \forall j \in \{1, 2, \dots, k\}$. For example, in a topic model built using a collection of news articles, a topic about sports is expected to attribute high probabilities to words such as *football*, *basketball*, and *tennis*, but very small probabilities to words like *congress*, *party*, and *bill*, which may have high probabilities in a topic about politics. Furthermore, each document in the training set is assigned with a multinomial distribution over the k topics, π_d , where $\pi_{d,j}$ is the probability that a word within the document d was drawn from topic j . The generative process according to LDA goes as follows: (1) Sample a multinomial distribution over topics, π_d , from a Dirichlet prior distribution. (2) For each position in the document, select a topic j by sampling from π_d . (3) Sample a word according to θ_j .

6.3.2 Aspect-guided Topic Modeling

Our goal is to use topic models to generate features and use these features for the automatic assessment of research articles. Our premise is that topics represent textual patterns in research articles, which correlate with performance in the different aspects used for reviewing. The standard approach for learning topics would be to use all available articles (from both the training and the test set) in a fully unsupervised manner. This type of model would benefit from using the maximum amount of data. We refer to this model as **StandardModel**.

However, such a model may not necessarily pick up topics that have high correlations with aspect scores. To potentially obtain such more discriminative topics, we propose an alternative approach, which is to use guidance from the review scores in the different aspects. To this end, for each aspect of the review, we learn two topic models using either the high-scoring or the low-scoring articles. The idea here is that the topics learned using high-scoring

articles can be expected to capture common patterns present in them, which may serve as useful indicators of a good score. Similarly, the topics learned using low-scoring articles may pick up common patterns in articles of lower quality. We note that while supervision was used for splitting the data, the topic modeling algorithm remains unsupervised. This is in sharp contrast to some existing supervised topic models where a specific topic modeling algorithm is tied to labeled data for supervision [171]. We refer to these models as **AspectGuided**.

In the experimental results, we analyze the performance of two versions of the Aspect-Guided model. In the first one, denoted **Multi-view**, for each article we generate topical features using all AspectGuided models (two models for each aspect), regardless of the review aspect score to be predicted. In the second one, denoted **Single-view**, we generate topical features using only the aspect to be predicted. We experiment with these two versions to study the extent to which topical features generated from modeling one review aspect may also be useful for predicting scores in another aspect.

6.3.3 Multi-scale Topic Modeling

We may further learn different models by using text segments extracted from the original articles in different levels of granularity. By doing so, we expect to capture semantics at different levels, which may be necessary for supporting automated assessment. For example, topic models learned using a low granularity of text (i.e., long text segments) may be able to capture high-level patterns, while models with high granularity (i.e., short text segments) may capture more implicit ones. Furthermore, prediction of performance in different review aspects may rely on different granularity levels of information. Technically, we split each article into n segments.¹ Then, we feed the model with the text segments, treating them as individual and independent documents.

6.3.4 Generating Topic Model Features

As discussed earlier, we use the data set in order to learn various topic models so as to obtain multiple views of the text data. Specifically, for each of our suggested approaches for topic modeling (StandardModel and AspectGuided), we learn several topic models by varying the level of text granularity (n) and the number of topics (k). Once we obtained those topic models, the next step is to define topical features and their values.

The multinomial distribution of the j 'th topic in a topic model with k topics and a granularity level of n is denoted $\theta_j^{n,k}$. The coverage of a topic in an article segment d_i (the

¹In this thesis, we split the articles into equal-length segments as defined by the number of sentences.

i 'th segment of an article d , $i \in \{1, 2, \dots, n\}$) is measured using an approximation of the Kullback-Leibler divergence (KL) between the distribution of the topic $p(\cdot|\theta_j^{n,k})$ and of the article segment $p(\cdot|d_i)$ over the vocabulary terms.²

$$score(\theta_j^{n,k}, d_i) = \sum_{w \in V: p(w|\theta_j^{n,k}) > 0} p(w|d_i) \log \frac{p(w|d_i)}{p(w|\theta_j^{n,k})}; \quad (6.1)$$

where w is a word in the vocabulary V . $p(w|d_i)$ is estimated using the maximum likelihood approach, that is $p(w|d_i) = \frac{tf(w \in d_i)}{|d_i|}$; $tf(w \in d_i)$ is the number of occurrences of w in d_i and $|d_i|$ is the total number of words in d_i . In order to generate a topical feature for each article, the scores of the different article segments are aggregated as follows:

$$f_j^{n,k}(d) = \log \left(1 + \max_{i \in \{1, \dots, n\}} score(\theta_j^{n,k}, d_i) \right); \quad (6.2)$$

$j \in \{1, \dots, k\}$, i.e., for a single topic model we generate k topical features per article. We use the max aggregation function in order to capture for each article the most salient features. Indeed, our experiments showed that this approach performs better than other approaches such as taking the average or using all features; we do not report the actual results as they do not convey further insight.

An alternative approach for estimating $score(\theta_j^{n,k}, d_i)$ would be to directly use the distribution of documents over topics, $\{\pi_{d_i}^{n,1}, \pi_{d_i}^{n,2}, \dots, \pi_{d_i}^{n,k}\}$; $\pi_{d_i}^{n,j}$ is the coverage of topic j in the i 'th segment of article d . This distribution is learned for articles in the training set and can be easily inferred for unseen documents. However, our experimental results showed that using this distribution is not as effective for automatic assessment as using the KL-divergence measure as in Equation 6.1. Thus, we have mainly used the KL-divergence measure in most of our experiments. In Section 6.5.2 of the empirical evaluation, we empirically analyze the difference between the two approaches.

6.3.5 Feature Combination

We explore different approaches for combining the features extracted using different topic models, and combining topical features with bag-of-words features. In the first approach, denoted **FeatureComb**, we pool all features together to form a larger feature set. Then, the weights for each feature can be learned using any supervised machine learning algorithm. We will further discuss the specific algorithm we used in our experiments in the next section.

²This is an approximation as the summation is only over terms with positive probabilities in $\theta_j^{n,k}$.

One potential limitation of the FeatureComb approach is that when we have many features, the machine learning program may not necessarily assign the optimal relative weights among all the topics since the topics would be mixed in the feature representation. Furthermore, in the experiments, we also study the combination of topic model features and word features for which this problem might even be worse due to a large number of words in the vocabulary.

To address that limitation, we propose a second approach where we learn several models corresponding to different types of features. Then, we combine for each article the prediction results according to each model. In such an approach, the relative weights among the topical features can be optimized when we train separate machine learning models. The method we propose to that end takes the average of the prediction scores of an article in the different models and is denoted **ScoreComb**.

6.4 EXPERIMENTAL SETUP

6.4.1 Data Sets

Two data sets were used for the evaluation. The first data set is of articles in the veterinary medicine domain and the second one is in the computer science domain. Below, we provide the details about those two data sets.

VetMed We used a collection of articles, written by first-year veterinary medicine students. The articles were written by the students as part of their training in clinical problem-solving. Specifically, the students were given the assignment of analyzing a clinical case by the development of a multimedia-containing text document. Students were asked to provide answers to specific questions about the case and also to connect the animal’s problems to their basic physiology and anatomical understanding. The exercise was designed also to challenge the students to reflect upon elements of the case that forced deeper self-study and/or review. Furthermore, students were asked to identify and justify the references that they chose.

We used articles written by students in two consecutive years for the evaluation. Both classes were given the same case with the same instructions; students in both classes were at the same level of their studies. We used the articles of the most recent class as a test set (134 articles), and of the other class as a training set (160 articles). In general, the analyses ranged from about 1,400 to 3,400 words in length, with the average being about 2,400 words; students were not confined to a specific structure. Students were also provided with the different review aspects that were used to assess the quality of their works as follows:

1. *Problems (Pro.)*: The students should list the three most serious clinical problems in the case and defend their reasoning.
2. *Differentials (Dif.)*: The students should identify at least two major differential diagnoses for the animal and defend their choices with evidence from the case and information from the literature.
3. *Evidence (Evi.)*: The students should identify the clinical observations in the case to support their problem list and differential diagnosis list.
4. *Understanding (Und.)*: The students should respond to various questions to evaluate their understanding of the case (for example: “If unmanaged, what kind of additional clinical signs would you expect?”).
5. *Conclusions (Con.)*: The students should identify and explain at least two personal learning issues from the exercise.
6. *References (Ref.)*: The students should provide references that helped them to understand the case.
7. *Overall (Ove.)*: The overall impression of the reviewer from the analysis.

The articles were scored in each of the review aspects by three peer-reviewers (also students in the class) as part of the formative feedback phase between a first and final draft of the authoring student’s work. We use the average score of the reviewers in each aspect (each reviewer selects a score for each aspect from $\{0, 1, 2, 3, 4\}$). In Table 6.1, we report the inter-reviewer agreement in terms of the standard deviation of scores (Std) and the number of reviewers who agreed on a specific score (Agreements).

ICLR For the evaluation, we also leveraged the PeerRead collection of research articles in the computer science domain [10]. Specifically, we used the articles of the ICLR’2017 conference in our experiments since these include manually annotated scores for the articles with respect to the different review aspects. This data set consists of 427 articles where each article is ideally annotated in eight different review aspects. A review aspect score is given by at least one reviewer and by four reviewers at the most; in the case of multiple reviewers, we use the average score as the review aspect score for the article; a single score is selected from $\{1, 2, \dots, 5\}$. The inter-reviewer statistics for this data set are reported in the bottom block of Table 6.1.

Table 6.1: Inter-reviewer agreement. The average standard deviation of scores (Std) and the average number of reviewers who agreed on a score (Agreements) are reported.

		VetMed						
		Pro.	Dif.	Evi.	Und.	Con.	Ref.	Ove.
Training	Std	.476	.530	.611	.616	.479	.513	.617
	Agreements	2.6	2.3	2.1	2.0	2.8	2.6	2.1
Test	Std	.539	.550	.614	.679	.517	.574	.511
	Agreements	2.4	2.4	2.0	1.8	2.8	2.4	2.4

		ICLR				
		Cla.	Ori.	Sou.	Sub.	Imp.
Training	Std	.203	.288	.183	.160	.164
	Agreements	1.4	1.3	1.3	1.2	1.1
Test	Std	.234	.304	.149	.241	.183
	Agreements	1.4	1.2	1.3	1.2	1.0

Table 6.2: Statistics of the ICLR data set: (1) The number of training and test examples in each review aspect. (2) The mean and maximal number of reviewers in each review aspect.

		Cla.	Ori.	Sou.	Sub.	Imp.
# Papers		314	335	288	226	230
Training	Mean(# Reviewers)	1.7	1.8	1.6	1.5	1.4
	Max(# Reviewers)	4	3	3	3	4
# Papers		30	32	29	20	23
Test	Mean(# Reviewers)	1.8	1.6	1.5	1.6	1.3
	Max(# Reviewers)	3	3	3	3	3

In the ICLR data set, it is often the case where a paper has a score for only some of the review aspects. Thus, the data set for each aspect can differ in size. In this thesis, to learn sufficiently effective supervised models and to evaluate them properly, we focus on aspects that have at least 200 training examples and at least 20 articles in the test set, resulting in 5 review aspects; we used the given splits of the data set to training/test/validation (we merged the validation and training set to improve performance). The number of articles in each aspect set and the mean and maximal number of reviewers are summarized in Table 6.2. The instructions given to the reviewers for scoring the five aspects that were used in our experiments are as follows:³

1. *Clarity (Cla.)*: For the reasonably well-prepared reader, is it clear what was done and why? Is the paper well-written and well-structured?

³This part was taken directly from the original paper [10].

2. *Originality (Ori.)*: How original is the approach? Does this paper break new ground in topic, methodology, or content? How exciting and innovative is the research it describes?
3. *Soundness (Sou.)*: First, is the technical approach sound and well-chosen? Second, can one trust the empirical claims of the paper – are they supported by proper experiments and are the results of the experiments correctly interpreted?
4. *Substance (Sub.)*: Does this paper have enough substance, or would it benefit from more ideas or results?
5. *Impact (Imp.)*: How significant is the work described? If the ideas are novel, will they also be useful or inspirational? Does the paper bring any new insights into the nature of the problem?

6.4.2 Implementation Details

Topic model implementation The text was extracted from the articles and then pre-processed, including stopword removal and Porter stemming, using the NLTK library;⁴ we use only the first 1000 words in an article and exclude words that do not appear in at least 5 articles in the data set. For the implementation of the LDA model, we used the Scikit-learn library (Online Variational Bayes was used for inference).⁵ We learn topic models using two levels of text granularity (n): (1) *FullText*: articles are not split into paragraphs ($n = 1$), (2) *Paragraphs*: articles are split into three equal-length paragraphs ($n = 3$). The number of topics was set from $\{5, 15, 25\}$ using cross-validation; the validation set is a random sample of 15% of the training set.

Learning framework The proposed topical features (and the baselines as well) can be used in any machine learning framework. In this thesis, however, our focus is on studying the effectiveness of various features, so we want to fix the learning framework. Thus, in our experiments, we used the regression approach to predict the scores for an article in the different review aspects. Specifically, we used linear regression in our experiments to predict the score on an article using any type of article representation.

Evaluation metric We report the Kendall’s- τ correlation between the ranking of articles based on their known scores and the ranking of the same set of articles based on the scores

⁴<https://www.nltk.org> (accessed August 25, 2021)

⁵<https://scikit-learn.org> (accessed August 25, 2021)

produced using our automated assessment method. The correlation takes values between $[-1,1]$ where -1 and 1 correspond to a perfect negative and positive correlation, respectively.

6.4.3 Baselines

Word embeddings A common approach for text representation is to use word embeddings. This approach was also used in the previous work on the automatic assessment of research articles [10]. Word embeddings are dense vector representations of words that capture the complex semantics of text (e.g., [99, 107]). The main approach for using word embeddings for text classification/regression is to combine them to obtain a representation for the entire text by using neural networks with the most popular ones being LSTM [117], CNN [175], and transformers [107]. Then, the combined representation is used to predict a score/label for the text. The word embeddings can be learned from scratch using the final task objective or pre-trained using a large data set of general text [107]. In this thesis, we use two representative models from this line of work: LSTM and BERT.

We learn an LSTM model [117] which consists of a word embedding layer (with dimension size of 50), trained from scratch using the data set at hand, and a single LSTM layer; we used the TensorFlow library.⁶ Furthermore, we used a sequence length of 100 words in the case of LSTM due to its limited capability in effectively learning from very long sequences; the vocabulary of the model was set to the 1000 most frequent terms. Similarly to the implementation of the topic models, the number of hidden units in the LSTM layer was selected from $\{5, 15, 25\}$ using cross-validation. Finally, the last hidden state was used as an input to a feed-forward layer that outputs the predicted score. The model was trained using the Mean Squared Error loss (MSE) for 3 epochs with a batch size of 16 and the Adam optimizer.

For the BERT model, we used SciBERT [54] which is a BERT model that was pre-trained on a large collection of research articles in different domains. We fine-tuned this model using our data with the Huggingface⁷ library (3 epochs and a batch size of 16).

Unigrams A simple, yet effective approach for text representation is the bag-of-words. According to this approach, all words in the article are used to represent it where each word is associated with a weight. A popular choice for the weight function that we also use in this thesis is $tf.idf$, which is the multiplication of the term frequency (tf) in the article and the inverse document frequency (idf) of the term in the collection. Due to the “soft” matching

⁶<https://www.tensorflow.org> (accessed August 25, 2021)

⁷<https://huggingface.co> (accessed August 25, 2021)

Table 6.3: The performance (Kendall’s- τ) of using Unigrams, Topics, LSTM, and BERT for the automatic assessment of research articles. ‘F’ stands for the FeatureComb method and ‘S’ stands for the ScoreComb method. Boldface: best result in a column.

Approach	VetMed							ICLR					
	Pro.	Dif.	Evi.	Und.	Con.	Ref.	Ove.	Cl.	Ori.	Sou.	Sub.	Imp.	
Unigrams	.235	.146	.315	.258	.148	.302	.229	-.143	.184	-.085	.213	.065	
LSTM	F-Single	.092	.030	.038	.017	.063	.005	.043	-.016	-.086	.014	.052	.144
	F-Multi	.093	.049	-.059	.046	-.006	.074	-.051	.290	-.004	.162	-.155	-.126
	S-Multi	.105	.086	-.109	.051	-.001	.094	-.041	.042	-.128	-.008	-.167	.074
BERT	F-Single	.060	.128	.136	.029	.085	.128	.120	.216	.116	.140	.029	-.344
	F-Multi	.098	.136	.134	.128	.096	.280	.152	.169	.094	.267	.236	-.187
	S-Multi	.093	.136	.138	.088	.069	.247	.154	.016	-.056	.047	.201	-.170
Topics	F-Single	.028	.084	.079	.325	.116	.211	.241	.274	.030	.377	.017	.065
	S-Single	.303	.338	.303	.334	.210	.380	.274	.216	.141	.173	.098	.048
	F-Multi	.106	.066	.119	-.018	.008	.294	-.006	.174	.094	-.014	.132	-.048
	S-Multi	.317	.318	.376	.375	.214	.394	.277	.348	.214	.168	.305	-.030

inherent in a topic (word distribution), topic features may not be as discriminative as lexical features. In general, they may be complementary to each other. We thus also explore the combination of topics and Unigrams using the approaches in Section 6.3.5.

6.5 EXPERIMENTAL RESULTS

6.5.1 Main Findings

Using topic models for automatic assessment The performance of using topic models for the automatic assessment of research articles is presented in Table 6.3. We compare the performance of using topics with that of using simple lexical features (Unigrams), LSTM, and BERT. The results in Table 6.3 demonstrate the effectiveness of topic models for the task. Specifically, for all but one review aspect, the best performance is obtained when using a topic model-based approach. Comparing the FeatureComb (‘F’) approach with the ScoreComb (‘S’) approach, we can see that the ScoreComb approach is often better. Specifically, in the case of VetMed, ScoreComb outperforms FeatureComb for all review aspects for both Multi-view and Single-view features. In the case of ICLR, on the other hand, ScoreComb outperforms FeatureComb only in the case of Multi-view. A possible reason for this can be that in the case of Multi-view the number of features is much larger which makes the ScoreComb approach better for this case. The results also show that Multi-view topic model features usually perform better than Single-view features. We conclude from the results in Table 6.3 that the topic modeling approach S-Multi is overall the most effective among the methods compared, outperforming all baselines in the majority of review aspects for both

data sets.

Using LSTM does not generally result in a good performance for both data sets. In the case of VetMed, LSTM never outperforms the Unigrams baseline. More specifically, LSTM achieves a correlation greater than 0.1 only in a single case for VetMed. LSTM achieves better performance than Unigrams for three review aspects in ICLR (in one of the cases the performance achieved is the highest among all methods); two of those cases are for Multi-view and one of them is for Single-view. A possible explanation for the difference between the data sets can be that the training set in the case of ICLR is larger. Differently from the case of Topics, however, the best performance of LSTM is obtained for the FeatureComb models. Still, the LSTM model is not very robust to the review aspects. For example, using the F-Multi approach results in improvements over using Unigrams in two aspects and results in negative correlations in two other.

Next, we turn our focus to BERT. In the case of VetMed, it performs better than LSTM with correlations greater than 0.1 for five out of the seven review dimensions. Still, it never outperforms the Unigrams approach. Similar to the case of LSTM, in the case of ICLR, BERT outperforms Unigrams for three review aspects and performs the best with the FeatureComb approach. As for the effectiveness of Multi-view compared with Single-view, we can see that it is sensitive to the test dimension. Finally, we can see that BERT is generally more robust than LSTM, resulting in negative correlations only for a single aspect.

An interesting finding, based on Table 6.3, is that combining features using their prediction results is more effective than the standard approach of pooling all features together for combining topical features. These results suggest that when we pool different kinds of features together, the learning algorithm may not be able to optimize the relative weights on the same type of features as well as when we train a separate classifier for each group of features separately. It would be interesting to further investigate this issue in future work.

Another observation from Table 6.3 is regarding the performance sensitivity to the review aspect. The results show that while all approaches find some aspects equally hard to predict (e.g., Imp. and Con.), for other dimensions some approaches can be much better than others (e.g., comparing the performance of Topics in Dif. with all other approaches). That said, using semantic representations can generally improve the performance of simple lexical approaches.

Finally, the results also show that performance patterns can vary dramatically in some cases. Specifically, some approaches perform better than others in certain aspects of quality, and the opposite holds in the case of other review aspects. One example of such a case is when comparing the Multi-view approach with the Single-view approach for topical features. While in many quality aspects the Multi-view approach performs better than the Single-view

approach, in some aspects the opposite trend is observed. A possible explanation for this can be that some quality aspects have low correlations with several other quality aspects. For this reason, when using topics learned with the guidance of quality aspects that are different than the target quality aspect, it may degrade the effectiveness of the prediction model. Another example of inconsistent performance patterns is in the case of BERT. Specifically, the performance of BERT for some quality aspects can be close to that of Topics but very poor in other cases. A possible explanation for this can be that some dimensions of review are better captured by the pre-trained model, which relies on a large and diverse set of articles, than by the topics learned only from the data at hand. For example, it might be the case that the Originality aspect can be hardly captured by using pre-training as it is a highly domain-specific aspect. On the other hand, it might be easier to generalize information regarding other aspects, such as Clarity, which might be shared by articles in different research domains. These findings suggest that there might not be a one-size-fits-all solution for all review dimensions as each one has its unique characteristics. Studying techniques that automatically adapt for a specific review dimension is an interesting future work direction worth exploring.

Table 6.4: The performance (Kendall’s- τ) of BERT as a function of the training data size in the ICLR data set. The average (standard deviation) of the performance of 10 models (based on 10 random samples) is reported. Boldface: best result in a column.

# Examples	Clarity	Originality	Soundness	Substance	Impact
50	.179 (.127)	.012 (.054)	.062 (.039)	.055 (.091)	.094 (.070)
150	.149 (.098)	.035 (.105)	.206 (.111)	.138 (.137)	-.166 (.152)
250	.122 (.153)	-.052 (.121)	.166 (.112)	.029	-.344
350	.216	.116	.140		

Analysis of BERT The results in Table 6.3 show that BERT is generally not as effective as other techniques for the task. This finding is interesting as in other prediction problems, using BERT resulted in state-of-the-art performance [54, 107]. A possible explanation for this finding may be the small size of the training data used for fine-tuning the assessment models (160 training examples in VetMed and 200-300 examples in ICLR). To further investigate this issue, we varied the training data set size used for fine-tuning the model. The results of this experiment are presented in Table 6.4. Specifically, for each data size, we report the average performance (and standard deviation in the parenthesis) of 10 models learned using different random samples of the data. Note that the last number in each column corresponds to using the entire training data set (no sampling). We conducted this experiment only for ICLR as in VetMed the data set is much smaller and does not enable us to perform this type of analysis.

The results in Table 6.4 demonstrate the sensitivity of BERT to the small training data set. Specifically, for some of the review aspects (Clarity and Originality), increasing the data set size results in better performance. This finding might suggest that more data is needed to improve the performance of those models in the case of some review aspects. For other review aspects, on the other hand, it is interesting to see that the highest performance is obtained for using fewer training examples than in the entire data set. Still, the high variance of performance across samples indicates high sensitivity to the data sampled. For this reason, in practice, it is not clear how to obtain a good sample of the data for fine-tuning and perhaps a model ensemble approach should be leveraged to this end. How to optimize BERT to our task in which the data available for fine-tuning is very small is an interesting question worth exploring in future work.

Table 6.5: The Jaccard index between the prediction mistakes of different approaches.

Method	VetMed							ICLR				
	Pro.	Dif.	Evi.	Und.	Con.	Ref.	Ove.	Cl.	Ori.	Sou.	Sub.	Imp.
Topics vs. Unigrams	.402	.462	.553	.417	.456	.371	.490	.519	.379	.640	.385	.368
BERT vs. Unigrams	.495	.415	.417	.451	.450	.413	.472	.480	.269	.577	.412	.364
Topics vs. BERT	.440	.390	.396	.394	.463	.418	.464	.455	.400	.407	.333	.619

Table 6.6: The performance (Kendall’s- τ) of combining different models. ‘F’ stands for the FeatureComb method, and ‘S’ stands for the ScoreComb method. ‘All’ stands for a method that combines Unigrams, LSTM, BERT, and Topics. Boldface: best result in a column.

Approach		VetMed							ICLR				
		Pro.	Dif.	Evi.	Und.	Con.	Ref.	Ove.	Cl.	Ori.	Sou.	Sub.	Imp.
Unigrams		.235	.146	.315	.258	.148	.302	.229	-.143	.184	-.085	.213	.065
LSTM		.105	.086	.038	.051	.063	.094	.043	.290	-.004	.162	.052	.144
BERT		.098	.136	.138	.128	.096	.247	.154	.216	.116	.267	.236	-.170
Topics		.317	.338	.376	.375	.214	.394	.277	.348	.214	.377	.305	.065
Topics + Unigrams	F	.341	.210	.332	.332	.196	.421	.260	-.095	.176	-.014	.443	.030
	S	.311	.337	.382	.376	.219	.399	.279	.121	.205	.069	.305	.126
All	F	.271	.231	.305	.372	.207	.389	.248	.195	-.081	.195	.432	-.048
	S	.319	.343	.255	.400	.144	.402	.236	.201	.223	.140	.282	-.135

Model combinations A question that comes up from the results in Table 6.3 is to what extent the studied approaches make similar prediction mistakes. Answering this question can help deepen our understanding of whether the studied techniques share strengths/weaknesses and should be combined. To answer this question, we examined the overlap between the prediction mistakes of some of the automatic assessment techniques. We first ranked the articles according to each review aspect score using the human-annotated scores. Then, we split the ranked list into three equal-sized groups of research articles at the top, middle, and bottom of the list. We then regard a research article that the algorithm placed in the wrong

group as a mistake. Finally, we calculated the Jaccard index between the sets of mistakenly predicted research articles of two approaches. The results are presented in Table 6.5; we fixed the configuration of the models based on the best performing one for each aspect according to Table 6.3 (i.e., ScoreComb vs. FeatureComb and Multi-view vs. Single-view).

According to the results, we can see that many of the prediction mistakes are not shared between the different assessment techniques. Specifically, for most cases in the table, the Jaccard index is lower than 0.5. The results also show that the overlap between methods depends on the specific review aspect. Specifically, the pair of assessment approaches with the highest overlap value often changes depending on the quality aspect. Furthermore, the results suggest that some of the information used for the prediction of research quality in different aspects can be shared among techniques and that the extent of this depends on the actual quality aspect score to be predicted. Specifically, in six of the review aspects, the highest overlap value is between Topics and Unigrams, which may be because both rely on the bag-of-words assumption. In five of the review aspects, the highest overlap value is between Topics and BERT, which can be attributed to the fact that both are semantic dense representations. Finally, only in three review aspects, the overlap between BERT and Unigrams is the highest. This result makes sense as these two approaches rely on very different assumptions.

Motivated by the results in Table 6.5, we analyze the performance of combining Topics with Unigrams as well as of combining the different semantic models studied in this thesis. The results are presented in Table 6.6. In the top block of the table, we report the performance of the individual models (the highest performance for those models based on Table 6.3 is reported). In the bottom two blocks, we report the performance of different model combinations.

Focusing on the combination of Topics with Unigrams, we can see that its effectiveness is sensitive to the review aspect. Specifically, the combination results in higher performance than that of the individual components in the majority of dimensions in the VetMed data set; still, the improvements are substantial only in three aspects. In the case of ICLR, we observe improvements in only two dimensions (Sub. and Imp.). A possible explanation for this can be the relatively low performance of Unigrams for the ICLR data set compared with Topics. Finally, similarly to the results in Table 6.3, ScoreComb is better performing than FeatureComb for combining Topics and Unigrams.

Moving on to the combination of all semantic models with Unigrams ('All'), we can see that in the majority of relevant comparisons it does not outperform the individual components. Still, we can see that for two review dimensions in the VetMed data set (Dif. and Und.) and for a single dimension in ICLR (Ori.) substantial improvements are observed when using the

ScoreComb method. This might suggest that such a combination can be of merit in some cases. Studying this type of combination is not at the focus of this work and we leave it for future work.

6.5.2 Analysis of Topic Model Features

Number of topics In the following, we analyze the performance of the different topic model approaches as a function of the number of topics used. We are interested in studying the sensitivity to this parameter as in all other results in this chapter we set it up using cross-validation. The results of this analysis are reported in Figure 6.1 and 6.2 for VetMed and ICLR, respectively. According to the figures, the optimal number of topics depends on the automatic assessment approach and the review aspect.

Focusing on the ScoreComb approach, we can see that it is usually more robust to changes in the number of topics than FeatureComb. Furthermore, in most graphs, ScoreComb-Multi achieves the highest performance among all methods when the number of topical features is set optimally. In the case of the FeatureComb approach, on the other hand, we can see that it is often sensitive to the number of topics used. Specifically, the graphs show that changing the value of the number of topics can have a dramatic influence on the performance of the model. A possible explanation for the difference observed between these two approaches is that ScoreComb is more robust since it combines multiple prediction models and FeatureComb is more sensitive to the number of features since it uses a single model.

Another finding from the figures is that in many cases, the curves for ScoreComb are increasing as a function of the number of topics and the opposite usually holds for FeatureComb. This result insinuates that one of the problems of FeatureComb is the limited ability to learn effective weights of many different features that are combined in a single model.

Finally, the results also demonstrate the sensitivity of the optimal value for the number of topics to the review aspect. For example, while the optimal value for the number of topics is 25 for the Substance dimension when using ScoreComb-Multi, it is 5 for the Impact dimension. This result suggests that characteristics of different review aspects are captured differently in articles and thus require a different number of topics.

Topic model components In Table 6.7, we compare the performance of the different topic models used to generate the topical features. Specifically, we report the performance of the ScoreComb approach, the best performing one according to Table 6.3, when different topic models are used for topical feature generation. AspectGuided topics built using only the high scoring articles and only the low scoring articles are denoted AspectGuided+ and

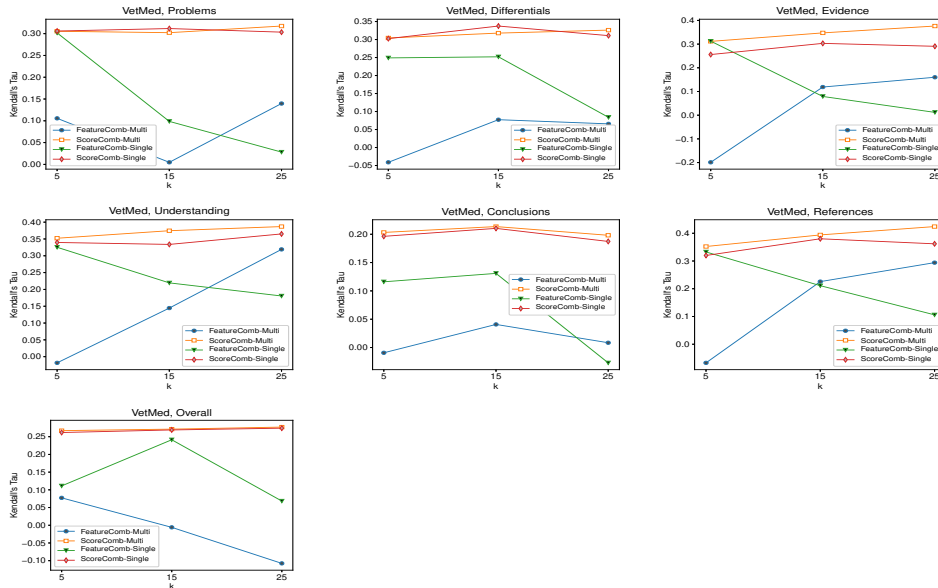


Figure 6.1: The performance of different approaches for automatic assessment using topic models as a function of the number of topics, k (VetMed data set). Note: figures are not to the same scale.

AspectGuided-, respectively.

According to the results, both components of AspectGuided are highly effective. Using these topical features outperforms the Unigrams baseline in all review aspects in the VetMed data set. In the case of ICLR, for all review dimensions, at least one of these approaches outperforms the baseline. Comparing AspectGuided+ with AspectGuided-, we can see that the best performing model in most aspects, for both data sets, is AspectGuided+. Still, it is worth mentioning that for some dimensions, AspectGuided- performs better; specifically, this is the case for two review aspects in each data set. This result shows that for some review dimensions, topics of high scoring articles are more indicative of the target score while for other dimensions it is the topics of low scoring ones. Finally, the results demonstrate the effectiveness of combining AspectGuided+ and AspectGuided-. For all review aspects in both data sets, AspectGuided does not result in a substantial performance decrease compared with the best performing component. Furthermore, in ICLR, AspectGuided outperforms the individual model components in three out of five dimensions.

The results in Table 6.7 also show that StandardModel does not perform as well as AspectGuided in the majority of review aspects. This finding can be explained by the fact that StandardModel is fully unsupervised. That said, StandardModel still performs very well as a stand-alone model. Specifically, it outperforms Unigrams in all review aspects for both data sets. StandardModel also results in the highest performance for a single review aspect

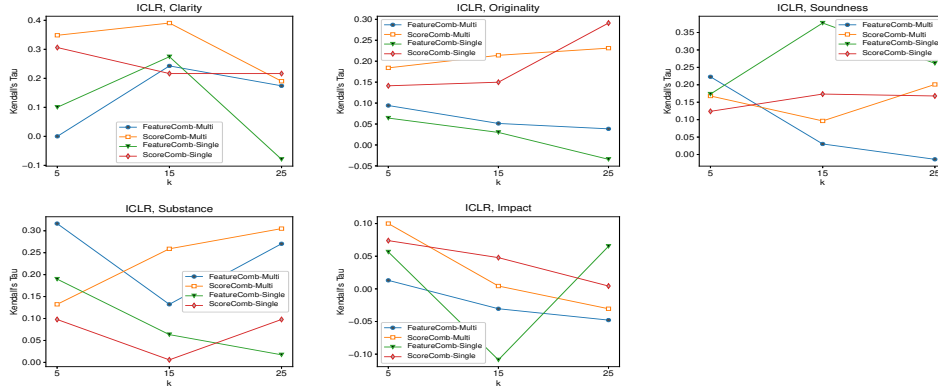


Figure 6.2: The performance of different approaches for automatic assessment using topic models as a function of the number of topics, k (ICLR data set). Note: figures are not to the same scale.

in both data sets. This shows that sometimes the unsupervised model can better capture useful information compared with the AspectGuided models.

Finally, we see that the approach of combining all models is generally very effective. For six dimensions in VetMed and two in ICLR, combining all models results in either the highest or very close the highest performance. Yet, in some cases, as can be seen in Table 6.7, the combination of all models can decrease the performance compared with the highest scoring model (e.g., in the case of Ove. and Sub.). In future work, we plan to further study the combination of those models to make it less sensitive to the review dimension.

Table 6.7: Topic model components analysis. The performance (Kendall's- τ) of Standard-Model vs. AspectGuided. The ScoreComb approach was used for feature combination. Boldface: best result in a column.

Method	VetMed							ICLR				
	Pro.	Dif.	Evi.	Und.	Con.	Ref.	Ove.	Cla.	Ori.	Sou.	Sub.	Imp.
Unigrams	.235	.146	.315	.258	.148	.302	.229	-.143	.184	-.085	.213	.065
AspectGuided+	.309	.314	.376	.371	.232	.358	.307	.238	.163	.146	.374	.083
AspectGuided-	.294	.299	.378	.380	.184	.327	.254	.301	.214	.118	.213	-.144
AspectGuided	.304	.314	.377	.379	.210	.352	.282	.333	.218	.146	.328	.109
StandardModel	.301	.263	.337	.313	.203	.398	.231	.042	.184	.217	.109	.039
All Topics	.317	.318	.376	.375	.214	.394	.277	.348	.214	.168	.305	-.030

Text granularity analysis In the following analysis, we study the effectiveness of topic models that were learned using different levels of granularity of the text data. As a reminder, in this dissertation, we learn topic models using two levels of granularity. We either use all of the article's text (denoted Full Text) or split an article into three paragraphs (denoted Paragraphs). The results for this analysis are presented in Table 6.8. According to the

results, we can see that the optimal level of granularity depends on the data set at hand. Specifically, while the Paragraphs model performs better than Full Text in five out of six dimensions in VetMed, it is the case only for a single dimension in ICLR. A possible explanation for this can be that the articles in ICLR usually share a similar structure while the articles in VetMed were written by students without any restrictions regarding a specific structure. Finally, the results show that the approach taken in this thesis of combining both levels of granularity is usually the best performing one. This result, together with the result in Table 6.7, demonstrates the effectiveness of using topic models that were learned using different views of the text data.

Table 6.8: The effectiveness of topic models learned from text with different levels of granularity. The ScoreComb approach was used for feature combination. Boldface: best result in a column.

Granularity	VetMed							ICLR				
	Pro.	Dif.	Evi.	Und.	Con.	Ref.	Ove.	Cla.	Ori.	Sou.	Sub.	Imp.
Full Text	.272	.286	.324	.340	.182	.245	.266	.264	.257	.047	.259	-.022
Paragraphs	.315	.295	.373	.360	.187	.445	.238	.238	.137	.228	.178	-.135
Both	.317	.318	.376	.375	.214	.394	.277	.348	.214	.168	.305	-.030

Methods for construction of topical features As mentioned previously, a natural way to use topics as features in supervised learning would be to use the distribution of a document over the topics. This distribution is learned jointly with the topic distributions and thus can be provided for documents in the training data. For unseen documents, it is fairly easy to infer this distribution by following the generative mechanism. Yet, we found that in our case using such an approach is not as effective as measuring the distance between the topic distribution and the document distribution over terms. In Table 6.9, we further explore this finding by comparing different approaches for the construction of topical features. We experiment with the following approaches: (1) Distribution: using the distribution of a document over topics. (2) KL: using KL-divergence between the topic distribution and the document distribution over terms (this approach was used throughout this chapter). (3) KL-norm: sum normalizing the KL-divergence-based features so that all topical features per topic model would sum up to 1. We report the performance of using only topical features, combined with the ScoreComb method. The analysis also compares two different inference approaches for the LDA model, based on different implementations. The first approach is Online Variational Bayes, implemented as part of the Scikit-learn library. The second approach is Gibbs sampling, implemented as part of the Gensim library.⁸

⁸<https://radimrehurek.com/gensim/models/ldamodel.html> (accessed August 25, 2021)

Table 6.9: Comparing the performance (Kendall’s- τ) of different approaches for topical feature construction. Only topical features are used and are combined using the ScoreComb approach. Boldface: best result in a column in an inference method block.

Method	VetMed						ICLR					
	Pro.	Dif.	Evi.	Und.	Con.	Ref.	Ove.	Cla.	Ori.	Sou.	Sub.	Imp.
	OVb											
Distribution	.267	.287	.345	.367	.171	.389	.294	.174	.158	.245	.293	.109
KL	.317	.318	.376	.375	.214	.394	.277	.348	.214	.168	.305	-.030
KL-norm	.270	.303	.297	.378	.122	.438	.212	.253	.227	.085	.282	.030
	Gibbs											
Distribution	.253	.239	.314	.371	.100	.323	.223	.132	.056	.228	.293	.074
KL	.306	.295	.297	.391	.161	.321	.284	.116	.167	.096	.293	-.039
KL-norm	.209	.199	.309	.311	.145	.352	.203	.179	.124	.019	.190	-.126

Comparing the performance of using document distributions with that of using KL, we can see that the latter is overall more effective. Specifically, using KL outperforms the document distribution approach in the majority of aspects for both data sets and inference methods. A possible explanation for this finding might be attributed to the comparability of features. That is, in the case of document distribution over topics, a non-negative probability must be assigned to each topic, and all probabilities must sum up to 1. Thus, it might be the case where a document is assigned a probability for a topic just to satisfy this condition. This is of course not the case when KL is used. We experiment with the KL-norm approach to test this hypothesis. Indeed, we can see in Table 6.9 that the performance of KL drops for most review aspects when normalization is applied. Moreover, it is interesting to see that sometimes using normalization improves the performance of KL in cases where Distribution has similar/higher performance than KL (e.g., in Cla. and Ref.). This result suggests that there can be some review aspects that would benefit from the normalization of topical features.

We note that further exploration must be done to reach conclusive findings regarding this issue. This should include, for instance, the exploration of other data sets and other supervised learning algorithms. Such exploration is out of the scope of this work and is left for future work.

6.5.3 A Case Study

To demonstrate the interpretability of topic model features for the task, we use a case study in this section. For the simplicity of discussion, we focus on a single review aspect in each data set (Overall in VetMed and Clarity in ICLR). In Table 6.10, we present 15 representative terms for the four most correlative topics with the scores in the corresponding aspect. Terms in each topic were extracted as follows. We first extract 100 terms from each

topic based on their probability in the topic’s multinomial distribution. Then, for each topic, we leave only the terms that do not appear in the other three topics. Finally, we use the 15 terms with the highest probabilities. We do that to better distinguish between the different topics. We also present the 15 most correlative terms in the Unigrams baseline (Unigrams+ and Unigrams– are the most positively and negatively correlated terms, respectively).

VetMed First, we analyze the topics of the VetMed data set. The first topic presented in Table 6.10 includes general verbs such as “relate”, “consider”, and “mean”. A positive correlation with such terms might suggest that the author has tried to explain their thinking. Some other terms are appropriate for the references part of the work. For example, “Merck” refers to the common general reference “Merck Veterinary Manual”. The second topic tends to include general terms taken directly from the case provided to the student such as “systolic”, “wave”, and “QRS”. Such terms were used in the case to describe the animal’s condition and the tests that were performed. A positive correlation with these terms might suggest that the author has emphasized details given in the case to better support the analysis and diagnosis. The third topic contains terms that reflect novel findings of the work (for example, the term “sibling” might refer to the finding that the animal’s weight could be compared with the weight of his healthy sibling). On the other hand, this topic also contains more general terms such as “note”, and “found”, suggesting that positive attributes were given to those analyses trying to explain signs, history, and diagnostic findings in the case. The last topic in the table mostly includes generic anatomical terms that most students might have needed to use in high-scoring explanations, such as “procedure”, “infection”, and “septum”.

Next, we examine the terms in the Unigrams baseline. Unigrams+ contains the most positively correlated terms in the articles. Most of the terms, except for “CO”, “II”, and “infection”, are fairly general and might reflect the author integrating and explaining well. We finally turn our attention to examine the most negatively correlated unigrams. Surprisingly, the terms reflect recognizable features of the pathophysiology and correct diagnosis in this case. One explanation for that would be that students scored it negatively because less explanation of alternatives was included by these authors.

ICLR Moving to the ICLR data set, we can see that also in this case the correlated topics can be used to explain the important features for a review dimension. For example, the first topic contains different terms that are usually used in papers in the computer vision domain such as “video”, “pixel”, and “object”. This topic suggests that articles in this domain scored higher in the Clarity dimension than in other domains. The second topic in the table

contains more general terms that can be used in different domains. Still, it is interesting to see that it contains terms such as “zi”, “nj”, and “vi”. A possible explanation for this can be that high-scoring articles made good use of notation, which is usually appreciated and improves the clarity of a paper when used appropriately. Finally, focusing on the fourth topic, we can see terms that are related to model size and efficiency such as “prune”, “efficient”, and “memory”. This suggests that articles that included discussion of this aspect of the research work are overall written more clearly than others.

Finally, we turn our attention to the Unigrams+ words. We can see in the table that it contains words related to sequence models such as “sequence”, “rnn”, “gate”, and “forget”. Based on these words, it seems like the Unigrams model learned to distinguish between articles based on a single domain. This result further demonstrates the low flexibility of using simple lexical features for the task compared with semantic approaches such as topic models.

Discussion The case study presented in Table 6.10 also provides some intuition on why topic model features are effective for the task of assessing the quality of research articles.

One possible way in which topics can be useful for the task is by capturing a writing style that is shared among articles in different research areas and that indicates a specific level of quality in a review aspect. For example, Topic 2 (ICLR) contains general terminology that is potentially shared among a large number of papers in the general area of artificial intelligence (e.g., “variable”, “term”, “variance”, and “bernoulli”). Topic models also can potentially capture a more general style of writing that is not specific to a research area and that can indicate high/low quality in some aspects. For example, an article that proposes an algorithm that is an extension of an existing one would use more frequently words such as “extend”, “revise”, and “modify”. Such an article might be regarded as having lower originality compared with the article that proposed the baseline algorithm. Using topic models to capture the writing style of an article is advantageous since groups of words can serve as a good description of that information.

Another way in which topics are useful for the task is by capturing research areas whose papers generally demonstrate high quality in some aspects compared with papers in other research areas. For example, Topic 1 (ICLR) contains terminology of the computer vision research area. Thus, using this topic, research articles in the general area of computer vision would be expected to score higher than papers in other areas. Using topic models is especially advantageous for this purpose as groups of words can effectively capture a research area.

By taking our proposed topic modeling approach of learning models that capture different views of the text data, one can further learn topics within every research direction that are

correlated with quality scores (e.g., by splitting the articles in the training set based on those scores). By doing that, we expect to find specific words in a research area that would help differentiate between papers on a given topic.

Finally, it might be the case where some of the topics learned using our approach would be effective for predicting multiple quality aspects. In that case, a machine learning framework could be leveraged to determine their relative importance in predicting each quality aspect. For example, while articles in a specific research area can have high originality (e.g., in emerging topics), it might be the case where their clarity is low. In the case of some other topics, on the other hand, it may not make sense to share them between quality prediction models of multiple aspects. In such a case, it might be possible to remove them using feature selection techniques, an approach we leave to study in future work.

6.5.4 The Applicability of Findings to Other Text Classification Tasks

Although we focused on studying a particular application task, there are several findings from our study that apply more broadly to general text classification tasks and other applications of topical features that warrant highlighting. In this section, we discuss some of those findings.

Topical feature generation We first want to discuss how one ought to generate topical features in general. What we have demonstrated in this chapter is that there is much to gain by going a bit further than simply using topic proportion features estimated from an entire corpus and stopping there. Three techniques seem to work well that ought to be generally useful across many different classification tasks for the generation of topical features. The first is to generate more granular topical features than just at the entire document level by considering topics within individual text sections—this seems to improve performance, particularly when one can assume that the documents tend to have a certain underlying structure that can be exploited. The second is to leverage supervision (if available) to discover aspect-specific topics (e.g., finding topics for high-scoring articles and topics for low-scoring articles separately). These are both relatively straightforward tweaks to topic inference but can have a substantial impact as seen in our experiments. Finally, we observed a large performance gap between using inferred topic proportions when compared with computing the KL-divergence between a document (or document-segment) model and each of the individual topics. Both approaches produce k features for k different topics, but the KL-divergence approach achieves strikingly better performance. Though out of the scope for this thesis, it would be extremely interesting to see if this observation generalizes to other

Table 6.10: Topic examples. 15 representative terms for the four most correlative topics with the performance in the Overall aspect in VetMed and the Clarity aspect in ICLR. The two right-most columns correspond to positively and negatively correlated Unigrams, respectively.

VetMed					
Topic 1	Topic 2	Topic 3	Topic 4	Unigrams+	Unigrams-
Vegas	work	weight	small	occur	work
level	obstruct	neutrophils	Jan	weak	ventricle
understand	web	VI	sign	need	hole
echocardiogram	negative	neonatal	infection	evaluation	serious
TOU	wave	ultrasound	sever	heard	major
contraction	narrow	IV	function	II	change
consider	tract	space	septum	back	determine
resistance	lung	found	dilation	reduce	ST
relate	systolic	sibling	return	get	congenital
anesthesia	large	elevated	differential	partial	result
reason	shift	respiratory	report	sufficient	obstruct
number	reverse	range	procedure	infection	like
Merck	QRS	pleural	patent	CO	bypass
mean	pump	PCO	mild	sign	TOF
manual	position	note	medical	mean	animal

ICLR					
Topic 1	Topic 2	Topic 3	Topic 4	Unigrams+	Unigrams-
label	noise	structure	weight	sequence	belong
cnm	rbm	domain	prune	temporal	written
classification	term	infer	time	decompose	visible
video	hidden	variable	reduce	predict	nonlinear
visual	relu	given	size	rnn	acquisition
inform	gaussian	loss	dnn	enough	world
object	nj	latent	accuracy	approach	vice
target	hinton	autoencoder	neuron	forget	vi
supervision	visible	framework	increase	occur	maximum
application	ai	case	sparse	summary	cluster
detect	bernoulli	present	memory	formulate	stream
predict	zi	local	sparsity	sophisticated	sparsity
attack	variance	point	design	neuron	adjust
pixel	boltzmann	minimal	figure	correspond	pipeline
imagenet	vi	estimate	efficient	gate	detail

classification tasks, as it may suggest a superior approach for determining the topic proportion feature weights for any topic-based classification task, which would be an important

finding given that the current practice seems to be taking the topic coverage distributions learned from a topic model directly as feature values for topical features, but this is most likely non-optimal.

Combination of topical features with non-topical features Combining topical features with traditional bag-of-words style features is a common approach to enable a classifier to capture orthogonal representations of a piece of text, but often little thought is placed into the optimal way to perform this combination. As we have shown, it may not be the best idea to simply concatenate the features into one single vector, which is often done in many existing studies. Even a very simplistic model combination approach seems to perform significantly better, which prompts us to recommend this ensemble-based approach in general for combining high-dimension but sparse features with low-dimension but dense features in classification tasks. One could go farther to improve this simple model combination by using “model stacking”, where the outputs of the bag-of-words and topic feature-based models are used as inputs to a third model that is used for the final prediction—this allows for automatic tuning of the amount of trust one places in the individual models. Our positive results in this direction may also suggest that we could potentially develop a more general approach to leverage any latent correlated structures that might exist in a given set of features to partition all the features into different groups, which can then be used to train separate models to be eventually combined.

6.6 RESULT LIST RANKING USING ASPECT SCORES

In the previous section, we demonstrated the effectiveness of using topic models for the automatic assessment of research articles. Our results showed that it is feasible to use text data and a relatively small number of manual annotations for predicting the quality of research articles in different aspects. In this section, we study whether the predicted aspect scores can improve the current literature search systems. Specifically, our main idea is to enrich research articles with aspect scores and use these to enhance current retrieval systems through advanced filtering, sorting, and analysis of the result list. Current literature search engines rely mainly on text-based relevance matching and other limited sources of information such as citation counts and entities. For this reason, aspect scores can potentially improve their performance. For example, filtering results using the Clarity aspect can potentially help to identify articles on a specific topic that are easier to read by novice researchers. In another scenario, for example, using the Originality aspect, one can detect the state-of-the-art approaches in a research area (or approaches that are predicted as such by

the aspect scores).

To study the benefits of using review aspect scores for ranking, we perform an empirical study. Using two data sets of research articles and queries, we examine the result lists obtained by relevance ranking and by aspect score ranking quantitatively and qualitatively. The empirical results show that using aspect scores can substantially improve the quality of the result list. Furthermore, the result lists obtained by using scores in different aspects are different from each other and emphasize distinct aspects of the research work. In the following, we first describe the setup of the experiments and then move on to the empirical analysis.

6.6.1 Experimental Setup

Article collections We used two data sets for the experiments in this section. The first one, denoted ICLR, is of articles crawled from the OpenReview website.⁹ Specifically, we collected 5,595 articles submitted to the ICLR conference between the years 2017-2020. For the articles in ICLR’2017, we also obtained citation counts (487 articles).¹⁰ The second data set, denoted ACL, is of articles in the natural language processing domain. To build this collection, we crawled papers from the ACL Anthology that were published up to October 2018, resulting in 40,376 articles.¹¹ For the ACL articles, we also obtained citation counts by analyzing the citations between pairs of articles within the collection. In both data sets, we extracted the text from the PDF files and used the concatenation of the title, abstract, and introduction as the article’s representation.

Query sets For each article collection, we also generated a set of queries as follows. Using the list of all titles in each data set separately, we extracted meaningful n-gram phrases using the AutoPhrase [176, 177] tool and the top 100 phrases were used as queries. Some examples for queries that were generated for the ACL data set include “active learning”, “anaphora resolution”, and “aspect-based sentiment analysis”; we made the full set of queries publicly available.¹²

Implementation details Both queries and articles were pre-processed including stop-words removal and lemmatization using a WordNet Lemmatizer. To perform a relevance-based initial retrieval for all queries, we used the cosine similarity between the *tf.idf* vectors

⁹<https://openreview.net> (accessed August 25, 2021)

¹⁰<https://github.com/Chillee/OpenReviewExplorer> (accessed August 25, 2021)

¹¹<https://www.aclweb.org/anthology> (accessed August 25, 2021)

¹²<https://github.com/saarku/fig-explorer/tree/master/queries> (accessed August 25, 2021)

of the query and the articles. To generate aspect scores for articles in each data set, we leveraged the best performing topic models from the previous section that were learned using a subset of the ICLR data (note that this is only a small portion of ICLR for which we had the full annotations available to us; for more details please refer to Section 6.4.1). We only used four review aspects that resulted in the highest performance according to Section 6.5 including Clarity, Originality, Soundness, and Substance. To quantitatively measure the ranking performance of using aspect scores, we calculated the Normalized Discounted Cumulative Gain (*ndcg*) using citation counts. To calculate *ndcg* using citations, we assigned the label 0 to articles with no citations, and the labels 1, 2, 3, 4 to articles with citation counts in $[1, 5]$, $[6, 10]$, $[11, 20]$, and $[21, \infty]$, respectively.

6.6.2 Empirical Results

The effectiveness of aspect ranking First, we are interested in examining the effectiveness of sorting a result list using scores in a specific review aspect. In Table 6.11, we report the performance of ranking based on relevance and different review aspects for both data sets. For the ranking based on review aspect scores, we first perform relevance-based retrieval to obtain 50 articles which we then re-rank using aspect scores.

Table 6.11: The effectiveness of ranking a search result list using aspect scores (based on citation counts). Statistically significant differences with Relevance are marked with ‘*’ (*pval* < 0.05 according to a two-tailed paired t-test).

	ICLR		ACL	
	<i>ndcg@5</i>	<i>ndcg@10</i>	<i>ndcg@5</i>	<i>ndcg@10</i>
Relevance	.325	.330	.156	.207
Clarity	.367	.389*	.243*	.288*
Originality	.341	.359	.118	.149*
Soundness	.405*	.412*	.138	.194
Substance	.507*	.468*	.113*	.153*

According to the results, we can see that the ranking based on aspect scores can significantly improve the quality of the result list from the perspective of citation counts. Specifically, in the case of ICLR, for all review aspects, an improvement is observed compared with relevance ranking in terms of *ndcg@5* and *ndcg@10*; most of the improvements are statistically significant. In the case of ACL, there are performance improvements only when using the Clarity dimension. Furthermore, in the Soundness and Substance dimensions, we can see that aspect score ranking does not decrease the performance of relevance ranking

to a statistically significant degree. This finding is important since we will later show that the highly-ranked documents are substantially different in the different ranking approaches. The difference in the performance of the two data sets is likely since our aspect score models were learned using a subset of the ICLR data set that contains articles that can be different than those in ACL (in terms of the format and topics, for example). Still, it is interesting to see that for some dimensions (e.g., Clarity), we were able to successfully leverage a model that was learned on a different domain, showing that some aspect models can be shared across domains. Finally, the results in Table 6.11 also suggest that aspect scores can potentially be used as a substitute for citation counts. This is important as it can help identify high-quality articles with possibly low citation counts which is a limitation of the current literature search systems.

Quantitative analysis of the result lists Our analysis also reveals that the top-ranked documents based on aspect scores and based on relevance can be substantially different. First, we quantitatively compare the vocabulary of the top-ranked documents according to the different approaches in Table 6.12. To perform this analysis, we calculated a weighted mean of the *tf.idf* vectors of the top-10 documents according to each approach (the weights were set based on the reciprocal rank). Then, we selected 50 terms with the highest scores in the aggregated vector. Finally, we calculated the Jaccard index between two sets of terms and reported the average over queries.¹³

Table 6.12: The Jaccard index between groups of words representing the top-ranked articles in different ranking approaches.

	ICLR					ACL				
	Rel.	Cla.	Ori.	Sou.	Sub.	Rel.	Cla.	Ori.	Sou.	Sub.
Clarity	.196	—	.177	.198	.237	.186	—	.123	.153	.145
Originality	.171	.177	—	.238	.312	.203	.123	—	.273	.352
Soundness	.196	.198	.238	—	.225	.210	.153	.273	—	.280
Substance	.192	.237	.312	.225	—	.205	.145	.352	.280	—

According to the results, the vocabulary overlap between the different ranking approaches is low for both data sets; specifically, for every pair of approaches, the Jaccard index is lower than 0.4. The results also show that most aspects have a similar level of low overlap with relevance ranking. This finding demonstrates the extent to which all aspects deviate from the initial ranking. Furthermore, we can see that the highest values are obtained for the overlap of two aspect scores (as opposed to the overlap of an aspect score and relevance ranking).

¹³Jaccard index is a measure for the overlap between two sets and can get values between 0 and 1.

This result suggests that there are correlations between different review aspects. This is also consistent with the results in the previous section that demonstrated the effectiveness of the Multi-view approach, which leverages representations that were learned using several aspects.

Table 6.13: The average Jaccard index between the top-ranked articles in different ranking approaches.

	ICLR					ACL				
	Rel.	Cl.	Ori.	Sou.	Sub.	Rel.	Cl.	Ori.	Sou.	Sub.
Clarity	.131	–	.135	.137	.240	.091	–	.006	.078	.040
Originality	.121	.135	–	.163	.271	.158	.006	–	.192	.293
Soundness	.106	.137	.163	–	.164	.127	.078	.192	–	.225
Substance	.113	.240	.271	.164	–	.134	.040	.293	.225	–

To further support the findings in Table 6.12, we also calculated the Jaccard index between the sets of top documents in each approach (using the document identifiers). The results are presented in Table 6.13 and show that there is also low overlap between the result lists at the document level; the actual values are even lower than in Table 6.13. The findings regarding the overlap of result lists are important especially in light of the results in Table 6.11 which showed that aspect scores sometimes yield similar performance to relevance ranking (in terms of citation counts). Thus, the conclusion based on the overlap analysis is that using aspect scores, we can still promote high-quality information (in terms of citations) but focusing on specific aspects of the research work.

Qualitative analysis of result lists To further shed light on the differences between the rankings of the different aspect scores, we conducted a qualitative analysis. In Table 6.14, we present the 10 most representative terms according to each aspect (and relevance ranking as well) for two queries from the ICLR data set (“Question Answering” and “Self Attention”); the terms were selected using the same procedure as in the analysis in Table 6.12.

Focusing on the query “Question Answering”, we can see the difference in the vocabulary of the different approaches. First, when using relevance ranking, we observe general terms that are relevant to the topic such as “model”, “sentence”, and “paragraph”. Moving on to the aspect-based ranking, we can see that each aspect reveals a different theme. For example, in the Originality aspect, we can see terms such as “agent”, “student”, and “teacher” which are in the theme of information seeking conversations, thus suggesting high originality of papers in this topic; this indeed makes sense as this is a topic that gained popularity in the recent years thus with good opportunity for conducting very original research. In the Clarity

dimension, on the other hand, we can see terms such as “regression”, “classification”, and “extraction”. These are more general terms that are prevalent in papers with good Clarity.

Moving our focus to the second query, “Self Attention”, we can also see that using different review aspects can help to focus on different themes of a given research topic. For example, representative words in the Substance aspect include “NLP”, “translation”, and “phrase”. These terms suggest that the substance of NLP-related articles on this topic is very good. This is in contrast to the terms of relevance ranking (such as “transformer”, “context”, and “head”) that are used often in articles on this topic.

Table 6.14: Representative terms in the different review aspects for two queries in the ICLR data set. Boldface: a unique word for an aspect.

Question Answering				
Relevance	Clarity	Originality	Soundness	Substance
question	question	question	reasoning	question
answering	answering	answer	question	answer
topic	span	ask	answering	passage
answer	text	asking	visual	QA
generation	reasoning	dialogue	retrieval	MRC
compound	task	learner	answer	answering
model	regression	teacher	task	conversational
reasoning	classification	student	network	cotton
sentence	extraction	agent	model	comprehension
paragraph	inductive	passage	text	reading

Self Attention				
Relevance	Clarity	Originality	Soundness	Substance
attention	self	attention	attention	attention
self	attention	spatial	self	phrasal
layer	training	frame	convolution	token
convolution	data	GT	spatial	mechanism
sequence	supervision	temporal	frame	phrase
transformer	semi	self	GT	translation
model	supervised	mechanism	temporal	model
mechanism	sequence	model	video	alignment
context	GAN	memory	lstm	machine
head	MUSE	video	equivariant	NLP

Analysis of rejected papers The ICLR data set provides information on whether an article was accepted to the conference. In the following analysis, we use this information to

further shed light on the effectiveness of using review aspect scores for re-ranking a result list of research articles. Specifically, we examine the top-5 documents based on relevance ranking and aspect score ranking in Table 6.15. In the table, we report the average number of rejected articles (# Rejected), the average number of citations of rejected articles (# Citations), and the number of queries (out of 100) for which the top-ranked rejected article had more citations than the top-ranked accepted article (# Queries).

Table 6.15: Analysis of rejected papers: (1) # Rejected: the number of rejected papers. (2) # Citations: the number of citations of rejected papers. (3) # Queries: queries in which the top-ranked rejected paper has a higher citation count than the top-ranked accepted paper.

	# Rejected	# Citations	# Queries
Relevance	2.90	8.04	16
Clarity	1.92	9.28	12
Originality	2.59	7.46	23
Soundness	2.33	11.73	21
Substance	2.28	17.08	27

The results in Table 6.15 show that the average number of rejected articles at the top of the list is lower for all aspect scores compared with relevance ranking. This result demonstrates the ability of aspect scores to identify high-quality papers as attested by the acceptance decision. Furthermore, the results also show that the average number of citations of rejected articles at the top of the list is almost always higher than that of relevance ranking. That is, using aspect scores can be useful to locate well-cited articles that got rejected. Finally, we can see that also the number of queries for which a top-ranked rejected paper is cited more than a top-ranked accepted paper is almost always higher in the case of aspect scoring compared with relevance ranking. To conclude, the results in Table 6.15 demonstrate the ability of aspect scores to improve the quality of the result list by decreasing the number of rejected papers that are shown to the user and presenting rejected articles with high citation counts.

Examples of rejected papers In Table 6.16, we provide a few examples of highly cited rejected articles that appeared at the top-5 results according to aspect scores and did not appear at the top results of relevance ranking.¹⁴ The examples in Table 6.16 further demonstrate the ability of aspect score ranking to identify rejected papers with high impact in practice. In the first query, “Latent Variable”, the top-ranked rejected article is about an optimization technique for neural networks. A possible explanation for this is the relative

¹⁴The citation information in Table 6.16 is based on Google Scholar (accessed August 25, 2021).

simplicity of the approach as also mentioned in the reviews for this paper.¹⁵ In the second query, “Model Compression”, we can see that one of the top results for the query when ranking based on Originality is the article about the SqueezeNet architecture which is well-known in the computer vision domain. It is interesting that this paper got ranked high by the Originality dimension given the meta-review of the paper which stated that “The novelty of the submission is very limited”.¹⁶ A possible reason can be that this article is “original” from the engineering perspective which is supported by its wide adoption in practice.

Table 6.16: Examples of well-cited rejected papers that are highly-ranked by aspect scores.

	Query	Citations	Title
Clarity	Latent Variable	349	Adding Gradient Noise Improves Learning for Very Deep Networks
Originality	Model Compression	4,218	SqueezeNet: AlexNet-level Accuracy with 50x Fewer Parameters and <0.5MB Model Size
Soundness	Self Attention	1,099	ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation
Substance	Adversarial Examples	1,942	Conditional Image Synthesis with Auxiliary Classifier GANs

6.7 GENERATING EXPLANATIONS FOR AUTOMATICALLY GENERATED QUALITY SCORES

To fully integrate the proposed assessment models in real-life literature systems, it is crucial to supplement the predicted quality scores with concrete explanations on why the algorithm chose a specific grade. Providing such explanations for the scores can improve their usability to researchers in at least three ways. First, the explanations have the potential of increasing the trust of users in the system’s quality assessment algorithm. Second, they can give researchers concrete recommendations to improve their work. Finally, providing explanations could potentially accelerate the digestion of research articles by enabling users to quickly focus on the specific strengths and weaknesses of a paper.

In Section 6.5.3, we proposed one approach to using topic models for explaining the generated quality scores. Specifically, our idea was to present the user with the topics that are most correlated with a specific grade. Topics have a clear advantage when using this approach compared with other types of machine learning models (e.g., neural networks) since they rely on clusters of words that can be very interpretable.

¹⁵<https://openreview.net/forum?id=rkjZ2Pcxe> (accessed August 25, 2021)

¹⁶<https://openreview.net/forum?id=S1xh5sYgx> (accessed August 25, 2021)

An alternative way to generate quality score explanations using topic models can be to highlight parts of the article that are indicative of a specific score. A possible implementation for this approach could potentially leverage the generative mechanism of probabilistic topic models. Specifically, it is possible to find the most correlated topics with a specific score and then calculate the likelihood that article segments were generated from them. A clear advantage of this approach over showing only the correlated topics is that it provides concrete explanations for the paper at hand and does not require extra effort from the user in identifying the actual content that explains the score.

Finally, developing a separate model for generating quality score explanations is another possible approach for addressing this problem. For example, one work proposed to leverage knowledge graphs to generate reviews for articles which can be used for generating explanations for quality scores [48]. Other approaches that leverage retrieval of reviews/comments and sequence-to-sequence models can also be used for addressing this problem. In this thesis, our focus is on studying the effectiveness of topic models for the task and for improving literature search engines. We thus leave the study of generating quality score explanations for future work.

6.8 CONCLUSIONS

We studied the problem of automated assessment of articles and explored features for the task based on topic models. We proposed multiple ways to construct topical features and combine them. Evaluation results on two data sets demonstrated the effectiveness of using topic models for the task compared with a bag-of-words baseline and neural network-based representations. Furthermore, the use of labels to guide the extraction of topics is effective and in general, combining topics learned using multiple views of the text data appears to be the most effective and robust. The results also showed that combining topics with unigrams can be effective for some of the review aspects and the combination of their prediction scores is usually the best approach for that. Finally, our empirical study demonstrated the potential effectiveness of using the generated scores for improving literature search.

The findings of our study can be used directly for improving the current methods for automated assessment of research articles, thus potentially helping improve and speed up the reviewing process and accelerating research in general. Furthermore, the generated scores can be used to enhance the performance of the current literature search systems by supporting novel functions such as sorting, filtering, and analysis of the result list; in Chapter 7, we discuss the implementation of a research literature search system that directly leverages those scores. Finally, as our approaches to topic feature construction and combination are

all general, they can be used in any application problem involving the use of text data for predictive modeling.

This work can be extended in multiple directions. First, one direction is to generalize the different findings in the work for text classification tasks in general. Another direction for future work is the development of applications and data mining techniques that leverage the predicted aspect scores to facilitate the work of researchers. The ultimate evaluation of the proposed technique will have to be done by building and deploying an intelligent assessment tool in a real literature system and obtain feedback from researchers.

CHAPTER 7: ACADEMIC-EXPLORER: A SYSTEM FOR RETRIEVAL AND EXPLORATION OF RESEARCH ARTICLE COLLECTIONS

To facilitate the study of the potential impact of the approaches proposed in this thesis on literature systems, we developed `AcademicExplorer`, a novel system for the search and exploration of collections of research articles. The system integrates all three lines of research of this thesis and its main novelties are as follows: (1) Supporting the search for figures using keyword queries where the user can select the different textual fields to represent a figure and the retrieval model (based on the ideas discussed in Chapter 3). (2) Supporting various novel exploration functions that we implemented using the figure embedding approach discussed in Chapter 4. (3) Supporting article search where review aspect scores can be used for sorting of the result list as discussed in Chapter 6. (4) Supporting collaborative query construction to improve the accuracy of poor-performing queries as discussed in Chapter 5. (5) The system can facilitate the data collection process to create realistic data sets for the figure retrieval and recommendation tasks and is open-source to support future research on the topic.

7.1 INTRODUCTION

The current literature search systems, such as Google Scholar and Microsoft Academic, have several limitations worth addressing to increase their utility for researchers. Improving various components of those systems would potentially facilitate the work of researchers and accelerate scientific discoveries.

One of the limitations of those systems is that they do not leverage any content analysis to assess the quality of research articles and rely mainly on citation counts. Automating the assessment of research articles using text data can benefit academic search engines in many ways. For example, automatically generated scores in different quality aspects can be used to sort a result list to help discover articles with strengths in various dimensions of the research work. Furthermore, the scores can also help to assess the quality of newly published articles (with low citation counts) and articles in pre-print repositories.

Another limitation of the current systems is that the retrieval units are mostly research articles. Research articles, however, contain different elements that carry unique information that can be valuable for researchers. Research article figures, which we studied in this thesis, are examples of such elements. Figures, for instance, can often be used for understanding the methods used in an article and digesting the experimental findings. Thus, having systems that can support the exploration of research collections using article figures would facilitate

the work of researchers. Such systems can help researchers digest the knowledge buried in the literature quickly, thus accelerating scientific discovery and technology innovation. In the most basic mode of exploration, users can retrieve figures using keyword queries. To this end, several systems for figure retrieval were developed [52, 83, 84]. Using retrieval only, however, may not be sufficient for the effective exploration of research figure collections. Furthermore, the development of figure mining and retrieval algorithms is impeded currently due to the lack of test collections and training data. Thus, developing a system to facilitate the collection of such data can be of merit.

The current academic search systems also do not provide much support for users to construct queries. Addressing this limitation is crucial since researchers often would find it challenging to formulate effective search queries. One of the reasons for this can be because researchers often explore new research topics and areas, resulting in a significant vocabulary gap between their search queries and the relevant articles. In such cases, when using the current systems, users would end up reformulating their queries many times with little support from the system.

To address those limitations, we leveraged the three lines of research results of this thesis to develop a novel system, **AcademicExplorer** (<http://academicexplorer.web.illinois.edu>), for search and exploration of research collections. First, **AcademicExplorer** allows users to directly retrieve figures of research articles using keyword queries. Then, using each retrieved figure as a seed, the user can further explore the collection and refine the search. Specifically, the user can view *related figures* from other articles or the same article. The user can also explore the general topic of the figure by viewing *clusters of figures* constructed from the citation network of the figure. Finally, the user can select any figure to be used for *re-ranking* the result list. We implemented the different exploration functions using *figure embeddings* learned from the text data representing the figure with a neural network as discussed in Chapter 4. To train the model, we used a weak supervision approach that leverages the citation connections between articles. For the figure retrieval part, we implemented multiple ways to represent figures with text data, which the user can experiment with, and several ranking algorithms.

AcademicExplorer also supports research article search and users can sort the result list using different *academic review aspects* such as clarity, originality, and soundness. To generate the scores in each aspect, we leveraged the topic modeling approach studied in Chapter 6. The scores in the different dimensions are also presented to the user next to each article to highlight its strengths and weaknesses. Furthermore, the system also includes a tool to assist users in *formulating queries*. This tool can be used by researchers to improve ineffective queries through collaboration with the system as described in Chapter 5.

The system also supports the *annotation* of relevance judgments by allowing a user to make such judgments on the retrieved figures and articles using radio buttons. *Implicit feedback*, which can be used for model training and evaluation, can also be easily collected based on user actions in the system.

AcademicExplorer is an open-source toolkit that was designed to facilitate future research on the topic.¹ Specifically, the toolkit can be used by other researchers for (1) learning about the state-of-the-art figure retrieval methods and new functions supported by embeddings, (2) building test collections, (3) testing interactive approaches, (4) building their applications, and (5) developing and testing new models.

7.2 RELATED SYSTEMS

The currently major academic search engines, including Google Scholar [2], Microsoft Academic [3], and Semantic Scholar [4], have limited support for query construction and content-based assessment of research articles. These systems support query construction mostly in the form of query suggestion and auto-completion. Different from those systems, **AcademicExplorer** has the “Help Me Search” feature to actively support users in the process of constructing an effective query. Most of the current engines also do not use content analysis for the quality assessment of papers and mostly rely on citation counts. One exception for this is the Microsoft Academic system that offers the option to sort the result list based on a predicted score for the article’s saliency.

The major commercial search engines for research articles do not support the search for figures of research articles. For this reason, several figure search engines were previously developed, mainly for the biomedical domain [52, 83, 84, 85]. The BioText engine [83] enables the search of figures using keyword queries. In another system [85], figure search was also implemented with some basic exploration capabilities. The SLIF system [84] also performed figure retrieval but proposed a topic model approach to browse the result list. Finally, the FigSearch retrieval system [52] was tailored for the use-case of gene-related figures. Compared with previous systems, **AcademicExplorer** is the first open-source general system that supports figure retrieval and exploration. Specifically, the system includes several novel functions, which utilize embeddings, that can be used to perform exploration of the collection. Furthermore, the system is general enough to facilitate future research on the topic.

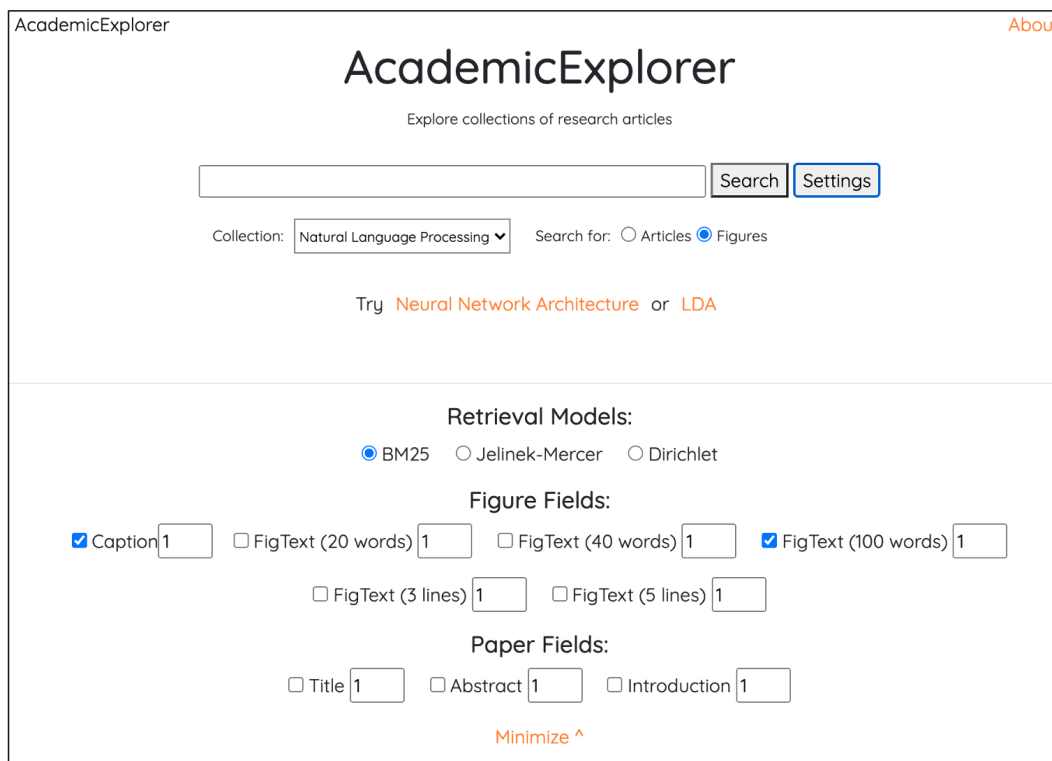
Other systems focused on the extraction, summarization and indexing of figures [178, 179, 180, 181]. In our system, the focus is on the search and exploration of figure collections, while

¹<http://github.com/saarku/fig-explorer> (accessed August 25, 2021)

we use existing tools for the extraction part. Thus, **AcademicExplorer** is complementary with these existing systems and they can be combined.

7.3 SYSTEM FUNCTIONS

We implemented **AcademicExplorer** as a Web application (the main page is presented in Figure 7.1). The most basic mode of exploration in the system is the retrieval of figures and articles using a keyword query. To search the collection, the user needs to type a query in the text box and select whether to search for figures or articles. The user can also select the search configuration (i.e., collection, model, and figure/article fields); if the user does not specify a configuration for the search, default settings are used.



The screenshot shows the main page of the AcademicExplorer web application. At the top left, the text "AcademicExplorer" is displayed. At the top right, there is a link labeled "About". The main heading is "AcademicExplorer" in a large font, with the subtitle "Explore collections of research articles" below it. A search bar is located in the center, with a "Search" button to its right and a "Settings" button to its left. Below the search bar, there is a dropdown menu for "Collection" currently set to "Natural Language Processing". To the right of the dropdown, there are radio buttons for "Search for:" with "Articles" unselected and "Figures" selected. Below this, there is a suggestion: "Try Neural Network Architecture or LDA". The page is divided into sections for configuration. The "Retrieval Models:" section has three radio buttons: "BM25" (selected), "Jelinek-Mercer", and "Dirichlet". The "Figure Fields:" section has several checkboxes with associated input boxes: "Caption" (checked, 1), "FigText (20 words)" (unchecked, 1), "FigText (40 words)" (unchecked, 1), "FigText (100 words)" (checked, 1), "FigText (3 lines)" (unchecked, 1), and "FigText (5 lines)" (unchecked, 1). The "Paper Fields:" section has three checkboxes with associated input boxes: "Title" (unchecked, 1), "Abstract" (unchecked, 1), and "Introduction" (unchecked, 1). At the bottom center, there is a "Minimize ^" link.

Figure 7.1: The main page of **AcademicExplorer**.

After issuing the query, the user can view a search result page with up to 10 articles/figures (see Figures 7.2 and 7.3). At the top of the search result page, the user can choose to sort the result list based on different criteria including the citation count and different review aspects; in the case of figures, the user can sort only based on the number of citations. The top of the page also includes the “Help Me Search!” button. Clicking on this button, the user would be presented with terms that can be added to the query to improve the results.

For each article in the result list, we can first see its title, which also serves as a hyperlink to the article's PDF. Below, the user can mark if the article is relevant to the query and view its citation count and figures (the user can initiate a search for figures similar to any of the article figures). Following this, the user can also view the abstract of the article. Finally, the system displays the scores of the article in the different review dimensions using colors (we use ten colors ranging from dark red through yellow and ending with dark green).

For each figure in the result list, the system presents the caption in the first line (also serves as a hyperlink to the article). Below the caption, the user can indicate whether the figure is relevant to the query by clicking on the corresponding radio button. Next to those radio buttons, the user can view the citation count of the paper and use some buttons to perform further exploration of the collection. Finally, the user can view the image file of the figure and a short textual summary of the figure, which is automatically extracted from the article by the system.

The screenshot shows the AcademicExplorer interface. At the top, there is a search bar containing 'Neural Network Architecture' and buttons for 'Search' and 'Settings'. Below the search bar, there are filters for 'Collection: Natural Language Processing' and 'Search for: Articles (radio), Figures (radio checked)'. A suggestion bar shows 'Try Neural Network Architecture or LDA'. On the right, there is a 'Sort by: Relevance' dropdown and 'Apply' and 'Help Me Search!' buttons.

The first result is a figure with the caption 'Figure 1: The convolutional recurrent neural network architecture.' It has radio buttons for 'Relevant' and 'NOT Relevant', and is cited by 13. It includes links for 'Paper Info', 'Same Paper Figures', and 'Re-rank using this figure'. The abstract text reads: 'Therefore, we have mainly focused on recurrent **networks** in this paper. This section gives a description of the recurrent **neural network architecture** that we have used for the essay scoring task and the training process. The **neural network architecture** that we have used in this paper is illustrated in **Figure 1**. We now describe each layer in our **neural network** in detail. Lookup Table Layer: The first layer of our **neura** ...'

The second result is a figure with the caption 'Figure 2: Our neural network architecture.' It has radio buttons for 'Relevant' and 'NOT Relevant', and is cited by 251. It includes links for 'Paper Info', 'Same Paper Figures', 'Related Figures', 'Citation Clusters', and 'Re-rank using this figure'. The abstract text reads: 'In this section, we first present our **neural network** model and its main components. Later, we give details of training and speedup of parsing process. **Figure 2** describes our **neural network architecture**. First, as usual word embeddings, we represent each word as a d-dimensional vector $e \in \mathbb{R}^d$ and the full embedding matrix is $E \in \mathbb{R}^{d \times N}$ where N is the dictionary size. Meanwhile, we also map POS tags and arc labels to a d-dimensional vector ...'

The third result is a figure with the caption 'Figure 1: Neural network language model architecture.' It has radio buttons for 'Relevant' and 'NOT Relevant', and is cited by 251. It includes links for 'Paper Info', 'Same Paper Figures', 'Related Figures', 'Citation Clusters', and 'Re-rank using this figure'. The abstract text reads: 'This section describes a general framework for feedforward NNLMs. We will follow the notations given in (Schwenk, 2007). **Figure 1** shows the **architecture** of a **neural network** language model. Each word in the vocabulary is represented by a N dimensional sparse vector where only the index of that word is 1 and the rest of the entries are 0. The input to the **network** is the concatenated discrete feature representations of n-1 previous words (history), i ...'

Figure 7.2: Search result page example (figures).

The user can perform further exploration of the collection using buttons that appear for

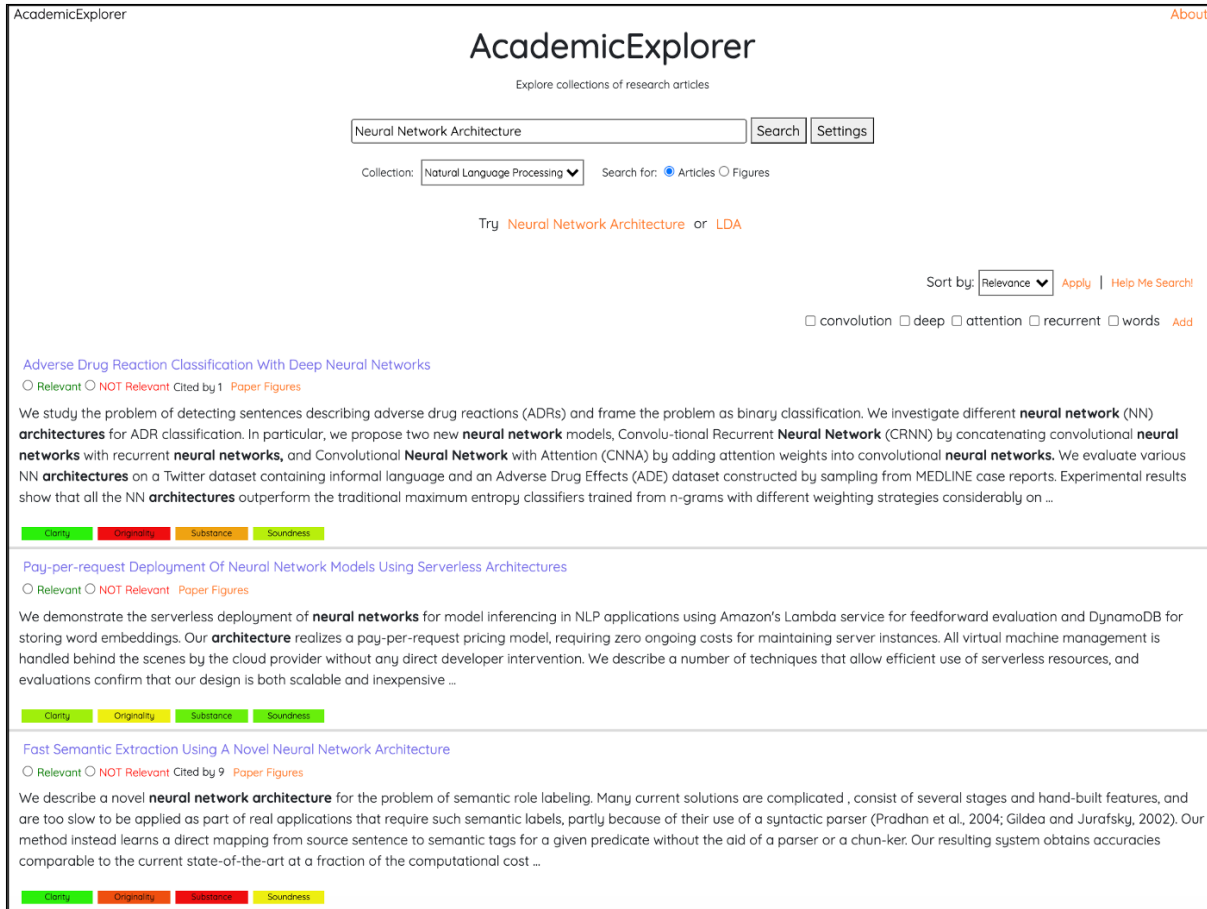


Figure 7.3: Search result page example (articles).

each figure. The system presents the output of each exploration function below the figure, and it can be removed by either pressing the button again or using the “Minimize” button. A screen-shot of a single figure, after two exploration functions were used, is presented in Figure 7.4. The exploration functions include:

1. Paper Info: allowing the user to view information about the article containing a figure such as its title and abstract. The user can also issue a search for similar articles using the “Search using this paper” option.
2. Same Paper Figures: displaying other figures in the same article.
3. Related Figures: presenting the user semantically related figures.
4. Citation Clusters: enabling a broader exploration of the topic of the figure by clustering the citation network of the figure and presenting representative figures for each cluster.
5. Re-rank using this figure: re-ranking the existing result list using the embedding-based

representation of the specific figure; this option exists also for the related figures and the figures of the citation clusters.

Figure 1: Various neural network architectures.

○ Relevant ○ NOT Relevant Cited by 1 [Paper Info](#) [Same Paper Figures](#) [Related Figures](#) [Re-rank using this figure](#)



In this section, we introduce a number of **neural network architectures** and propose two new models, Convolutional Recurrent **Neural Networks** (CRNN) and Convolutional **Neural Network** with Attention (CNNA) 5. Deep Convolutional **Neural Networks** (CNN)s are recently extensively used in many computer vision (Alex Krizhevsky et al, 2012;Szegedy et al, 2014;Simonyan and Zisserman, 2014;He et al, 2015) and NLP tasks. In NLP, CNNs (**Figure**

Adverse Drug Reaction Classification With Deep Neural Networks

[Search using this paper](#)

We study the problem of detecting sentences describing adverse drug reactions (ADRs) and frame the problem as binary classification. We investigate different neural network (NN) architectures for ADR classification. In particular, we propose two new neural network models, Convolutional Recurrent Neural Network (CRNN) by concatenating convolutional neural networks with recurrent neural networks, and Convolutional Neural Network with Attention (CNNA) by adding attention weights into convolutional neural networks. We evaluate various NN architectures on a Twitter dataset containing informal language and an Adverse Drug Effects (ADE) dataset constructed by sampling from MEDLINE case reports. Experimental results show that all the NN architectures outperform the traditional maximum entropy classifiers trained from n-grams with different weighting strategies considerably on both datasets. On the Twitter dataset, all the NN architectures perform similarly. But on the ADE dataset, CNN performs better than other more complex CNN variants. Nevertheless, CNNA allows the visualisation of attention weights of words when making classification decisions and hence is more appropriate for the extraction of word subsequences describing ADRs.

[Minimize ^](#)

Related Figures:



Figure 3: Convolutional Neural Networks for orthographical feature extraction. Only the first convolutional layer and its following max-pooling layer are presented. [\[Re-rank using this figure\]](#)



Figure 3: Our architecture for VQA: Multimodal Compact Bilinear (MCB) with Attention. Conv implies convolutional layers and FC implies fully connected layers. [\[Re-rank using this figure\]](#)



Figure 1: Illustration of an example term pair relation classification using convolutional neural networks. [\[Re-rank using this figure\]](#)



Figure 1: Alternative neural composition architectures for error detection. a) Convolutional network b) Deep convolutional network c) Recurrent bidirectional network d) Deep recurrent bidirectional network. [\[Re-rank using this figure\]](#)

[Minimize ^](#)

Figure 7.4: Example of a single search result after two exploration functions were used.

AcademicExplorer also logs information about user actions. This information includes:

1. Clicks on a caption/title of a result figure/article.
2. Clicks on the relevant/not-relevant button.
3. Events of issuing a query or of using any of the exploration functions.

The logging of such information can be useful for creating a test collection for different tasks such as figure retrieval (relevance between a query and a figure) and figure relatedness prediction (relevance between two figures).

7.4 IMPLEMENTATION

The high-level architecture of AcademicExplorer is presented in Figure 7.5. AcademicExplorer is an open-source toolkit that was designed to be flexible enough for future extensions. Next, we describe the front-end and back-end of the system.

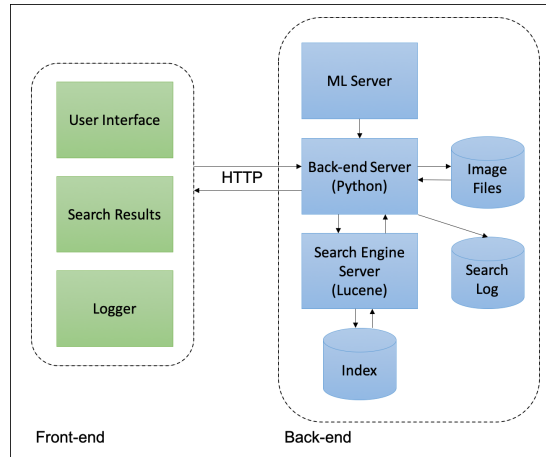


Figure 7.5: System Architecture.

7.4.1 Front-end

AcademicExplorer is a Web application that is written in JavaScript using jQuery and Ajax. The front-end has three main functionalities:

1. User Interface: obtains the user input and sends it to the server. The input usually includes the query and the search settings in the case of retrieval and the figure ID in the case of an exploration function.
2. Search Results: receives the search results (or an exploration function output) from the server and presents them in the browser.
3. Logger: collects user interactions from the browser.

7.4.2 Back-end

The back-end of the system is composed of the following components:

1. A Back-end Server: written in Python using the Flask library.²
2. A Search Engine Server: written in Java using the Lucene library.³
3. A Machine Learning (ML) Server: written in Python.
4. Index: a Lucene-based inverted index (one for figures and one for articles).

²<https://flask.palletsprojects.com/en/2.0.x/> (accessed August 25, 2021)

³<https://lucene.apache.org> (accessed August 25, 2021)

5. Image Files repository.
6. Search Log.

Back-end server The back-end server forms a bridge between the front-end and the search engine server. The main tasks of the back-end server are to handle search requests, perform aspect sorting, and execute the different exploration functions. The back-end server communicates with the search engine server to get the result list to complete a search request. To complete an exploration or sorting request, the back-end server obtains the necessary information from the ML server. Once the results are returned from either server, the back-end server performs some post-processing of the content and sends it to the browser. Some basic computations are also performed in the back-end server including bold-facing of the query terms in the snippets, fetching the image files of figures from the image files repository, and updating the search log.

Search engine server The main task of the search engine server is to perform retrieval using a keyword query. The search engine uses an index that stores the figures/articles in the collection using textual information. Specifically, a figure is represented in the index using multiple textual fields including its caption, text in the article that explicitly discusses the figure, and some of the article’s text. An article is presented using the title, abstract, and introduction. The index also stores, for every figure, the image file directory which can be used to get an image from the repository. To assign a score to an element given a query, the linear interpolation of the scores of the query in the textual fields is computed; the weights in the interpolation can be controlled at the settings of the system (see Figure 7.1). The search engine server is also responsible for generating expansion terms for the “Help Me Search!” function for collaborative query construction; for details regarding the implementation of this approach, please refer to Chapter 5. The search engine server and the back-end server communicate using the py4j library.⁴

ML server The ML Server is responsible for learning the embedding-based representation of figures and utilizing it for the different exploration functions. Another responsibility of this server is to learn automatic assessment models for the articles and infer the scores for the articles in the result list.

Embedding-based representation of figures is learned using the caption of the figure as well as text in the article that directly describes the figure (both are concatenated). The

⁴<https://py4j.org> (accessed August 25, 2021)

textual information of a figure is fed into an LSTM network whose output serves as a vector representation for the figure. We use the binary Cross-Entropy loss function (i.e., the goal is to predict whether two figures are related or not). Weak supervision is used to build the training data by assuming that two figures in citing articles or in the same article are related; negative examples are obtained using random sampling. The model is learned by leveraging all figures in the collection using the TensorFlow library.⁵ (For more details about this approach, please refer to Chapter 4.)

The outputs for the “Related Figures” and “Citation Clusters” functions are computed in an offline manner, using the embedding vectors, and are stored for fast serving. To find related figures, for each figure in the collection, a KNN search is performed to find the most similar figures using the cosine similarity. To perform the citation-based clustering for a figure, a citation network of a figure is constructed first. To do that, all figures in articles that have a citation relation with the article of the seed figure (a direct connection or a connection through a third article) are included. Then, K-means clustering is performed using the embedding vectors. Finally, a representative figure is selected for each cluster based on the distance to the cluster’s mean. The “Re-rank using this figure” function is performed in an online manner. First, 100 figures are retrieved using the keyword query. Then, these figures are re-ranked based on their similarity with the seed figure in the embedding space.

To learn the models for automatic quality assessment, we use the topic modeling approach as in Chapter 6. Specifically, we learn models using articles from the ICLR conference for which we had the full annotations available. We then infer, in an off-line manner, the scores for articles in our indexed collections. The sorting of the result list based on those scores is performed by the back-end server.

7.5 DATA SETS

Two research collections are currently used in the system:

1. Natural Language Processing: 40,367 articles (73,409 figures) whose copyright belongs to ACL up to October 2018.⁶
2. Mechanical Engineering: 1,377 articles (9,712 figures) on bearing failures. This data set was created to explore the potential use of **AcademicExplorer** for supporting mechanical failure diagnosis where the analyst may conveniently retrieve figure plots showing

⁵<https://tensorflow.org> (accessed August 25, 2021)

⁶<https://aclweb.org/anthology> (accessed August 25, 2021)

typical vibration signal patterns for any hypothesized failure of a bearing (e.g., outer ring face) which can help finalize a diagnosis.

To build a collection of figures from the articles, we follow the next steps: (1) Obtaining a set of research articles. (2) Extracting the figures from the PDF files using the PdfFigures toolkit [101]. (3) Extracting the full text of the articles using the Grobid toolkit.⁷ (4) Processing the full text of the documents to extract the textual fields for a figure. (5) Indexing the figures using the textual fields. Currently, this is an offline pipeline that runs separately from the system. The pipeline is general enough to handle the indexing of any other data set of research articles where PDF files are available such as CORD-19 [182] and Arxiv [183], which we plan to add to the system in future work. Another direction for future work can be to integrate this pipeline into the system to support automated figure crawling and indexing.

7.6 APPLICATION SCENARIOS

The different functions of `AcademicExplorer` can be used to demonstrate different application scenarios. Here are a few examples of such demonstrations:

1. Sample applications of figure search: Users can input different queries to retrieve different types of figures such as figures illustrating technical approaches, experimental results, and illustrations of an example.
2. Exploratory search: The system can be used to support an exploratory search process. For example, a process of creating a literature review of a new topic using figures and articles.
3. Difficult queries: Users can use the “Help Me Search” function to help improve the performance of difficult queries.
4. Aspect sorting: Users can compare the results of using different sorting criteria to reveal different aspects of a research topic.
5. Exploration functions: The system can be used in cases where a keyword query is not a sufficient tool for satisfying the information need of the user and exploration functions can be used to improve the process.

⁷<https://github.com/kermitt2/grobid> (accessed August 25, 2021)

6. Comparison of different figure/article ranking algorithms: The system allows a user to easily configure the choices of the retrieval methods. The user can vary the configurations to compare different ways to represent figures/articles with text data and different ranking algorithms.
7. Collection of relevance judgments: The system can be used to collect users' queries and relevance judgments for building test collections.

7.7 CONCLUSIONS

In this chapter, we presented **AcademicExplorer**, a novel system for literature search and exploration. We leveraged the research results of this thesis to develop the system to address three main limitations of the current literature search systems, including (1) lack of support for figure search, (2) lack of support for interactive query construction, and (3) minimal content-based quality assessment.

There are different options for future work on this system. One direction is to add more functionalities to it. For example, implementing more exploration functions based on research figures, such as searching for research figures using an example figure, can further enhance the system. Creating an analysis module in which users can save figures and articles of interest to perform further analysis of them also has the potential of making the system more usable for researchers. Another possible direction for future work is to use the system to perform controlled user studies that would be useful to understand the effectiveness of different approaches and build test collections.

Finally, integrating **AcademicExplorer** with other literature systems such as BioMed Explorer [62] and the ACL Anthology search system [184] is also a direction worth exploring. Such integration can improve the utility of the system for researchers by combining the unique functionalities of each system (e.g., **AcademicExplorer** supports figure search while BioMed Explorer supports question answering). Another advantage of such integration would be the ability to further evaluate **AcademicExplorer** using researchers as users in a real-world application environment (e.g., **AcademicExplorer** includes the ACL collection that would be directly applicable to the case of the ACL Anthology search system).

CHAPTER 8: CONCLUSIONS

8.1 SUMMARY OF CONTRIBUTIONS

Intelligent scientific literature systems are crucial for conducting successful research. Improving those systems thus has the potential of accelerating scientific research and discovery in general. In this thesis, we focused on improving the existing literature search engines by addressing three of their limitations as follows:

1. The current systems only treat articles as the retrieval units.
2. The current systems do not support users in the construction of difficult search queries.
3. The current systems do not assess the quality of articles using content analysis.

In our first contribution of this thesis, we proposed to improve the current search engines using figures of research articles. Specifically, in Chapter 3, we introduced and studied the problem of figure retrieval from collections of research articles. The empirical results in that chapter demonstrated the effectiveness of using multiple textual sources (extracted from the research article of the figure) to represent a figure and of combining different retrieval models. Following that, we then further studied the problem of figure representation in Chapter 4 and proposed to use deep neural networks to that end. Our results in that chapter demonstrated the effectiveness of using embedding-based representation for figures and showed that weak supervision, using the citation network of papers, can be leveraged to do that. Our findings have the potential of improving the current scientific literature systems. Specifically, research figures can be used directly by systems to improve the search and mining of research articles.

The second research direction that we focused on in this thesis is optimizing the collaboration between the user and the system in literature search engines to improve the effectiveness of poor-performing queries. Specifically, in Chapter 5, we studied an interactive approach in which the system actively engages the user in the process of constructing a query. The experimental results, based on a simulated user study, demonstrated the effectiveness of the approach. The results also showed that minimal effort is required from the user to achieve good performance. The proposed interactive approach has the potential of improving the current search systems by improving those poor-performing queries. Furthermore, this approach can be integrated easily into any system as an optional feature (an “Help Me Search” button) that the user can use if needed.

In Chapter 6, we addressed the problem of automating the quality assessment of research articles. Automating the quality assessment of articles can help improve academic search

engines by adding quality signals in different aspects to enrich article representation. For example, the signals can be used by academic search engines to improve their ranking and support the analysis of the result list by users. Furthermore, the automated assessment can potentially speed up the research process by helping reviewers and providing feedback to authors at an early stage of the research. In this thesis, we studied the effectiveness of using topic model features for the task, which are more interpretable than the previously studied features. Our experiments demonstrated the merits of using topic models for the task. The results showed that using multiple views of the text data to learn the models and combining them using the prediction scores is the best performing approach. Finally, our study showed that the generated scores can improve the ranking of academic search engines thus motivating the use of those models in actual systems.

To facilitate further evaluation of the benefit of using the proposed techniques in real systems, we developed **AcademicExplorer**, a demo system that implements all the three lines of major ideas and approaches proposed in the thesis. Specifically, the system supports the search for figures, interactive query construction, and sorting of the result list using automatically generated quality scores.

One of the advantages of the approaches developed in this thesis is that they do not rely on massive amounts of data. It is crucial to consider this factor in the development of approaches for the research domain due to the limited amount of data available (e.g., user data and text). In this thesis, we addressed this challenge in different ways. For example, in Chapter 4, we leveraged the already existing citation network of papers to learn deep neural network models. In Chapter 5, we proposed an unsupervised approach for assisting users in constructing effective queries that only leverages information from the collection of documents and does not rely on query logs. Finally, we used topic models learned in an unsupervised manner from the existing research article collection to automate the assessment of research articles.

Some of the approaches developed in this thesis are also general enough to improve the performance of applications in other domains as well. Specifically, it is possible to use our collaborative query construction strategy in other applications as well. The idea of optimizing the collaboration between the user and the system can also be leveraged to improve other challenging tasks, such as the summarization of documents and recommending citations for articles. Another example is our approach for the automatic assessment of research articles that can also apply to other domains such as education and e-commerce.

8.2 DEPLOYMENT OF ACADEMIC-EXPLORER

The development of intelligent approaches and algorithms to support researchers is an

important topic to study. To make those developments usable by researchers, however, it is crucial to integrate the novel approaches into systems for research support. In this section, we discuss some of the options for making this integration.

Enhancement of general literature search engines The first option is to gradually enhance the current literature search engines. The algorithms developed in this thesis are all general and can thus be potentially directly integrated into a current general literature search engine, such as Google Scholar and Microsoft Academic, with new features. This approach would enable the maximum number of users to potentially benefit from those new algorithms. To demonstrate this idea, we developed `AcademicExplorer` as a standard search engine, with all of the existing features of the current literature search engines, and we added features such as figure search, interactive query construction support, and automatic assessment to complement it. One of the advantages of taking this approach is that existing systems could include novel features as optional first, which would lower the risk of such integration from a user experience perspective. The algorithms developed in this thesis can be integrated into existing systems in such a way as we illustrate in `AcademicExplorer`. Specifically, users can choose whether to search for articles or figures, query construction support would be provided to users only after pressing the “Help Me Search” button, and the ranking of articles using review aspect scores is performed only per the user request. Adding extra features to the existing systems also has advantages from a user adoption perspective. The main reason for this is that researchers are used to the interfaces, features, and modes of interaction in existing literature systems. Thus, adding new features to those systems might be a better option than developing an entirely new system, which will require researchers to learn how to use it first. Another consideration that needs to be taken into account in this type of integration is latency. When integrating the new functionalities into an existing system, those features should not increase the current latency significantly. For example, figure search was implemented in `AcademicExplorer` as efficiently as article search using a textual representation of figures and inverted index. Another example is our approach for interactive support for query construction, designed as an unsupervised approach, resulting in an efficient process of generating expansion terms. Finally, to integrate new components in existing systems as seamlessly as possible, further research should determine the actual interface design to be used. For example, should the predicted quality assessment scores be presented for each article and where? Should figures be mixed with articles in the result list or appear in a separate vertical? Thus, the integration of new features into existing systems in the general case should be based on user studies to determine the best configuration for that.

Integration with domain-specific literature search engines A domain-specific literature search engine may leverage special knowledge resources in a domain to provide more effective services to researchers. Integrating `AcademicExplorer` with such a search engine opens up interesting opportunities to leverage special knowledge resources in specific domains to further improve the algorithms proposed in this thesis. For example, `AcademicExplorer` can be integrated with the BioMed Explorer system [62] to leverage the complementary benefits of both systems (BioMed Explorer supports advanced question-answering techniques and `AcademicExplorer` supports some advanced retrieval techniques). Another potential integration can be done with the ACL Anthology search system [184] as `AcademicExplorer` already supports the ACL collection, which would enable the quick evaluation of the system by actual researchers.

Development of research task support systems A major limitation of all existing search engines is that they can only support search, but search is only a means to the end of finishing a task. The algorithms developed in this thesis have the potential to help support some of the research tasks more directly. For example, the quality assessment algorithm can be potentially useful for helping a researcher check the novelty of a research idea or get into a research field by finding articles that are easy to read (high clarity scores). Thus, another option is to develop a new system to support the work of researchers that would eventually replace the existing ones. The main idea is to have a system that would assist researchers in all different aspects of the research work. The system could then depart from the standard search engine paradigm to support researches using novel interaction options and machine learning to complete various research tasks. For example, this system could use our proposed automated assessment approach for analyzing the literature to suggest new research topics worth exploring or examine the novelty of a proposed idea. Some other examples for features that the system could support are citation management, article recommendation, and collaboration finding. The need for such a system is justified further by the different existing literature systems that support various parts of the research work. Some examples for such systems are Google Scholar (search), Mendeley (bibliography), ResearchGate (collaboration), and EasyChair (assessment). Thus, effectively combining those systems to create a unified tool would potentially benefit many researchers.

8.3 FUTURE WORK

Developing intelligent tools to assist researchers is important and can have a direct impact on our lives. In addition to the exploration of multiple ways to deploy `AcademicExplorer` as

discussed in the previous section, which should facilitate further evaluation of the proposed algorithms with real users and realistic user tasks, we also need to address many additional challenges that have not yet been fully addressed in this thesis, which we further elaborate below.

Limited data The availability of data sets in the scientific research domain is relatively low. This can be partially attributed to privacy and copyright consideration. The lack of data sets in this domain impedes the development of intelligent assistant systems for two reasons: (1) It is hard to evaluate the performance of novel methods and tasks. (2) Machine learning approaches require large amounts of data to be effective. In this thesis, we addressed this problem in different ways by modifying existing data sets for the evaluation of our tasks, user simulations, and weak supervision techniques. Still, there is much room for improvement in this direction. For example, efforts to collect and annotate research data can boost the research on many important topics. How to leverage the limited data to learn effective machine learning models is another possible avenue for future work.

Interface design In this thesis, we focused on the development of algorithms and approaches to assist researchers. A related direction to this, which is crucial as discussed in the previous section, is how to design an interface in the system that supports those new features. A potential direction for future work thus can be the study of different interface design options to maximize the utility of the proposed approaches.

User studies The study of novel tasks and interaction paradigms for literature search require user studies to evaluate their actual benefit. Since user studies are expensive to conduct, user simulation experiments can be used instead but are limited in their findings. User studies in this domain would be useful to understand user needs and to also collect data to build test collections for various tasks. The research domain has an advantage in this aspect since researchers can potentially test their tools (or by using their colleagues/students). Another interesting related direction for future work is the development of realistic user simulators to expedite the study of interactive approaches.

Explainable models The adoption of machine learning-based tools in the research domain is somehow limited by the lack of explainable machine learning approaches. For example, one of the possible reasons why systems for citation recommendation were not commercially adopted can be that the current systems do not provide an actual explanation of why a paper should be cited. Another example is the automated assessment problem studied

in this thesis. Such an approach would highly benefit from explanations for the different automatic quality scores.

Efficient, effective, and timely access to the scientific literature by researchers is crucial for accelerating scientific research and discovery. To this end, this thesis has developed multiple new general algorithms for intelligent assessment and retrieval of research content and the innovative `AcademicExplorer` system, which we hope serve as a small step toward the ultimate goal of developing and deploying an intelligent research task support system to enable all researchers to improve their productivity.

REFERENCES

- [1] L. Bornmann and R. Mutz, “Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references,” *Journal of the Association for Information Science and Technology*, vol. 66, no. 11, pp. 2215–2222, 2015.
- [2] *Google Scholar*, (accessed August 25, 2021). [Online]. Available: <https://scholar.google.com/>
- [3] *Microsoft Academic*, (accessed August 25, 2021). [Online]. Available: <https://academic.microsoft.com/>
- [4] *Semantic Scholar*, (accessed August 25, 2021). [Online]. Available: <https://www.semanticscholar.org/>
- [5] X. Li and M. de Rijke, “Do topic shift and query reformulation patterns correlate in academic search?” in *European Conference on Information Retrieval*. Springer, 2017, pp. 146–159.
- [6] S. Jones, S. J. Cunningham, R. McNab, and S. Boddie, “A transaction log analysis of a digital library,” *International Journal on Digital Libraries*, vol. 3, no. 2, pp. 152–169, 2000.
- [7] S. Kuzi and C. Zhai, “Figure retrieval from collections of research articles,” in *European Conference on Information Retrieval*. Springer, 2019, pp. 696–710.
- [8] S. Kuzi and C. X. Zhai, “A study of distributed representations for figures of research articles,” in *European Conference on Information Retrieval*. Springer, 2021, pp. 284–297.
- [9] S. Kuzi, A. Narwekar, A. Pampari, and C. Zhai, “Help me search: Leveraging user-system collaboration for query construction to improve accuracy for difficult queries,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 1221–1224.
- [10] D. Kang, W. Ammar, B. Dalvi, M. van Zuylen, S. Kohlmeier, E. Hovy, and R. Schwartz, “A dataset of peer reviews (peerread): Collection, insights and nlp applications,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 1647–1661.
- [11] S. Kuzi, W. Cope, D. Ferguson, C. Geigle, and C. Zhai, “Automatic assessment of complex assignments using topic models,” in *Proceedings of the Sixth ACM Conference on Learning@ Scale*, 2019, pp. 1–10.

- [12] S. Kuzi, C. Zhai, Y. Tian, and H. Tang, “Figexplorer: A system for retrieval and exploration of figures from collections of research articles,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 2133–2136.
- [13] *ACM Digital Library*, (accessed August 25, 2021). [Online]. Available: <https://dl.acm.org/>
- [14] *PubMed*, (accessed August 25, 2021). [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/>
- [15] B. M. Hemminger, D. Lu, K. Vaughan, and S. J. Adams, “Information seeking behavior of academic scientists,” *Journal of the American Society for Information Science and Technology*, vol. 58, no. 14, pp. 2205–2225, 2007.
- [16] M. Färber, “The microsoft academic knowledge graph: A linked data source with 8 billion triples of scholarly data,” in *International Semantic Web Conference*. Springer, 2019, pp. 113–129.
- [17] *AMiner*, (accessed August 25, 2021). [Online]. Available: <https://www.aminer.org/>
- [18] *ResearchGate*, (accessed August 25, 2021). [Online]. Available: <https://www.researchgate.net/>
- [19] *Google Dataset Search*, (accessed August 25, 2021). [Online]. Available: <https://datasetsearch.research.google.com/>
- [20] *Mendeley*, (accessed August 25, 2021). [Online]. Available: <https://www.mendeley.com/>
- [21] T. Strohman, W. B. Croft, and D. Jensen, “Recommending citations for academic papers,” in *Proceedings of the 30th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007, pp. 705–706.
- [22] S. Bethard and D. Jurafsky, “Who should i cite: Learning literature search models from citation behavior,” in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 2010, pp. 609–618.
- [23] W. Huang, S. Kataria, C. Caragea, P. Mitra, C. L. Giles, and L. Rokach, “Recommending citations: Translating papers into references,” in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2012, pp. 1910–1914.
- [24] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen, “Joint latent topic models for text and citations,” in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 542–550.

- [25] S. M. McNee, I. Albert, D. Cosley, P. Gopalkrishnan, S. K. Lam, A. M. Rashid, J. A. Konstan, and J. Riedl, “On the recommending of citations for research papers,” in *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*, 2002, pp. 116–125.
- [26] C. Jeong, S. Jang, E. Park, and S. Choi, “A context-aware citation recommendation model with bert and graph convolutional networks,” *Scientometrics*, vol. 124, no. 3, pp. 1907–1922, 2020.
- [27] Y. Wang, X. Liu, and Z. Gao, “Neural related work summarization with a joint context-driven attention mechanism,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1776–1786.
- [28] C. D. V. Hoang and M.-Y. Kan, “Towards automated related work summarization,” in *Coling 2010: Posters*, 2010, pp. 427–435.
- [29] Y. Hu and X. Wan, “Automatic generation of related work sections in scientific papers: An optimization approach,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1624–1633.
- [30] N. Agarwal, R. S. Reddy, G. Kiran, and C. Rose, “Towards multi-document summarization of scientific articles: Making interesting comparisons with scisumm,” in *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, 2011, pp. 8–15.
- [31] O. Yeloglu, E. Milios, and N. Zincir-Heywood, “Multi-document summarization of scientific corpora,” in *Proceedings of the 2011 ACM Symposium on Applied Computing*, 2011, pp. 252–258.
- [32] Q. Mei and C. Zhai, “Generating impact-based summaries for scientific literature,” in *Proceedings of ACL-08: HLT*, 2008, pp. 816–824.
- [33] J. Beel, S. Langer, M. Genzmehr, B. Gipp, C. Breitingner, and A. Nürnberger, “Research paper recommender system evaluation: A quantitative literature survey,” in *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*, 2013, pp. 15–22.
- [34] P. Jomsri, S. Sanguansintukul, and W. Choochaiwattana, “A framework for tag-based research paper recommender system: An ir approach,” in *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 2010, pp. 103–108.
- [35] J. Lee, K. Lee, and J. G. Kim, “Personalized academic research paper recommendation system,” *arXiv preprint arXiv:1304.5457*, 2013.
- [36] C. Geigle, Q. Mei, and C. Zhai, “Feature engineering for text data,” *Feature Engineering for Machine Learning and Data Analytics*, vol. 15, 2018.

- [37] D. King, D. Downey, and D. S. Weld, “High-precision extraction of emerging concepts from scientific literature,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 1549–1552.
- [38] A. Brack, J. D’Souza, A. Hoppe, S. Auer, and R. Ewerth, “Domain-independent extraction of scientific concepts from research articles,” in *European Conference on Information Retrieval*. Springer, 2020, pp. 251–266.
- [39] F. Dernoncourt and J. Y. Lee, “Pubmed 200k rct: A dataset for sequential sentence classification in medical abstracts,” in *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, 2017, pp. 308–313.
- [40] S. N. Kim, D. Martinez, L. Cavedon, and L. Yencken, “Automatic classification of sentences to support evidence based medicine,” in *BMC Bioinformatics*, vol. 12, no. S2. Springer, 2011, p. S5.
- [41] I. Augenstein, M. Das, S. Riedel, L. Vikraman, and A. McCallum, “Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications,” in *Proceedings of the 11th International Workshop on Semantic Evaluation*, 2017, pp. 546–555.
- [42] Y. Luan, L. He, M. Ostendorf, and H. Hajishirzi, “Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3219–3232.
- [43] W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. Ha et al., “Construction of the literature graph in semantic scholar,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 84–91.
- [44] M. Y. Jaradeh, A. Oelen, K. E. Farfar, M. Prinz, J. D’Souza, G. Kismihók, M. Stocker, and S. Auer, “Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge,” in *Proceedings of the 10th International Conference on Knowledge Capture*, 2019, pp. 243–246.
- [45] R. Wang, Y. Yan, J. Wang, Y. Jia, Y. Zhang, W. Zhang, and X. Wang, “Acekg: A large-scale knowledge graph for academic data mining,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1487–1490.
- [46] C. Xiong, R. Power, and J. Callan, “Explicit semantic ranking for academic search via knowledge graph embedding,” in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 1271–1279.

- [47] J. Shen, J. Xiao, X. He, J. Shang, S. Sinha, and J. Han, “Entity set search of scientific literature: An unsupervised ranking approach,” in *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2018, pp. 565–574.
- [48] Q. Wang, Q. Zeng, L. Huang, K. Knight, H. Ji, and N. F. Rajani, “Reviewrobot: Explainable paper review generation based on knowledge synthesis,” in *Proceedings of the 13th International Conference on Natural Language Generation*, 2020, pp. 384–397.
- [49] M. Lu, S. Bangalore, G. Cormode, M. Hadjieleftheriou, and D. Srivastava, “A dataset search engine for the research document corpus,” in *2012 IEEE 28th International Conference on Data Engineering*. IEEE, 2012, pp. 1237–1240.
- [50] D. Brickley, M. Burgess, and N. Noy, “Google dataset search: Building a search engine for datasets in an open web ecosystem,” in *The World Wide Web Conference*, 2019, pp. 1365–1375.
- [51] F. Liu and H. Yu, “Learning to rank figures within a biomedical article,” *PloS One*, vol. 9, no. 3, p. e61567, 2014.
- [52] F. Liu, T.-K. Jenssen, V. Nygaard, J. Sack, and E. Hovig, “Figsearch: A figure legend indexing and classification system,” *Bioinformatics*, vol. 20, no. 16, pp. 2880–2882, 2004.
- [53] X. Wang, C. Zhai, and D. Roth, “Understanding evolution of research themes: A probabilistic generative model for citations,” in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 1115–1123.
- [54] I. Beltagy, K. Lo, and A. Cohan, “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*, 2019, pp. 3615–3620.
- [55] I. Ruthven, “Interactive information retrieval,” *Annual Review of Information Science and Technology*, vol. 42, pp. 43–92, 2008.
- [56] P. Borlund, “The iir evaluation model: A framework for evaluation of interactive information retrieval systems,” *Information Research*, vol. 8, no. 3, pp. 8–3, 2003.
- [57] D. Kelly, “Methods for evaluating interactive information retrieval systems with users,” *Foundations and Trends in Information Retrieval*, vol. 3, no. 1–2, pp. 1–224, 2009.
- [58] A. Medlar, K. Ilves, P. Wang, W. Buntine, and D. Glowacka, “Pulp: A system for exploratory search of scientific literature,” in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2016, pp. 1133–1136.

- [59] M. Dunaiski, G. J. Greene, and B. Fischer, “Exploratory search of academic publication and citation data using interactive tag cloud visualizations,” *Scientometrics*, vol. 110, no. 3, pp. 1539–1571, 2017.
- [60] A. Sorkhei, K. Ilves, and D. Glowacka, “Exploring scientific literature search through topic models,” in *Proceedings of the 2017 ACM Workshop on Exploratory Search and Interactive Data Analytics*, 2017, pp. 65–68.
- [61] H. Yu, T. Kim, J. Oh, I. Ko, and S. Kim, “Refmed: Relevance feedback retrieval system for pubmed,” in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, 2009, pp. 2099–2100.
- [62] *BioMed Explorer*, (accessed August 25, 2021). [Online]. Available: <https://sites.research.google/biomedexplorer/>
- [63] A. Shen, B. Salehi, T. Baldwin, and J. Qi, “A joint model for multimodal document quality assessment,” in *2019 ACM/IEEE Joint Conference on Digital Libraries*. IEEE, 2019, pp. 107–110.
- [64] V. V. Vydiswaran, C. Zhai, and D. Roth, “Content-driven trust propagation framework,” in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 974–982.
- [65] C. Geigle, C. Zhai, and D. C. Ferguson, “An exploration of automated grading of complex assignments,” in *Proceedings of the Third ACM Conference on Learning@Scale*. ACM, 2016, pp. 351–360.
- [66] C. Sutton and L. Gong, “Popularity of arxiv. org within computer science,” *arXiv preprint arXiv:1710.05225*, 2017.
- [67] T. Ghosal, R. Verma, A. Ekbal, and P. Bhattacharyya, “A sentiment augmented deep architecture to predict peer review outcomes,” in *2019 ACM/IEEE Joint Conference on Digital Libraries*. IEEE, 2019, pp. 414–415.
- [68] Z. Deng, H. Peng, C. Xia, J. Li, L. He, and S. Y. Philip, “Hierarchical bi-directional self-attention networks for paper review rating recommendation,” in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 6302–6314.
- [69] S. Chakraborty, P. Goyal, and A. Mukherjee, “Aspect-based sentiment analysis of scientific reviews,” in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, 2020, pp. 207–216.
- [70] F. Qiao, L. Xu, and X. Han, “Modularized and attention-based recurrent convolutional neural network for automatic academic paper aspect scoring,” in *Proceedings of the International Conference on Web Information Systems and Applications*. Springer, 2018, pp. 68–76.

- [71] J. Li, A. Sato, K. Shimura, and F. Fukumoto, “Multi-task peer-review score prediction,” in *Proceedings of the First Workshop on Scholarly Document Processing*, 2020, pp. 121–126.
- [72] M. Skorikov and S. Momen, “Machine learning approach to predicting the acceptance of academic papers,” in *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology*. IEEE, 2020, pp. 113–117.
- [73] G. M. de Buy Wenniger, T. van Dongen, E. Aedmaa, H. T. Kruitbosch, E. A. Valentijn, and L. Schomaker, “Structure-tags improve text classification for scholarly document quality prediction,” in *Proceedings of the First Workshop on Scholarly Document Processing*, 2020, pp. 158–167.
- [74] G. Demartini, M. M. S. Missen, R. Blanco, and H. Zaragoza, “Entity summarization of news articles,” in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2010, pp. 795–796.
- [75] S. F. Adafre, M. de Rijke, and E. T. K. Sang, “Entity retrieval,” *Recent Advances in Natural Language Processing*, 2007.
- [76] D. Petkova and W. B. Croft, “Proximity-based document representation for named entity retrieval,” in *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*. ACM, 2007, pp. 731–740.
- [77] H. Raviv, D. Carmel, and O. Kurland, “A ranking framework for entity oriented search using markov random fields,” in *Proceedings of the 1st Joint International Workshop on Entity-Oriented and Semantic Search*. ACM, 2012, p. 1.
- [78] G. Salton, J. Allan, and C. Buckley, “Approaches to passage retrieval in full text information systems,” in *Proceedings of the 16th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1993, pp. 49–58.
- [79] S. Tellex, B. Katz, J. Lin, A. Fernandes, and G. Marton, “Quantitative evaluation of passage retrieval algorithms for question answering,” in *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. ACM, 2003, pp. 41–47.
- [80] M. Kaszkiel and J. Zobel, “Passage retrieval revisited,” in *ACM SIGIR Forum*, vol. 31, no. SI. ACM, 1997, pp. 178–185.
- [81] B. Mansouri, R. Zanibbi, and D. W. Oard, “Learning to rank for mathematical formula retrieval,” in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2021, p. 952–961.
- [82] S. R. Kunze and S. Auer, “Dataset retrieval,” in *2013 IEEE Seventh International Conference on Semantic Computing*. IEEE, 2013, pp. 1–8.

- [83] M. A. Hearst, A. Divoli, H. Guturu, A. Ksikes, P. Nakov, M. A. Wooldridge, and J. Ye, “Biotext search engine: Beyond abstract search,” *Bioinformatics*, vol. 23, no. 16, pp. 2196–2197, 2007.
- [84] A.-S. Sheikh, A. Ahmed, A. Arnold, L. P. Coelho, J. Kangas, E. P. Xing, W. Cohen, and R. F. Murphy, “Structured literature image finder: Open source software for extracting and disseminating information from text and figures in biomedical literature,” Carnegie Mellon University School of Computer Science, Pittsburgh, USA, CMU-CB-09-101, Tech. Rep., 2009.
- [85] H. Yu, F. Liu, and B. P. Ramesh, “Automatic figure ranking and user interfacing for intelligent figure search,” *PLoS One*, vol. 5, no. 10, p. e12983, 2010.
- [86] D. Kim and H. Yu, “Figure text extraction in biomedical literature,” *PloS One*, vol. 6, no. 1, p. e15338, 2011.
- [87] X.-C. Yin, C. Yang, W.-Y. Pei, H. Man, J. Zhang, E. Learned-Miller, and H. Yu, “Detext: A database for evaluating text extraction from biomedical literature figures,” *PLoS One*, vol. 10, no. 5, p. e0126200, 2015.
- [88] R. F. Murphy, Z. Kou, J. Hua, M. Joffe, and W. W. Cohen, “Extracting and structuring subcellular location information from on-line journal articles: The subcellular location image finder,” in *Proceedings of the IASTED International Conference on Knowledge Sharing and Collaborative Engineering*, 2004, pp. 109–114.
- [89] H. Müller, T. Deselaers, T. Deserno, P. Clough, E. Kim, and W. Hersh, “Overview of the imageclefmed 2006 medical retrieval and medical annotation tasks,” in *Workshop of the Cross-Language Evaluation Forum for European Languages*. Springer, 2006, pp. 595–608.
- [90] J. Eakins, M. Graham, and T. Franklin, “Content-based image retrieval,” in *Library and Information Briefings*. Education-line, 1999.
- [91] D. S. Shete, M. Chavan, and K. Kolhapur, “Content based image retrieval,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 9, pp. 85–90, 2012.
- [92] J. Ah-Pine, G. Csurka, and S. Clinchant, “Unsupervised visual and textual information fusion in cbmir using graph-based methods,” *ACM Transactions on Information Systems*, vol. 33, no. 2, p. 9, 2015.
- [93] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156–3164.
- [94] S. Dey, A. Dutta, S. K. Ghosh, E. Valveny, J. Lladós, and U. Pal, “Learning cross-modal deep embeddings for multi-object image retrieval using text and sketch,” in *Proceedings of the 24th International Conference on Pattern Recognition*. IEEE, 2018, pp. 916–921.

- [95] Y. Ding, G. Zhang, T. Chambers, M. Song, X. Wang, and C. Zhai, “Content-based citation analysis: The next generation of citation analysis,” *Journal of the Association for Information Science and Technology*, vol. 65, no. 9, pp. 1820–1833, 2014.
- [96] H. Yu and M. Lee, “Accessing bioscience images from abstract sentences,” *Bioinformatics*, vol. 22, no. 14, pp. e547–e556, 2006.
- [97] T.-Y. Liu, “Learning to rank for information retrieval,” *Foundations and Trends in Information Retrieval*, vol. 3, no. 3, p. 225–331, 2009.
- [98] S. E. Robertson and S. Walker, “Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval,” in *Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Springer-Verlag New York, Inc., 1994, pp. 232–241.
- [99] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.
- [100] S. Bird, R. Dale, B. J. Dorr, B. Gibson, M. T. Joseph, M. Y. Kan, D. Lee, B. Powley, D. R. Radev, and Y. F. Tan, “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*. European Language Resources Association, 2008, pp. 1755–1759.
- [101] C. A. Clark and S. Divvala, “Looking beyond text: Extracting figures, tables and captions from computer science papers,” in *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
- [102] Q. Wu, C. J. Burges, K. M. Svore, and J. Gao, “Adapting boosting for information retrieval measures,” *Information Retrieval*, vol. 13, no. 3, pp. 254–270, 2010.
- [103] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.
- [104] S. Massung, C. Geigle, and C. Zhai, “Meta: A unified toolkit for text retrieval and analysis,” *Proceedings of ACL-2016 System Demonstrations*, pp. 91–96, 2016.
- [105] J. Weston, S. Bengio, and N. Usunier, “Large scale image annotation: Learning to rank with joint word-image embeddings,” *Machine Learning*, vol. 81, no. 1, pp. 21–35, 2010.
- [106] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.
- [107] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

- [108] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *Proceedings of the International Conference on Machine Learning*, 2014, pp. 1188–1196.
- [109] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, “Learning fine-grained image similarity with deep ranking,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1386–1393.
- [110] M. Deghani, H. Zamani, A. Severyn, J. Kamps, and W. B. Croft, “Neural ranking models with weak supervision,” in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 65–74.
- [111] A. Bordes, J. Weston, and N. Usunier, “Open question answering with weakly supervised embedding models,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 165–180.
- [112] J. P. Bockhorst, J. M. Conroy, S. Agarwal, D. P. O’Leary, and H. Yu, “Beyond captions: Linking figures with abstract sentences in biomedical articles,” *PloS One*, vol. 7, no. 7, p. e39618, 2012.
- [113] L. Wang, Y. Li, and S. Lazebnik, “Learning deep structure-preserving image-text embeddings,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5005–5013.
- [114] B. Klein, G. Lev, G. Sadeh, and L. Wolf, “Associating neural word embeddings with deep image representations using fisher vectors,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4437–4446.
- [115] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, “Devise: A deep visual-semantic embedding model,” in *Advances in Neural Information Processing Systems*, 2013, pp. 2121–2129.
- [116] J. Bromley, I. Guyon, Y. LeCun, E. Säcker, and R. Shah, “Signature verification using a ‘siamese’ time delay neural network,” in *Advances in Neural Information Processing Systems*, 1994, pp. 737–744.
- [117] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [118] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional lstm and other neural network architectures,” *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.
- [119] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

- [120] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [121] X. Wang and C. Zhai, “Mining term association patterns from search logs for effective query reformulation,” in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, 2008, pp. 479–488.
- [122] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, “Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search,” *ACM Transactions on Information Systems*, vol. 25, no. 2, p. 7, 2007.
- [123] H.-R. Ke, R. Kwakkelaar, Y.-M. Tai, and L.-C. Chen, “Exploring behavior of e-journal users in science and technology: Transaction log analysis of elsevier’s sciencedirect onsite in taiwan,” *Library and Information Science Research*, vol. 24, no. 3, pp. 265–291, 2002.
- [124] X. Li, B. J. Schijvenaars, and M. de Rijke, “Investigating queries and search failures in academic search,” *Information Processing and Management*, vol. 53, no. 3, pp. 666–683, 2017.
- [125] D. D. Lewis, “Representation and learning in information retrieval,” Ph.D. dissertation, University of Massachusetts at Amherst, 1992.
- [126] C. Zhai, “Towards a game-theoretic framework for text data retrieval,” *IEEE Data Eng. Bull.*, vol. 39, no. 3, pp. 51–62, 2016.
- [127] V. Lavrenko and W. B. Croft, “Relevance based language models,” in *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001, pp. 120–127.
- [128] N. J. Belkin, C. Cool, D. Kelly, S.-J. Lin, S. Park, J. Perez-Carballo, and C. Sikora, “Iterative exploration, design and evaluation of support for query reformulation in interactive information retrieval,” *Information Processing and Management*, vol. 37, no. 3, pp. 403–434, 2001.
- [129] P. Anick, “Using terminological feedback for web search refinement: A log-based study,” in *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003, pp. 88–95.
- [130] I. Ruthven, “Re-examining the potential effectiveness of interactive query expansion,” in *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2003, pp. 213–220.
- [131] H. Bast, D. Majumdar, and I. Weber, “Efficient interactive query expansion with complete search,” in *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, 2007, pp. 857–860.

- [132] A. Sihvonen and P. Vakkari, “Subject knowledge improves interactive query expansion assisted by a thesaurus,” *Journal of Documentation*, vol. 60, no. 6, pp. 673–690, 2004.
- [133] M. Magennis and C. J. van Rijsbergen, “The potential and actual effectiveness of interactive query expansion,” in *ACM SIGIR Forum*, vol. 31, no. SI. ACM, 1997, pp. 324–332.
- [134] G. Kumaran and J. Allan, “Effective and efficient user interaction for long queries,” in *Proceedings of the 31st International ACM SIGR Conference on Research and Development in Information Retrieval*, 2008, pp. 11–18.
- [135] M. Hancock-Beaulieu, M. Fieldhouse, and T. Do, “An evaluation of interactive query expansion in an online library catalogue with a graphical user interface,” *Journal of Documentation*, vol. 51, no. 3, pp. 225–243, 1995.
- [136] R. Baeza-Yates, C. Hurtado, and M. Mendoza, “Query recommendation using query logs in search engines,” in *International Conference on Extending Database Technology*. Springer, 2004, pp. 588–596.
- [137] Y. Song and L.-w. He, “Optimal rare query suggestion with implicit user feedback,” in *Proceedings of the 19th International Conference on World Wide Web*, 2010, pp. 901–910.
- [138] V. Dang and B. W. Croft, “Query reformulation using anchor text,” in *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*. ACM, 2010, pp. 41–50.
- [139] Y. Liu, R. Song, Y. Chen, J.-Y. Nie, and J.-R. Wen, “Adaptive query suggestion for difficult queries,” in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012, pp. 15–24.
- [140] F. Bonchi, R. Perego, F. Silvestri, H. Vahabi, and R. Venturini, “Efficient query recommendations in the long tail via center-piece subgraphs,” in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012, pp. 345–354.
- [141] S. Huo, M. Zhang, Y. Liu, and S. Ma, “Improving tail query performance by fusion model,” in *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, 2014, pp. 559–568.
- [142] D. Broccolo, L. Marcon, F. M. Nardini, R. Perego, and F. Silvestri, “Generating suggestions for queries in the long tail with an inverted index,” *Information Processing and Management*, vol. 48, no. 2, pp. 326–339, 2012.
- [143] I. Szpektor, A. Gionis, and Y. Maarek, “Improving recommendation for ong-tail queries via templates,” in *Proceedings of the 20th International Conference on World Wide Web*, 2011, pp. 47–56.

- [144] C. Carpineto and G. Romano, “A survey of automatic query expansion in information retrieval,” *ACM Computing Surveys*, vol. 44, no. 1, pp. 1–50, 2012.
- [145] S. Kuzi, A. Shtok, and O. Kurland, “Query expansion using word embeddings,” in *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, 2016, pp. 1929–1932.
- [146] F. Diaz, B. Mitra, and N. Craswell, “Query expansion with locally-trained word embeddings,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 367–377.
- [147] F. Diaz, “Pseudo-query reformulation,” in *European Conference on Information Retrieval*. Springer, 2016, pp. 521–532.
- [148] J. Rocchio, “Relevance feedback in information retrieval,” *The SMART Retrieval System: Experiments in Automatic Document Processing*, pp. 313–323, 1971.
- [149] E. M. Voorhees, “Query expansion using lexical-semantic relations,” in *Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994, pp. 61–69.
- [150] H. Zamani and W. B. Croft, “Embedding-based query language models,” in *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, 2016, pp. 147–156.
- [151] A. Spink, B. J. Jansen, and H. Cenk Ozmultu, “Use of query reformulation and relevance feedback by excite users,” *Internet Research*, vol. 10, no. 4, pp. 317–328, 2000.
- [152] E. Brondwine, A. Shtok, and O. Kurland, “Utilizing focused relevance feedback,” in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2016, pp. 1061–1064.
- [153] A. Shtok, O. Kurland, D. Carmel, F. Raiber, and G. Markovits, “Predicting query performance by query-drift estimation,” *ACM Transactions on Information Systems*, vol. 30, no. 2, pp. 1–35, 2012.
- [154] Y. Zhou and W. B. Croft, “Query performance prediction in web search environments,” in *Proceedings of the 30th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007, pp. 543–550.
- [155] S. Cronen-Townsend, Y. Zhou, and W. B. Croft, “Predicting query performance,” in *Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002, pp. 299–306.
- [156] C. Hauff, D. Hiemstra, and F. de Jong, “A survey of pre-retrieval query performance predictors,” in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, 2008, pp. 1419–1420.

- [157] N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade, “Umass at trec 2004: Novelty and hard,” Tech. Rep.
- [158] A. Ragone, K. Mirylenka, F. Casati, and M. Marchese, “A quantitative analysis of peer review,” in *Proceedings of the 13th International Society of Scientometrics and Informetrics Conference*, 2011.
- [159] P. U. De Silva and C. K. Vance, “Preserving the quality of scientific research: Peer review of research articles,” in *Scientific Scholarly Communication*. Springer, 2017, pp. 73–99.
- [160] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 50–57.
- [161] Y. Wang, D. Seyler, S. K. K. Santu, and C. Zhai, “A study of feature construction for text-based forecasting of time series variables,” in *Proceedings of the ACM Conference on Information and Knowledge Management*. ACM, 2017, pp. 2347–2350.
- [162] X. Yi and J. Allan, “A comparative study of utilizing topic models for information retrieval,” in *European Conference on Information Retrieval*. Springer, 2009, pp. 29–41.
- [163] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li, “Comparing twitter and traditional media using topic models,” in *European Conference on Information Retrieval*. Springer, 2011, pp. 338–349.
- [164] J.-B. Huang, “Deep paper gestalt,” *arXiv preprint arXiv:1812.08775*, 2018.
- [165] T. Ghosal, R. Verma, A. Ekbal, and P. Bhattacharyya, “Deepsentipeer: Harnessing sentiment in review texts to recommend peer review decisions,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1120–1130.
- [166] S. Li, W. X. Zhao, E. J. Yin, and J.-R. Wen, “A neural citation count prediction model based on peer review text,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, pp. 4916–4926.
- [167] W. Yuan, P. Liu, and G. Neubig, “Can we automate scientific reviewing?” *arXiv preprint arXiv:2102.00176*, 2021.
- [168] V. Balachandran, “Reducing human effort and improving quality in peer code reviews using automatic static analysis and reviewer recommendation,” in *Proceedings of the 35th International Conference on Software Engineering*. IEEE, 2013, pp. 931–940.
- [169] S. P. Balfour, “Assessing writing in moocs: Automated essay scoring and calibrated peer review.” *Research and Practice in Assessment*, vol. 8, pp. 40–48, 2013.

- [170] Y. Lu, Q. Mei, and C. Zhai, “Investigating task performance of probabilistic topic models: An empirical study of plsa and lda,” *Information Retrieval*, vol. 14, no. 2, pp. 178–203, 2011.
- [171] J. D. Mcauliffe and D. M. Blei, “Supervised topic models,” in *Advances in Neural Information Processing Systems*, 2008, pp. 121–128.
- [172] J. Zhu, A. Ahmed, and E. P. Xing, “Medlda: Maximum margin supervised topic models for regression and classification,” in *Proceedings of the 26th International Conference on Machine Learning*. ACM, 2009, pp. 1257–1264.
- [173] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, “The author-topic model for authors and documents,” in *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2004, pp. 487–494.
- [174] S. Hasan and S. V. Ukkusuri, “Urban activity pattern classification using topic models from online geo-location data,” *Transportation Research Part C: Emerging Technologies*, vol. 44, pp. 363–381, 2014.
- [175] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, “Very deep convolutional networks for text classification,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017, pp. 1107–1116.
- [176] J. Shang, J. Liu, M. Jiang, X. Ren, C. R. Voss, and J. Han, “Automated phrase mining from massive text corpora,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 10, pp. 1825–1837, 2018.
- [177] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han, “Mining quality phrases from massive text corpora,” in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015, pp. 1729–1744.
- [178] S. R. Choudhury, S. Tuarob, P. Mitra, L. Rokach, A. Kirk, S. Szep, D. Pellegrino, S. Jones, and C. L. Giles, “A figure search engine architecture for a chemistry digital library,” in *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*, 2013, pp. 369–370.
- [179] N. Siegel, Z. Horvitz, R. Levin, S. Divvala, and A. Farhadi, “Figureseer: Parsing result-figures in research papers,” in *European Conference on Computer Vision*. Springer, 2016, pp. 664–680.
- [180] P. B. Teregowda, M. Khabsa, and C. L. Giles, “A system for indexing tables, algorithms and figures,” in *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries*, 2012, pp. 343–344.
- [181] S. Bhatia and P. Mitra, “Summarizing figures, tables, and algorithms in scientific publications to augment search results,” *ACM Transactions on Information Systems*, vol. 30, no. 1, pp. 1–24, 2012.

- [182] L. L. Wang, K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Burdick, D. Eide, K. Funk, Y. Katsis, R. M. Kinney et al., “Cord-19: The covid-19 open research dataset,” in *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, 2020.
- [183] *Arxiv*, (accessed August 25, 2021). [Online]. Available: <https://arxiv.org/>
- [184] *ACL Anthology*, (accessed August 25, 2021). [Online]. Available: <https://aclanthology.org/>