Mining Research Problems from Scientific Literature

Thesis submitted in partial fulfillment
of the requirements for the degree of

(Master of Science in Computer Science and Engineering by Research)

by

Chanakya Aalla
201002140
chanakya.aalla@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Mining research problems in Scientific Literature" by Chanakya, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Dr. Vikram Pudi

To my family

Acknowledgments

I would like to extend my heartfelt gratitude to everyone who has supported and guided me throughout the challenging journey of completing this thesis.

First and foremost, I am deeply indebted to my thesis guide, Dr. Vikram Pudi, whose expertise, patience and encouragement have been invaluable. Their insightful feedback and unwavering support have been crucial in shaping this work. Despite the difficulties, their guidance helped me stay focused and motivated, ultimately leading me to complete this research.

I am also profoundly grateful for my family for their endless love, support and understanding. They have been my pillars of strength, providing me with the emotional and moral support needed to persevere through the hard times. Their belief in my abilities and constant encouragement have been a driving force behind my efforts.

My heartfelt thanks also go to my friends, whose camraderie and encouragement have been a source of comfort and motivation. Special thanks to Avinash, Dileep and Harsha, who have not only been there to share in my triumphs and setbacks but have also pushed me to strive for excellence. Their support has been a vital part of this journey, and I am incredibly grateful for their presence in my life.

This thesis process has been ardous and demanding, but the combined support and encouragement from Vikram sir, family and friends have been instrumental in helping me reach this milestone. Thier belief in me provided the strength and resilience needed to see this project through to completion.

Finally, I would like to acknowledge the broader academic community at IIIT Hyderabad for providing an enriching environment that has fostered my intellectual growth. The resources, facilities, and academic discourse have significantly contributed to my research and overall experience.

This thesis is a testament to the collective effort of all the wonderful people who have been part of my journey. To each of you, I extend my deepest gratitude and appreciation. Thank you for your unwavering support and belief in my potential.

Abstract

Extracting structured information from unstructured text is a critical problem. Over the past few years, various clustering algorithms have been proposed to solve this problem. In addition, various algorithms based on probabilistic topic models have been developed to find the hidden thematic structure from various corpora (i.e publications, blogs etc). Both types of algorithms have been transferred to the domain of scientific literature to extract structured information to solve problems like data exploration, expert detection etc.

In order to remain domain-agnostic, these algorithms do not exploit the structure present in a scientific publication. Majority of researchers interpret a scientific publication as research conducted to report progress in solving some *research problems*. Following this interpretation, in this paper we present a different outlook to the same problem by modelling scientific publications around research problems. By associating a scientific publication with a research problem, exploring the scientific literature becomes more intuitive.

In this thesis, we propose an unsupervised framework to mine research problems from titles and abstracts of scientific literature. Our framework uses weighted frequent phrase mining to generate phrases and filters them to obtain high-quality phrases. These high-quality phrases are then used to segment the scientific publication into meaningful semantic units. After segmenting publications, we apply a number of heuristics to score the phrases and sentences to identify the research problems. In a post-processing step we use a neighborhood based algorithm to merge different representations of the same problems. Experiments conducted on parts of DBLP dataset show promising results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Over the last few years, there has been an astronomic increase in data, in particular unstructured data. To contend with this surge of data, various information retrieval systems based on statistics have been developed over the years. Even with these systems in-place, it soon became a challenge to conflate concepts, disambiguate data, handle duplication and identify hidden structures between entities. To overcome this, many system evolved to supplement statistics by *extracting structure* from unstructured data. Consider the World Wide Web for example, where search engines of yesteryear relied only on unstructured text. As the data grew exponentially, this model wasn't sufficient as it had various problems. Eventually, these systems evolved into domain-specific knowledge bases [1, 20] to improve user experience. They improved by defining a domain-specific ontology and extracting structure[20] from aforementioned unstructured data.

These problems are more apparent for scientific literature, where the amount of research published each year is steadily increasing[17]. Consider Figure 1.1, which shows the number of publications

Figure 1.1: # of publications indexed by dblp across years

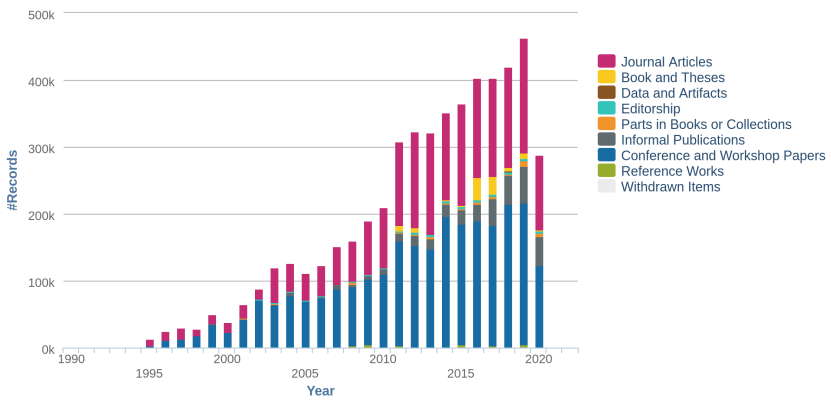
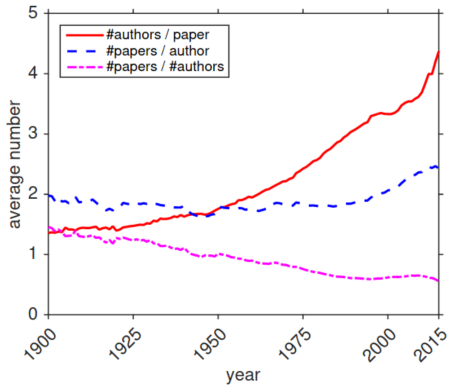


Figure 1.2: # OF RESEARCH PRODUCED ACROSS DECADES



indexed by the DBLP computer science bibliography project by year. We can observe that the number of publications roughly increased by approximately 25% in the last five years. It's evident that it's no longer trivial for a researcher to be well-informed of their research areas.

This upward trend is not exclusive to computer science literature, similar trends are observed in various other domains. Consider Figure 1.2, which shows the amout of multi-domain research published across decades as indexed by Microsoft Academic. With this huge influx of data, there is a critical need for systems which help researchers navigate, understand and analyze this data.

Over the years, many systems[21, **?**, 2, 31] have been developed to aid researchers to discover and analyze relevant literature. Traditionally these systems started out with keyword search and had little to no automatic organization. There have been attempts to employ topic detection approaches to organize research corpora. In recent years, many new systems [5, 44, 41] have been developed which use named entity recognition, topic detection and NLU techniques to discover and organize research corpora. Even in these systems we do not take into account of the inherent semantic structure found in scientific literature. We miss out on some valuable insights because of that.

In this thesis, we present an alternative approach to *organize* huge research corpora. We formally define the concept of research problem, then present a system that extracts high-quality research problems and explore it's applications to augment the existing systems to process and analyze research corpora.

The rest of the chapter is organized as follows. In Section 1.2, we present a brief overview of knowledge discovery in scientific literature. Next in Section 1.3, we discuss the usual challenges of working with scientific literature. In Section 1.4, we define the problem addressed by this thesis. In Section 1.5, we outline a high-level overview of our approach. In Section 1.5, we identify the contributions of our thesis followed by the organization of thesis in Section 1.6.

1.2 Knowledge Discovery in Scientific Literature

As mentioned in the previous section, the rapid increase of scientific literature complicated researcher's knowledge discovery workflows. Consider the following examples of finding relevant research:

- Investigating whether a research problem has been solved
- Explore how algorithms have evolved to solve a particular problem
- Understand how community interest has shifted on a research problem across years
- How solving a research problem affects related research problems

There's no trivial way to answer these questions unless one is intimately familiar with the domain. One straight-forward way to approach these questions to find appropriate search phrases/keywords and use an academic search engine to find research papers and traverse the author & citation graphs to explore more relevant work. This is a rather tiresome process and occasionally, leads to an incomplete exploration of the problem domain.

Similarly, in order to keep themselves updated with the latest developments in the field, researchers identify key researchers, conferences and journals and vigorously follow them to get the candidate literature and then filter them to find the relevant work. This approach is not efficient and time consuming. The existence of domain specific curated guides[1, 2, 3, 4] to research papers, geared towards students and new practitioners reinforces the complexity of this process.

Over the years, a myriad of approaches ranging from text mining to graph clustering approaches have been developed to *organize* scientific literature so that researchers can easily discover relevant research and stay upto date with the latest trends. For years, probabilistic topic models [12, 25] have been used to mine hidden thematic structure for documents. Variants of these topic models [43, 50, 22, 37, 28] have been developed to find topic distributions for publications, authors, conferences and use these distributions to answer aforementioned questions. Another class of approaches include using citation graph [39, 27, 26, 16, 3, 42, 46] and content-similarity to cluster scientific articles. There have been also approaches using various NLP techniques to find relevant work.

In this thesis, we attempt to provide a new perspective on discovering relevant research by extracting *research problems* using the semantic structure present in scientific literature and use them to enhance existing algorithms.

1.3 Challenges of working with Scientific Literature

As discussed in the previous section, various text mining techniques have been applied to scientific literature. However, working with scientific literature poses a distinct set of problems when compared to similar unstructured data e.g. tweets, blogposts, reviews etc. We discuss a few of them below:

1.3.1 Data Wrangling

Majority of scientific literature is available in PDF format, which was primarily designed for archival and portability across platforms. A PDF document encapsulates the description of page layouts, fonts, graphics and other images but doesn't allow to extract text matching semantic structure (e.g., title, abstract, author etc in scientific literature). In addition, the presence of mathematical equations, differing formats for conferences, journals etc adds more nuance to the problem. Over the years, many approaches [38, 32, 30, 7, 8] have been developed for extracting structure from PDF, many of which are targeted at Scientific Literature.

1.3.2 Wordings

In a natural language, the existence of multiple phrases to represent concepts is extremely prevalent. As such, this phenomena is also observed in scientific literature, where an unique scientific concept might be represented by numerous wordings. This problem is further exacerbated by existence of overlapping concepts, intersecting domains, evolution of aforementioned concepts and interpretations by various researchers with diverse backgrounds. Traditionally, approaches based on frequency statistics, latent semantic indexing and frameworks like WordNet[36] have been used to resolve these phrasings into concepts. However in recent times, with the advances in deep learning, word-embedding approaches [35, 15, 13] achieve better results.

1.3.3 Scalability

As mentioned in the previous section, the volume of scientific literature is huge. According to a study published by Microsoft Academic[17], the approximate number of scientific literature is upwards of 300 million. In addition to that, a single publication is many times larger than compared to articles on web, blog posts, tweets etc. This explosion of scale means that traditional algorithms are infeasible if we work with full text of scientific literature. Traditionally, algorithms used to work on titles[1, 2, 3] instead of the full-text to handle the scaling issues but that means we trade valuable semantic information captured by the document.

1.3.4 Evaluation

When compared with other forms of unstructured data, the absence of gold standard data for scientific literature makes evaluation challenging. In addition, the lack of systems to mine research problems, makes evaluating of quality of research problems non-trivial. We have to rely on manual evaluation which comes with it's own set of nuances e.g., handling researchers with different knowledge levels, inherent subjectiveness and evalution of huge datasets.

Later in Chapter 3, we discuss how we leveraged existing research to build a system that extracts text from PDF and present an approach based on word embeddings and statistics to handle the *wordings* problem. In Chapter 4, we address our approach to handle the nuances of manual evaluation.

1.4 Problem Definition

Consider a corpus C consisting of publications $p_1, p_2, p_3, \dots, p_n$, where each publication p_i is a sequence of words i.e. p_i = [w_{i1}, w_{i2}, w_{i3}, ...,w_{im}]. Without any loss of generality, we subscribe to the notion that a publication p_i solves a set of critical problems $\{r_{i1}, r_{i2}, \dots, r_{ik}\}$, henceforth referred to as *research problems*. This notion holds well even for review publications and case studies, where the research problem is to provide a comprehensive summary of the state of the research field, compare various approaches and their shortcomings.

1.4.1 Concepts

As mentioned above, each publication is represented as a sequence of words. In order to extract structured information, we need to separate meaningful entities from grammatical scaffoldings. Every research field has its own concepts i.e., body of words that represent meaning to researchers. Given a publication p_i in corpus C, we need to identify concepts. They are usually manifested as words and phrases.

Formally, we define a concept as a word w_i or phrase $[w_i, w_{i+1}, \dots, w_{i+k}]$ which has semantic relevance within the context of publication p_i and or corpus C. It is important to observe that concepts are defined in the scope of publication/corpus. Hence a specific word or phrase, considered as a concept for corpus C_1 might not be considered as a concept for corpus C_2. As with any unstructured text, many concepts and relations are manifested in publications. For example, consider Table 1.1, where we present the annotated abstracts excerpted from [12] and [23]:

Latent Dirichlet Allocation

We describe Latent Dirichlet allocation (LDA), a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document. We present efficient approximate inference techniques based on variational methods and an EM algorithm for empirical Bayes parameter estimation. We report results in document modeling, text classification, and collaborative filtering, comparing to a mixture of unigrams model and the probabilistic LSI model.

Mining Frequent Patterns without Candidate Generation

Mining frequent patterns in transaction databases, time-series databases, and many other kinds of databases has been studied popularly in data mining research. Most of the previous studies adopt an Apriori-like candidate set generation-and-test approach. However, candidate set generation is still costly, especially when there exist prolific patterns and/or long patterns. In this study, we propose a novel frequent pattern tree (FP-tree) structure, which is an extended prefix-tree structure for storing compressed, crucial information about frequent patterns, and develop an efficient FP-tree based mining method, FP-growth, for mining the complete set of frequent patterns by pattern fragment growth. Efficiency of mining is achieved with three techniques: (1) a large database is compressed into a highly condensed, much smaller data structure, which avoids costly, repeated database scans, (2) our FP-tree-based mining adopts a pattern fragment growth method to avoid the costly generation of a large number of candidate sets, and (3) a partitioning-based, divide-and-conquer method is used to decompose the mining task into a set of smaller tasks for mining confined patterns in conditional databases, which dramatically reduces the search space. Our performance study shows that the ...

Table 1.1: An example highlighting the concepts and their types

From the above example, we can discern the various concepts manifested in publications. It is important to distinguish the 'type' of a concept which can be inferred by examining the context of its occurence. It is extremely common to have various kinds of concepts manifested in publications. For example consider the phrases *time-series databases*, *long patterns*, *text corpora*, these concepts describe the charactersitics of input data where as the concepts *probabilistic topic model*, *variational methods*, *approximate inference techniques* describe the 'class' of algorithms used. Sometimes, concepts can be decomposed into mutliple sub-concepts. For example, consider the phrase *three-level hierarchical bayesian model* concepts e.g., *bayesian model*, *hierarchical bayesian model*. For the purposes of our framework, we consider all of the sub-concepts to be valid and actively try to detect them.

It's also quite possible that sometimes concepts are worded in such a way that makes it difficult to detect and conflate them. Consider the phrase *representation of the document*, it refers to the concept of *document representation model* but worded in a different way. By not conflating these concepts, we lose out the information that LDA generated topic probabilities can be used as document representation model. It is also not uncommon for a concept, which has semantic relevance but not directly related

to any of the aforementioned types. Phrases like *topic probabilities*, *unigrams model*, *candidate set generation* fall under this category. Other interesting examples of concept types are research problems, algorithm names, algorithm class and their attributes.

1.4.2 Research problems

We define research problem as a type of concept which describes a problem that a publication intends to solve. Consider the phrases *document modelling*, *collaborative filtering*, and *mining frequent patterns* from Table1.1, they either describe the broad/specific instances of problems that are impacted/solved by an algorithm proposed in the publication. It's not uncommon to see publications impact mutliple research problems e.g. Latent Dirichlet Allocation from Table1.1.

DEFINITION

Given a publication p in corpus C, we define a research problem r as a word w_i or a phrase $[w_i, w_{i+1}, \dots, w_{i+k}]$ which has a) semantic relevance within C and b) describes a problem that publication solves. We also define Impact I_p for a given publication, $I_p = \{r_1, r_2, r_3, ...\}$ which is a maximal set of research problems manifested in the publication.

In this thesis, we present a framework that extracts concepts c, research problems r and impact I_p from a given corpus C. We operate on the full-text of publications however for the most part, research problems are usually manifested in titles and abstracts. It is important to note that it's not always not possible to detect research problems. In such cases, we find appropriate sentences from abstracts which best represents the publication's contribution.

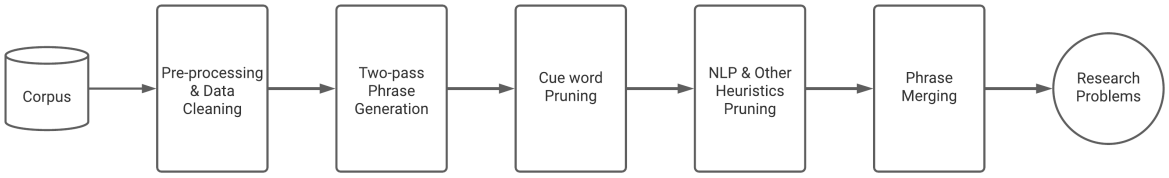
1.4.3 Application of Research problems

Mining research problems in and itself is an interesting problem but has further implications. They can be viewed as a simplified document representation model for various problems like data/ trends exploration e.g. browsing publications using research problems instead of keyword and topical exploration. This is arguably more intuitive iff research problems are of 'high-quality' and capture the semantics of the publication.

Another perspective is that research problems naturally the relationships between publications, authors and conferences implying applications for clustering (i.e grouping publicatons with similar research problems), expert-detection (i.e. finding authors with impactful papers for a given research problem) etc. Furthermore, using research problems as a document representation model instead of the entire publications has the following advantages

- Working at a conceptual level.
- Reducing the dimensionality of the data.

Figure 1.3: An high-level overview of our framework



1.5 Overview of Proposed Approach

We start by passing the corpus through a pre-processing pipeline to reliably extract the text. In order to mine research problems, it is essential to detect the all concepts that are manifested in publications. We use a two-phase weighted frequent pattern mining algorithm to efficiently generate the candidate phrases, then use an agglomerative algorithm to obtain a bag-of-phrases representation of the corpus. We then use a variety of heuristics based on cue words, statistics, and POS tags to aggressively to prune superflous concepts. To handle various wordings of research problems, we then use a text-rank based algorithm to merge similar research problems. Figure 1.3 shows the high level overview of our framework.

1.6 Contribution of Thesis

The main contribution of this thesis lie in the following aspects:

- Introduce the novel problem of mining research problems from scientific literature.
- Introduce a two-phase weighted frequent pattern mining to efficiently mine candidate phrases from a huge corpus.
- Develop an algorithm to generate an exhaustive list of cue words using a seed list of words as an input.
- Finding the set of phrase(s)/sentences(s) that best describe(s) the research problem that a publication intends to solve.
- Since a research problem can be often worded differently, we propose a text-rank based algorithm which capitalizes on neighborhood for merging similar wordings.

1.7 Organization of Thesis

The rest of the thesis is organized as follows:

Chapter 2 In this chapter, we discuss the relevant work in the fields topic modelling, phrase mining and document clustering. We also discuss how these approaches are applied to the domain of scientific literature. We also provide an overview of some common properties of titles and abstracts found in publications which we use in our framework. We also highlight key differences in our approaches.

Chapter 3 In this chapter, we elaborate on the unsupervised framework to extract research problems from scientific literature. We also elaborate on the motivation and trade-offs of our approach.

Chapter 4 In this chapter, we describe the various datasets, and the approaches we undertook to evaluate the different sub-systems of our framework. We also verify the impact of various design decisions of our framework.

Chapter 5 Finally, we conclude by providing a brief summary of our work and highlight some directions for future work.

Chapter 2

Related Work

In this chapter, we would like to present the related work in the fields of topic modelling, phrase mining and frequent pattern mining and word representations which can be useful to understand our work. In Section 2.3, we discuss the properties observed in titles and abstracts of publications which are later used in Chapter 3. Finally we conclude the chapter by highlighting the key differences between our approach and existing research.

2.1 Topic Modelling

In 2003, David Blei[12] proposed Latent Dirichlet Allocation, the simplest topic model in which topic distribution is assumed to follow Dirichlet prior distribution. Griffiths T. and Blei[10] applied topic models to find topics and correlated topics in Science journals like PNAS. For every probabilistic topic model there are two key components namely *generative model* which is the process used to generate the documents and the *inference method* used to estimate the parameters. Many variations have been proposed based on the inference method. Approximate inference algorithms like variational Expectation-Maximization[9, 24], Gibbs sampling using Markov Chain Monte Carlo method (MCMC) etc. have been proposed to find parameters that maximize the likelihood. Alternatively we have algorithms like Tensor Orthogonal Decomposition[49] and Scalable Tensor Orthogonal Decomposition which aim to find parameters that fit the method of moments.

In order to assist with data exploration, many topic modelling algorithms have moved from unigram topic representation to n-gram topic representation. Marina Danilevsky and Jiawei Han introduce KERT[14], which clusters words in the document dataset using LDA. KERT then extracts candidate keyphrases within each topic according to word topic assignment and finally rank them in the relevant order. With KERT the phrase quality relies on topic models for unigrams. Similar to KERT, TurboTopics[11] also estimates the LDA topic model and annotate the corpus with its most likely posterior topic and merge those words recursively to generate significant n-grams for each topic. Both of these methods apply topic modelling on unigrams and then generate phrases. There has been very little work done on generating phrases first and then performing topic modelling on those phrases. Kim

et.al[29] introduced frequent pattern enriched topic model which describes a general way to go beyond the bag-of-words representation for topic modelling by applying frequent pattern mining to discover frequent word patterns that can capture semantic associations between words and then use them as additional supplementary semantic units to augment the conventional bag-of-words representation. El-kishky and Jiwaei Han introduced TopMine[18] an efficient to generate topical phrases by first mining phrases, segmenting each document into single and multi-word phrases and using constraints from segmentation in topic modelling. This generates coherent topical phrases and is considered to be the state-of-the-art.

2.2 Phrase Mining

Automatic extraction of phrases is an extensively studied problem. A rudimentary approach extracts phrases by finding the top-k significant n-grams where the score of n-gram is defined as a summation of the individual tf-idf scores. Surprisingly this method has good accuracy and is domain agnostic. Another method applies frequent pattern mining to find all phrases that have support greater than threshold support. By applying the contiguity constraint (only continuous set of words are candidates) and downward closure property, FPM extracts high-quality phrases and is domain-agnostic. Mihalca and Tarau introduce TextRank[33], where text is represented as graph with words as vertices and co-occurrence relation within a window w as edge. TextRank applies page-rank algorithm to this graph and outputs top k adjacent vertices as phrases. Wan and Xiao introduce SingleRank[47] and ExpandRank[48], variants of TextRank with different interpretations of edge weight and window size. In SingleRank, edge weight is defined as co-occurrence frequency within a window w between vertices. Each candidate keyphrase is ranked and top k keyphrases are outputted. Apart from statistical and graph based approaches, NLP techniques like POS tagging, noun chunking can also be used to extract phrases. Ian Witten and Gordon W. Paynter introduce KEA[51], which treats the problem as a supervised learning from examples. It uses naive Bayes machine learning algorithm for training using tf-idf score and first occurrence as features. Franceso Sclano and Poala Velardi developed TermExtractor[40], in which a linguistic processor is used to parse and extract compounds, adjective-noun and noun-preposition-noun and filter them using various measures like domain pertinence, lexical cohesion etc. A much more comprehensive study of existing NLP approaches can be seen in [6].

Many of the aforementioned works extract phrases without considering any domain specific information, the quality of phrases can be improved by taking advantage of the structure present in titles and abstracts of scientific publications. In algorithms like TextRank, SingleRank etc., concepts manifested in titles are ignored. Our framework uses both frequency of a phrase across corpus and the contextual occurrences to extract high-quality phrases. We model the problem of extracting phrases as weighted frequent phrase mining. We use the significant score from TopMine Framework to guide our segmentation. After obtaining bag-of-phrases representation, we use multiple heuristics to rank sentences and extract the phrase(s)/sentence(s) that best describe the research problem a publication

intends to solve. We introduce a neighborhood based algorithm to merge different wordings of the same problem using neighbourhood information. To the best of our knowledge, there is no existing work which mines the research problems from publications.

2.3 Properties of Titles and Abstracts

In this section, we examine the properties observed by titles and abstracts of publications which are used later in our algorithm to extract research problems. We chose to work with titles and abstracts instead of entire publications content because:

1. Titles and abstracts contain sufficient information for majority of the cases.
2. To reduce the dimensionality of data.

It is important to observe that scientific publications loosely adhere to a standardized structure. For extracting research problems, we focus only on the titles and abstracts of publications. The main intent of a title is to help the reader determine the novelty/relevance of the publication. Titles can be categorized as follows.

- **Descriptive** - describing the publication's content e.g *Mining Association Rules from Large datasets*, *Mining Research Problems from Scientific Literature* etc.
- **Interrogative** - pique reader's interest by posing an interesting question e.g *Interacting Viruses in Networks: Can Both Survive?* etc.

In the domain of computer science, majority of the publications have descriptive titles. Descriptive titles are valuable because there is a good probability that research problem manifests in the title. Even in cases that it doesn't, we use the concepts in titles to guide our research problem extraction algorithm. In statistical methods, concepts manifested in title aren't detected by these methods because they heavily depend on frequency of the concept, but not on the context of occurrence. As mentioned in section 1, different authors use different wordings to represent the same problem, reducing their frequency and detection probability. It is essential to merge these different wordings into a research problem.

Similarly, the main intent of an abstract is to provide a summary of the publication. An abstract contains one or more segments which belong to one of the categories listed below:

- Introduce the research problem that the publication intends to solve, if it's a fairly new research problem or new interpretation of existing research problem.
- Explain the importance of solving that particular problem.
- Briefly comment on the related work highlighting differences between existing approach and the proposed approach.

- Explain the solution proposed in an abstract manner.
- Overview of the experiments conducted and results obtained.

It is important to note that not all abstracts follow the above structure. Our algorithm doesn't depend on abstracts having exact structure, but it gives better results for publications whose abstracts conform to this structure.

Chapter 3

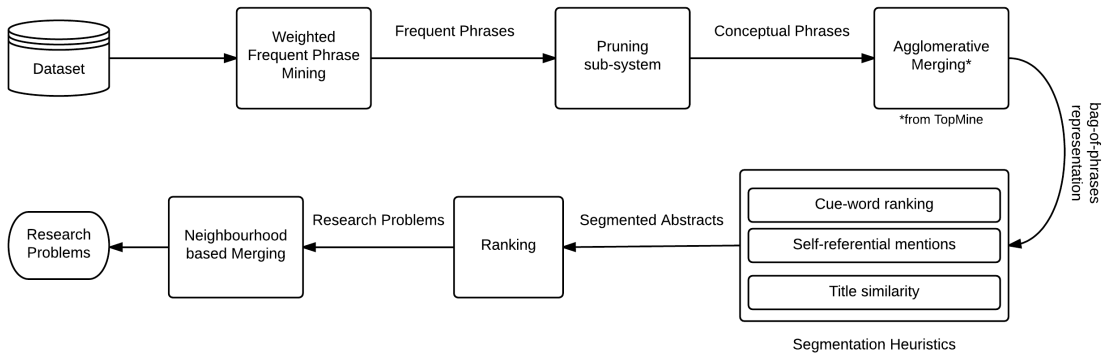
Mining Research Problems

In this chapter, we describe in detail the various stages of our framework to mine research problems. As mentioned in Chapter 1 previously, our framework can be broadly divided into the following stages:

1. Corpus Preprocessing
2. Phrase Generation
3. Phrase Pruning
4. Neighborhood Merging

Figure 3.1 shows the architecture of our framework in more detail. The core idea of our framework is that *research problems* are manifested as a phrases in publications. To extract these research problems, we propose two-phase weighted frequent phrase mining, a variant of frequent pattern mining to detect candidate phrases from publications. Once all the phrases have been extracted, we use an agglomerative merging algorithm to turn documents into bag-of-phrases representation. Then we use a combination of heuristics based on word embeddings, frequency statistics and natural language

Figure 3.1: Architecture of our framework



properties to segment documents. We further use this segmented information to filter out extraneous phrases. After obtaining these bag of phrases, we construct a phrase neighbourhood graph, and apply a variant of TextRank algorithm to merge similar wordings of phrases.

We start the chapter by elaborating on the practical difficulties faced in pre-processing stage, extracting text from PDF documents. We then discuss the motivation and the approach of two-phase weighted frequent pattern mining in the following section. Then, in the next section, we describe in detail the multiple heuristics used to segment the documents, followed by the last section in which we describe the phrase merging algorithm briefly.

3.1 Pre-processing

As mentioned in section 3, majority of the scientific literature is available in PDF. First, we need to pre-process the entire corpus into machine-understandable text, which as mentioned in Chapter 1 is not trivial. There are a plethora of open-source tools e.g. which can be employed to do the conversion. Due to wide variety of formats/layouts of publications, it turns out majority of open-source tools are only good for processing a subset of publications. For the purposes of our work, we focus on the following two important properties for evaluating PDF to text conversion:

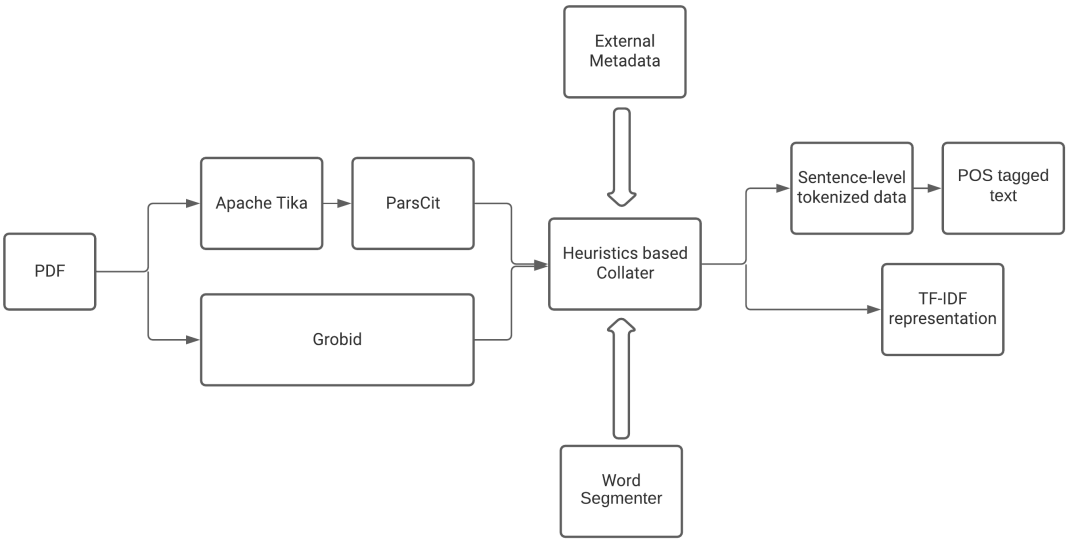
1. Introduction of spurious lines
2. Layout detection
3. Whitespace handling.

We specifically focus on these properties because they alter the definition of sentences, which causes problems in Frequent Phrase Mining, a step further down in the framework. In some cases, improper layout detection can scramble words across different sentences which alters the results of Part-Of-Speech tagger. Similarly, improper whitespace handling leads to either aggressively mixing up contiguous words or splitting words into incoherent units. To handle these issues, we introduce a pdf-extraction process which produces cleaner results for PDF to text conversion. Figure 3.2 highlights this process in detail.

Tika is an open source content detection and analysis toolkit backed by Apache Software Foundation. It uses mime-detection to identify the appropriate parser (for pdfs, default parser is Apache PDFBox) to extract content and metadata. We use Tika's Tessaract parser, an optical character recognition parser which is considered to be state-of-art open source OCR parser which uses LSTM models. To improve the quality of Tessaract parser, each publication was divided into images, (corresponding to pages), thresholded using openCV an CRF based tool which does high-level named entity recognition and identifies the citation context and other stuff

As good as Apache Tika is, in our experiments, we observed that it doesn't perform well in cases where mathematical-equations appear in the text and two-column layouts. To rectify this, we use

Figure 3.2: Pipeline to extract text from PDF



Grobid is an open source machine learning library which employs a cascading CRF models to do automatically extract metadata and structured content exraction from PDF. We use the fulltext model which includes layout analysis and semantically categorize the publication into different sections e.g. title, sections and paragraphs etc. We skip utilizing the built-in external bibliography validation service. In our experiments, we found Grobid is exceptionally good at maintaining the sequentiality of paragraph text even in non-trivial layouts, which becomes critical for ensuring the quality of POS tagger.

After obtaining the text from both parsers, we need to merge the data. We use heuristics to determine the better quality sentence and select it. we use an DBLP metadata for validation . We also use a wordsegmenter to resolve garbled text. which is based on Google Trillion Words Dataset. we need to clean it by doing the usual stuff like stemming, lemming, lemmatization, Since we also use POS tags down the pipeline, we also use Stanford POS tagger to tag the corpus.

3.2 Phrase Generation

As mentioned above, inorder to find research problems we first find phrases. Before we describe the algorithm, not all phrases are same, they're different types of phrases.

3.2.1 Conceptual Phrases vs Non-Conceptual Phrases

In Section 1, we defined concept as a word or phrase which has semantic relevance within the context of the publication. From Section 3, we know that research problems are often manifested in titles and abstracts of publications. Therefore in cases where we can represent research problem as a word or phrase, it is sufficient to model research problem as an n-gram (or a *phrase*). As a first step, we start out with identifying all phrases.

Table 3.1: Conceptual Phrases vs Non-Conceptual Phrases

in this paper	machine learning
our approach	hierarchical clustering

It is important to notice the different type of phrases that occur in a publication. Consider the list of phrases presented in Table 3.1. Phrases like "in this paper" "our approach" are valid phrases but not valid concepts. We call them as non-conceptual phrases. On the other hand phrases like "machine learning" "hierarchical clustering" are valid phrases as well as valid concepts. These phrases are defined as conceptual phrases. As mentioned in Section 1, these conceptual phrases may be of different types i.e. research problems, algorithms proposed/referred,input data attributes etc. In order to find research problems, we first need to find all conceptual phrases.

3.2.2 Phrase Quality

3.3 Frequent Pattern Mining

Frequent Pattern Mining

Given a database D with transactions t_1, t_2, t_3, ...t_n, where each transaction t_k = {i_x, i_y, i_z ...} where i_1, i_2, i_3, ...i_p are items occurring in the database D, find all patterns P (k-itemsets) that are present

3.4 Phrase Mining using Frequent Pattern Mining

To find all conceptual phrases, we use a variant of frequent pattern mining algorithm.

Frequent Phrase Mining

Given a database D with transactions t_1, t_2, t_3, ...t_n, where each transaction t_k = {i_x, i_y, i_z ...} where i_1, i_2, i_3, ...i_p are items occurring in the database D, find all patterns P (k-itemsets) that are present in at least a fraction s of the transactions.

Over the years, many approaches have been proposed for mining frequent patterns. Agarwal and Srikant introduce $Apriori$ algorithm[4], which mines frequent patterns level-wise by applying **downward closure property**. Downward closure property states that a k-itemset is frequent only if its

subsets are frequent. So $Apriori$ algorithm generates all candidates for 1-itemsets, filters out the frequent ones and use them to generate 2-itemsets and so on. $Apriori$ is considered to be inefficient because it generates huge, nearly exponential number of candidates to find all the frequent patterns. We can modify $Apriori$ algorithm for finding phrases (n-grams) efficiently.

It is important to observe that phrase is defined within a sentence i.e. words at the end of a sentence and words at the beginning of the following sentence are not candidates for valid phrases. By applying this property, the number of candidates (here candidates for phrases) can be reduced from exponential to linear. Similarly, we can apply the downward closure property for phrases as follows:

Downward-Closure Property for Phrases.

- A phrase P is frequent only if its constituent phrases p_1, p_2, p_3 ...are frequent.
- A phrase P is defined as a set of contiguous words w_1, w_2, w_3 ...w_n within a sentence.

By applying this variant of $Apriori$ algorithm, we can find most of the conceptual phrases manifested in titles and abstracts of publications. In further sections, we use the term *frequent phrase mining* to refer to this variant of frequent pattern mining by applying aforementioned downward closure property.

3.4.0.1 disadvantages of frequent pattern mining

The main problem with the above approach is mentioned below:

1. Unable to detect non-frequent conceptual phrases.

In the next subsections, we propose a simple yet efficient variant on frequent phrase mining using weights to guide the detection of non-frequent conceptual phrases.

3.5 Detecting non-frequent conceptual phrases

As explained in the above subsections, we develop a simple yet efficient method that detects non-frequent conceptual phrases. The trivial approach to do this is to decrease the support, which not only increases the probability of detecting non-frequent conceptual phrases but also increases the noise i.e frequent non-conceptual phrases. Through experiments, we found that majority of these non-frequent conceptual phrases are manifested in titles, even more so in the case of descriptive titles. Detecting conceptual phrases from titles solves most instances of non-frequent conceptual phrases problem. We can solve this problem by incorporating one of the following constraints to frequent phrase mining.

1. Decompose the problem into two separate sub-problems i.e extract conceptual phrases from titles and abstracts independently.

2. Model each publication as a single transaction, where some items have more weight than others.

3. Model each publication as a set of transactions, where some transactions have more weight than others.

In (1) since extraction of conceptual phrases from titles and abstracts is independent of one another, we can set different support for each sub-problem i.e. low support for titles to find missing non-frequent conceptual phrases. The problem with this approach is that frequency statistics are miscalculated, conceptual phrases detected in titles are not recognized as conceptual phrases in abstracts and vice versa. Even so, some non-frequent conceptual phrases do not get discovered because they are frequent only in a publication but not in the corpus (i.e. author used a different wording etc). Apart from that, as we see in the next sub-section, these frequency statistics are vital because they are used to segment the corpus into a bag-of-phrases model. In (2) modelling a transaction with weighted items seems intuitive, but attaching weights to items violates downward closure property. There's been considerable work done on extracting weighted association rules under these assumptions. Philip S.Yu developed an algorithm to mine weighted association rules which extracts weighted association rules by defining weighted downward closure property[45]. We can adapt Philip's work to define a variant of weighted downward closure property, but it's inefficient. In standard frequent phrase mining, downward closure property enables mining frequent phrases efficiently by reducing the search space for candidate phrases but that's not the case for weighted downward closure property. In (3) we model a publication as a set of transactions, including title as one of transactions, the rest being sentences. We can consider each sentence as a transaction because we're mining for phrases and phrases occur within a sentence. As mentioned above, title's transaction is assigned more weight than the others. In this approach, the downward closure property holds, so we can mine for phrases efficiently. Following the approach outlined in (3), we formally define weighted frequent pattern mining problem below and use it to mine non-frequent conceptual phrases.

Weighted Frequent Pattern Mining.

Given a database D with transactions t_1, t_2, t_3, ...t_n, with corresponding weights w_1, w_2, w_3, ...w_n where each transaction

t_k = {i_x, i_y, i_z ...} where i_1, i_2, i_3, ...i_p are items occurring in the database D find all patterns P (k-itemsets) that satisfy the support threshold s subjecting to the weight constraints. As mentioned above, by choosing to assign weights to transactions instead of items we can apply downward closure property for weighted frequent pattern mining. To take into account of weights of transactions, we introduce the notion of weighted support, a simple yet efficient measure which captures the problem constraints.

Weighted support

Given an itemset I occurring in the transactions set S, which consists of transactions t_1, t_2, t_3, ...with weights w_1, w_2, w_3, ...we define weighted support of an itemset as

$$w_I = \forall_{i \in S} \frac{\sum(w_i * t_i)}{\sum(w_i)}$$

Our algorithm implements the above mentioned weighted frequent phrase mining using weighted support. we set the maximum phrase length to be 7. Our algorithm uses a static weight i.e same weight for all title transactions. However, it is possible to build a classifier which classifies titles as descriptive, interrogative etc and assign weights appropriately (For example, descriptive titles should carry more weight etc). Through experiments, we found that high-quality results are obtained if the weights of titles are set to approximately 2/3rd of support threshold.

Pruning out frequent non-conceptual phrases is less of a inconvenience than it's counterpart i.e. detecting non-frequent conceptual phrases, because they get filtered off in the later stages of our framework. However in rare cases, these frequent non-conceptual phrases effect the segmentation process described in the later sections. We use a mixture of statistical measures and linguistic heuristics to evaluate both the quality and context of the phrase in order to determine whether to prune out a phrase or not.

3.6 Generating bag-of-phrases model

In this section, we use the agglomerative merging algorithm given in TopMine[18] and reproduce it below. However we get better results in terms of candidate phrases due to the procedures outlined in the above section. TopMine framework proposed by Ahmed-Elkishky and Jiwaei Han uses an agglomerative merging algorithm to iteratively merge the best possible pair of candidate phrases at each iteration guided by significance score to generate a bag-of-phrases representation. They define significance score as

$$sig(P_1, P_2) \approx \frac{f(P_1 \oplus P_2) - \mu_0(P_1, P_2)}{\sqrt{f(P_1 \oplus P_2)}}$$

where P_1 and P_2 denote individual units, $P_1 \oplus P_2$ denotes the phrase obtained by merging P_1 and P_2, $f(p)$ is phrase occurrence count, $\mu_0 = L * p(P_1) * p(P_2)$, L is total number of tokens in corpus, $p(k) = \frac{f(k)}{L}$.

This significance score computes the number of standard deviations away from the expected number of occurrences under the null model. They arrive at this significance score by assuming that the entire corpus is generated from a series of independent Bernoulli trails. A much more comprehensive study of segmentation can be found at [18]. Since we take into account of non-frequent conceptual phrases and frequent on-conceptual phrases, we obtain relatively accurate frequent statistics and a more accurate bag-of-phrases representation.

3.7 Abstract Segmentation

As mentioned in section 3, every abstract can be divided up into segments, where each segment conveys a different aspect of abstract. It is likely that research problem manifests in the segment which describes the problem that a publication intends to solve, if such segment exists. In this section, we

devise a series of linguistic heuristics to divide up abstract into various segments. Linguistic heuristics are needed to take advantage of the context in which these sentences occur.

3.7.1 Cue words

Table 3.2: Algorithmically generated Cue Words

propose	address	aim
establish	examine	formulate
focus	design	discover
demonstrate	introduce	survey
describe	extend	study
present	investigate	suggest
develop	contribute	adapt

Employing cue words is one of the many techniques prominently used in opinion mining, to find sentence's emotion in product reviews, blog posts, tweets etc. The idea is to use pre-defined opinion cue words (e.g. mildly, excellent, poor, terrible etc) to guide the algorithm in determining the sentence's intent. We transfer this approach to abstracts by defining cue words that help to divide the abstract into segments. Through experiments, we gathered the list of cue words which are shown in table 3.2. It is important to note that the choice of curated cue words affect the results of the heuristic significantly, so we validate our list of cue words using Word2Vec.

Word2Vec[34] proposed by Tomas Mikolov and Jeffery Dean uses shallow neural network models like skip-gram and CBOW learns the vector representation of words from a text corpus. Recent experiments show that Word2Vec captures linguistic regularities like meanings and associations[35]. We trained Word2Vec on DBLP dataset and selected sections of Wikipedia dumps (further details are in section 6) used the following algorithm to generate/validate the aforementioned exhaustive list of keywords.

Data: seed list of cue words U, a function f such that $f(word) = synonyms(word)$
Result: exhaustive list of cue words Q

```
F ← U;
Q ← ∅;
while F ≠ ∅ do
    frontier ←F;
    for f ∈ frontier do
        M ← MaxHeap();
        for syn ∈ s(f) do
            M ← syn;
            F ← M.cutoff();
```

Algorithm 1: Cue word generation algorithm

Cue word generation algorithm

Description. Given a seed list of cue words and a function s such that $s(word) = \{\text{SYNONYMS(WORD)}\}$ this algorithm generates exhaustive list of cue words

1. We start with the given seed list of cue words.

2. Frontier f, is a set of words, we need to examine. In other words, the algorithm terminates when frontier is an empty set. Initially f is the seed list

3. For each word in the seed list, we find it's synonyms and we add the synonyms to the frontier if and only if one of their synonyms belongs to the original seed list. We continue this until the frontier becomes empty.

In the aforementioned algorithm, the word2vec's cosine word distance which captures similarity between words is used for the function SYNONYMS(WORD). In order to generate the cue words in table 3.2, we used the words in table 3.3 as seeds and trained word2vec on the DBLP abstracts dataset and Wikipedia's computer science articles.

Table 3.3: Seed list of cue words to generate cue words presented in table 3.2

propose
aim
discover
survey
study
adapt
suggest

3.7.2 Self-referential mentions

By applying the cue-word heuristic mentioned in the above section, we identify the sentences which describe a concept e.g research problem, approach etc. with reasonable confidence. To further improve the chances of detecting the sentence(s) describing research problem, we identify the self-referential mentions in abstract. Self-referential mentions are sentences in which the publication's motivation/approach/impact is discussed. Some examples of self-referential mentions are 'In this paper', 'Our paper/framework/algorithm/approach etc.', 'We devise/introduce/develop etc.' To identify these self-referential mentions, we use P.O.S tagging to match the pronouns and verbs against the list of possible pronouns and verbs (i.e. aforementioned cue words). e.g. They/Their are less likely to be a part of sentence describing research problem compared to our, we, this etc. To avoid possible false positives in arbitrarily long sentences, the distance between the verb and pronoun is also taken into account (i.e. longer the distance, greater chance of being a false positive).

3.7.3 Title Similarity

In abstracts, it is common practice to introduce the research problem and refer to it indirectly using pronouns. Consider the following excerpt from a generic abstract *Hierarchical clustering is an important problem. Existing algorithms approach this problem by following this method, but have the following limitations. In this we solve this problem by introducing some novel algorithm which overcomes existing limitations by applying this another method.* By applying the aforementioned heuristics, we should be able to extract the third sentence, but not the sentence which describes the research problem. In this case the sentence is the first sentence. The main reason behind this the indirection introduced by pronouns. In order to accommodate for this indirection, we use the fact that descriptive title conveys the intent of the publication and we take into account of similarity with title. For each sentence in abstract, we measure the similarity between the sentence and title using cosine similarity to further improve the chance of detecting the sentence describing research problem.

Consolidating, each sentence in the abstract is evaluated by applying the above mentioned heuristics. Each sentence is assigned a score that denotes the degree that sentence satisfies the above mentioned heuristics. After conducting experiments, each heuristic is assigned a confidence denoting the likelihood that a sentence satisfying the heuristic is a true positive. We introduce the notion of score threshold, which is used to select sentence(s) which describe research problem. After obtaining the said sentence(s), we need to identify the research problem from the conceptual phrases present in the sentence(s).

3.8 Ranking Conceptual Phrases

So to summarize, each sentence in the abstract is evaluated based on (1) whether it's a self-referential mention (2) whether it has any cue words (3) similarity to title. If a sentence(s) satisfies two or more properties mentioned above, all conceptual phrases present in the sentence are considered to be candidates. In cases of ambiguity like two or more sentences satisfying a single property we order them by score, and all these sentences are considered. After obtaining the sentence(s), we need to single out the phrase representing research problem that a publication intends to solve. Since we are working on (1) bag-of-phrases representation and (2) an assumption that research problem is a phrase, we can formulate this problem as re-ordering the phrases in the descending order of relevance. We order candidate phrases by considering the following factors:

- Inverse Document Frequency (IDF)
- External Sources

As mentioned in section 3, research problems tend to have higher IDF values than other kinds of phrases. We calculate the IDF score of phrase by averaging IDF scores of words belonging to the

Table 3.4: Different wordings for same concept

mining association rules	phrase mining
association rule mining	phrase extraction
association rules extraction	quality phrase mining

phrase. Inverse document frequency is calculated using:

$$idf_{word} = \log \frac{N}{|\{d \in C : w \in d\}|} \tag{3.1}$$

where N is the total number of documents in corpus C and $|\{d \in C : w \in d\}|$ is the number of documents that word w appears. We use Word2Vec (trained on relevant datasets) to find the set of closest phrases for a given phrase P and evaluate the synonyms using IDF scores and use results to assign proportional scores to the given phrase. Phrases with a tight neighborhood of research problems will be assigned a high score than phrases with a neighborhood with conceptual phrases and these phrases in turn will be assigned a high score than phrases with a neighborhood consisting of non-conceptual phrases.

3.9 Context Merging

Reiterating definitions from previous sections, a concept is manifested as a word or phrase which has semantic relevance within the context of the publication. Due to the very nature of English language, it is quite possible that multiple words can refer to the same concept. For example, it is clear that 'mining association rules' and 'association rule mining' refer to the same concept i.e. the idea of extracting/mining phrases. It is important to note that clustering of these phrases is not trivial i.e. stemming, re-arranging order of words and removing qualifiers might work for the above case, but it doesn't for the examples listed in the following table.

Since we are using statistical methods (WFPM) and heuristics to mine research problems, we are actually mining the manifestations of those concepts through words and phrases. This leads to a huge number of research problems which makes it very important to merge these research problems. Merging these research problems also eases up the task of data exploration through research problems Our intuition is that even though representations are different, these phrases will manifest with similar neighborhood. We use this property to identify the phrases with the similar neighbourhoods and then identify the similar phrases. This allows us to merge similar phrases and also show the related research problems, which can be further used for recommender systems. For a phrase P, we define neighborhood as pair L,R where L and R are phrases obtained through the entire corpus. We say that a phrase P occurs in a context L,R with a window size w iff there exists a document such that sequence LRP exists and difference between their occurrence positions is less than w. From bag of

phrases representation, we have the list of phrases for each document. For each possible context, we iterate through all the phrases that occur in the same context. In this way we are able to group similar research problems into class.

Chapter 4

Experiments and Results

4.1 Datasets

4.1.1 Phrase Generation

From previous section, we know that we generate phrases from a dataset and use these phrases in the later stages of our framework to identify research problems. We've need a comprehensive dataset for this step because an improper dataset has skewed frequencies which leads to misidentification of conceptual and non-conceptual phrases. We've should also use the same datasets for training Word2Vec due to similar reasons. We use the following datasets in this step:

- DBLP - Huge collection of various computer science related proceedings, journal papers etc. This collection has 216K unique words and 43M tokens. This makes it a good candidate for phrase generation.

- Wikipedia dataset - Crawled collection of all computer science related articles.

4.1.2 Framework Evaluation

Since our framework operates on scientific publications, DBLP is a perfectly good dataset. However the problem with using DBLP dataset is it's huge size. As we'll see in the next section, most of our evaluation methods rely on domain experts, which means manually sifting through data. As of 2015, DBLP approximately has 529K documents, making it unfeasible to use in our experiments.

So we use small subsets of DBLP dataset organized around conferences. We've picked a list of conferences and pulled out the last few years publications for these conferences and use them as our datasets. Conferences studied are PAKDD, ICDM, ICDE, KDD, DSAA and ECML-PKDD. Further details about these datasets are in table 4.2.

4.2 Evaluation

The absence of reliable annotated data to compare against makes evaluating our framework challenging. Therefore we rely on domain experts to evaluate our framework (in our case, by evaluating the quality of research problems obtained We start off this section by analyzing framework output and then introduce evaluation metrics for research problems. We then use these evaluation metrics to analyze the decisions made in the framework.

4.2.1 Black-box Analysis

In this section, we try to analyze the framework by disregarding it's internal workings. As mentioned in section 3, for a publication, we know that our framework tries to find the research problem and if it's unable to identify a research problem, finds the statement that best represents it. So, depending upon the publication we can categorize framework's output as follows:

1. Single Phrase (which might or might not be a valid research problem)
2. Multiple phrases
3. Statements
4. No Output

It would be ideal to have the output fall into the first two categories because it implies that our framework is able to identify research problems that are manifested in the publications. However, there is also a possibility that even if the output falls into the first two categories, it might not be really the *correct* research problem. In this section, we proceed by assuming that first two categories are favorable than the other two categories. We justify this assumption by evaluating the *correctness* of research problems further in this section.

By measuring the distribution the output, we measure the following:

- applicability of the properties of titles and abstracts mentioned in section 3
- robustness of segmentation heuristics mentioned in section 4.5

In this experiment, we measure the distribution using the parameters mentioned in the previous sections across various datasets and present them in figure 4.1.

From figure 4.1, we can see the output distributions across various datasets is mostly consistent i.e. single phrases and multiple phrases form the major quadrants when compared to the other categories, which reinforces the effectiveness of heuristics. It is also important to note that in some cases, the percentage of publications returning no output is comparable to the percentage of publications returning single phrases, which also the indicates the volatility of the framework w.r.t datasets.

Figure 4.1: Output distribution across various datasets

4.2.2 Evaluating research problems

We've used a mix of graduate and under-graduate students (a total of 14 students) to participate in various tasks, which are designed to evaluate the effectiveness of framework. From the previous experiment, we observe that significant portion of our results are phrases (single or multiple). In this section we elaborate on the evaluating process using the following metrics (in the same order):

1. linguistic relevance
2. semantic relevance
3. correctness

Instead of measuring whether the framework produced the *correct* research problem, we chose to use the aforementioned metrics because they correlate with the efficiency of different sub-systems within the framework. In the further sub-sections, we define these metrics and describe the experiments conducted to evaluate research problems on these metrics.

4.2.2.1 Linguistic Relevance

We define a given phrase p to be *linguistically relevant* if it makes sense in the context of the publication i.e a conceptual phrase. By measuring the linguistic relevance of phrases, we are effectively measuring the impact of weighted frequent pattern mining, pruning and agglomerative merging sub-systems.

4.2.2.2 Semantic Relevance

We define a given conceptual phrase cp to be *semantically relevant* if the cp actually belongs to the set of research problems. Semantic relevance of a conceptual phrase depends on the validity of segmentation heuristics.

4.2.2.3 Correctness

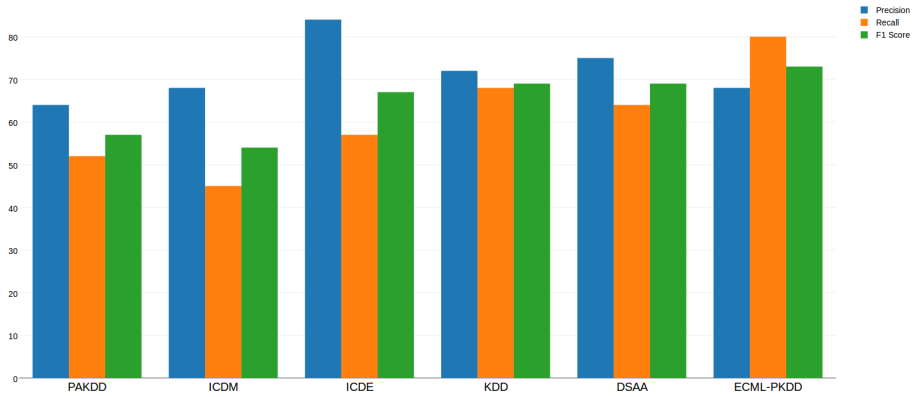
We define a given research problem rp to be *correct* for the publication p iff rp actually represents the research problem the publication p is trying to solve. Unlike the other two metrics, *correctness* is dependent on the entire framework.

To evaluate correctness of research problems, we calculate precision and recall of the framework and verify whether the research problems obtained are interpretable.

Precision-Recall Task

Precision is defined as number of 'correct' research problems divided by total number of research problems. Recall is defined as number of 'correct' research problems by total number of research problems. Since 'research problem mining' is at a nascent stage, there are no appropriate baselines to

Figure 4.2: Precision-Recall scores for various datasets



compare against, but this task reveals the effectiveness of our approach. The results are presented in Figure 4.2.

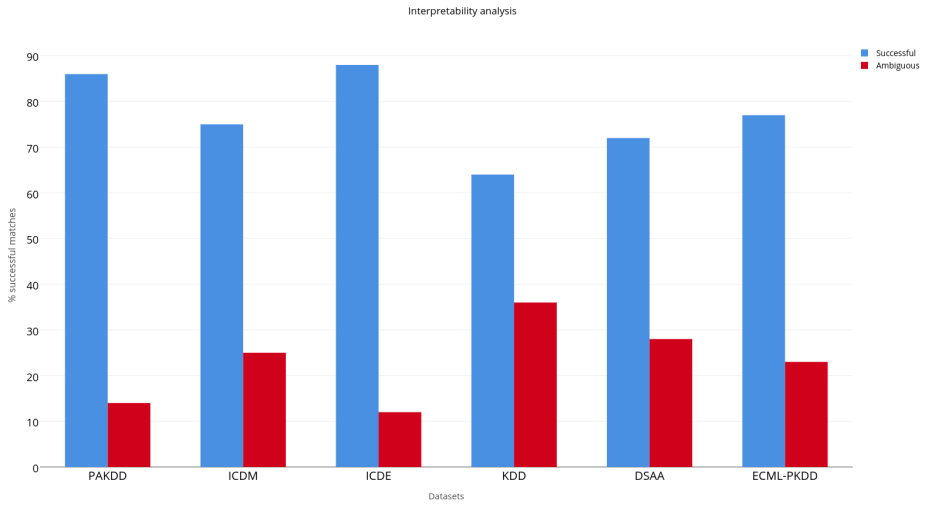
From Figure 4.2, we can observe that our framework achieves promising results. However, the actual values of precision-recall are significantly varying across different datasets, for examples consider the recall values of ICDM dataset (0.45) to ECML-PKDD dataset (0.80), precision values of PAKDD dataset (0.64) to ICDE dataset (0.84) and the precision recall pairs for ICDM (0.68,0.54), KDD (0.72,0.69) and ECML-PKDD (0.68,0.73) datasets. We hypothesize this variance is due to the volatile dependence on title's type and indirect references to research problems. Indirect references means the use of pronouns to refer to the research problems mentioned in the above sentences. Our framework ignores these kind of research problems because, the occurrence of these indirect references varies greatly from one publication to another, handling it without human interpretation generates lot of noise. Regarding the volatile dependence on title, we can build a classifier to distinguish descriptive titles and use weighted support for those titles.

Further inspection shows that linguistic complexities like conjunctions in titles and abbreviations also cause problems. Even in cases where abstracts conforms to the structure mentioned in section 3 our framework fails to recognize abbreviations as research problems (Dynamic Time-Warping is same as DTW, CBA is same as classification based association etc) because we're dealing with exact manifestations. For example consider the title *classification and pattern discovery of mood in weblogs* which yields only *pattern discovery* as research problem, instead of more appropriate *mood discovery* which is manifested in abstract. Another example is with title *classification and novel detection in data streams with active mining* which yields generic answer *novel detection*, instead of more specific *data stream classification* etc.

Interpretability task

In this task, experts are divided into two study groups. One group is given a large pool of research problems and small number of publication titles and are expected to match research problems to cor-

Figure 4.3: Results for Interpretability user study



responding titles or indicate ambiguity whereas the other group is given a small number of research problems and large pool of publication titles and are asked to do the same. In case of ambiguity, domain experts are given an option to use publication's abstract for performing a match. The main motivation for this task to evaluate whether research problems obtained are interpretable and relatable to the publication. For every pair of *{research problem, publication}*, we consider a research problem to be a relatable if either one of the groups makes a successful match. For each dataset, we report the number of successful and ambiguous matches which correlate with interpretability of research problems. The results of this user study are presented in Figure 4.3. From the Figure 4.3, we can see majority of research problems are indeed interpretable and therefore capture the semantics. Upon further inspection, we found that significant portion (38%) of ambiguous matches in KDD dataset are not non-conceptual phrases but generic conceptual-phrases.

4.2.3 Similarity based Evaluation

Evaluation by calculating precision-recall, as outlined in the previous section is a good way to measure the effectiveness of our approach but it's not exhaustive. While using precision-recall to evaluate results isn't an uncommon practice, it's not sufficient because 1) no baselines to compare against 2) doesn't represent insights of our approach. The novelty of our approach lies in the fact that the results i.e. research problems captures the semantics of a publication. To illustrate this, we calculate the similarity between publications using research problems and compare them against other similarity measures. We consider publications to be similar iff they have similar research problems. We use k-means algorithm with the following similarity measures:

- Cosine similarity

Table 4.1: Random sample of research problems obtained from DSAA'15 & PAKDD '12 datasets

dna sequence reconstruction	news recommendations	query answering
predictive modelling competitions	service descriptions	k-nearest neighbor
link prediction	web service discovery	proximity weighted synthetic oversampling
n-queens problem	active learning	quadratic correlation
reinforcement learning agent	semi-supervised associative classification	mining class imbalanced rules
tensor factorization	optimal specificity under perfect sensitivity	correlated time series
multi-label classification	meta-clustering	activity recognition
imbalanced multi-label classification	propagation structure	collaborative tweet ranking
item-set approximation	analyzing phising	delaunay triangulation
constrainted binary matrix factorization	hierarchical network environment	sparse collaborative filtering
community detection	signature analysis	semi-supervised clustering
ensemble clustering	privacy preserved data mining	scalable random
semi-randomized hashing	reverse nearest neighbors	collaborative anomaly detection
dynamic adaptive multi-tree search	business location planning	almond-dg model
location-promotion	influence maximization in social network	term translation
location-based social networks	sentiment detection	maximum margin matrix factorizatio
relevance feedback	mining influence in evolving entities	multi-class linear
parameterized proximity measure	remote sensing image classification	subtopic mining
matrix completion	critical class sensitive active learning	n-location predicator
feature vector and function approximation	coverage patterns	self-adaptive mixture copula
fingerprinting	patent evaluation	deep web crawling
location recommendation	mine sequential patterns	stochastic blockmodel
funding collaborators recommendation	Individual mobility network	label correlation
recommender systems	overlapping community detection	markov modeling
diverse recommendations	inferring potential users	intrusion detection

- Jensen-Shannon divergence

After converting publications into vector space model, we use cosine similarity as a similarity measure in k-means clustering. Cosine similarity is defined as

$$sim_{(a,b)} = \frac{\sum_{i=1}^{n} a_i.b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} \tag{4.1}$$

We use latent dirichlet allocation to reduce the publications to probability distributions of topics, we use Jensen-Shannon divergence as a similarity measure in k-means clustering. Jensen-Shannon divergence between two probability distributions P and Q is defined as

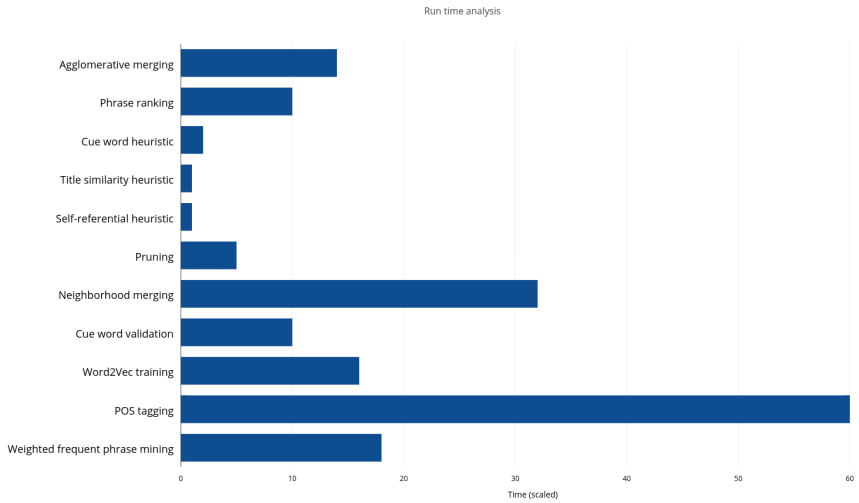
$$JSD(P,Q) = \frac{1}{2}KLD(P,M) + \frac{1}{2}KLD(Q,M) \tag{4.2}$$

where $M = \frac{1}{2}(P+Q)$ is called mid-point measure and KLD is Kullback-Leibler divergence, which is defined as

$$KLD(m,n) = \sum_i m_i \log \frac{m_i}{y_i} \tag{4.3}$$

We use F-score measure to compute the quality of clustering algorithms. The quality of our clustering solution was determined by analyzing how publications having similar research problems differ from

Figure 4.4: Runtime analysis of different sub-systems



each other and vice-versa. The primary motivation of this task is to test whether the research problems obtained capture the semantic information.

Table 4.2: Comparison of F-scores for various clustering algorithms

Dataset	# of Publications	Cosine Similarity	Jensen-Shannon Divergence	Research Problem Similarity
PAKDD	758	0.396	0.45	0.48
ICDM	327	0.324	0.462	0.574
ICDE	356	0.418	0.579	0.68
KDD	582	0.381	0.606	0.624
DSAA	113	0.52	0.469	0.457
ECML-PKDD	452	0.402	0.54	0.512

From table 4.2, we can observe that our framework achieves slightly better performance than the chosen clustering algorithms. We attribute the superior performance to domain specific modelling which helped research problems to capture the semantic information as opposed to generic k-means clustering. We can see our framework achieves similar performance to LDA based clustering. These results strengthen our notion that research problems can be used to enable large number of applications like data exploration, expert detection (i.e. research problems combined with citations/publication rank analyzed w.r.t author) where research problems can be used as stand-ins for publication's meta-data.

In table 4.1, we show randomly sampled 50 research problems from DSAA '15 dataset and PAKDD '12 dataset to give insight into the quality of research problems obtained (Note: multiple research problems per publication are also included)

4.3 Runtime Analysis

In Figure 4.4, we show runtime of different systems in our framework. We can see that majority of runtime is spent in POStagging (i.e. external library) followed by neighborhood merging, which can be easily parallelized, further reducing runtime. Overall, our framework has runtime compared to topic-modelling algorithms, where majority of time is spent in iterations to estimate parameters. All the values are measured on a machine with 16 cores @ 2.8Ghz with 32GB of memory. We used C++ to implement the intensive tasks like weighted frequent pattern mining. Rest of the framework is implemented in Python. We use Stanford POS tagger v 3.5.2 to tag the entire corpus. We use Porter stemming algorithm to stem the words in frequent pattern mining and while calculating the cosine similarity. The entire process takes about 2.2 hrs time on the DBLP dataset (529K titles and abstracts, 48M tokens) excluding POS tagging.

Chapter 5

Conclusions

In this paper, we introduce the problem of mining research problems from scientific literature by using domain related observations. We define the notion of weighted support to mine weighted frequent itemset efficiently without compromising downward closure property. Our framework uses linguistic heuristics to find the statement which describes the research problem and extract research problems from it. We use a novel neighborhood based algorithm to merge different representations of the same research problem. Future work includes applying similar methods to the sections 'Future Work' and 'Conclusion' to find the open problems in a research field.

However the work presented in this paper is not independent, it is a first, yet significant step towards building vertical search systems for the domain of scientific literature. Such systems leverage domain knowledge like the research problem being solved, the approach used to solve the problem etc. The goal of our research is to build systems which complements the existing mental models of users. e.g (find all publications which solve hierarchical clustering using frequent pattern mining or find all publications which solve phrase mining using graph based approaches etc). This process of extracting research problems is fundamental to above described systems.

Related Publications

- Chanakya A, Vikram Pudi: “Mining Research Problems from Scientific Literature”, in proceedings of 3rd IEEE International Conference on Data Science and Analytics. DSAA 2016, Montreal, QC, Canada.

Bibliography

[1] *The Google Knowledge Graph*, 2016. https://kdd2018tutorialt39.azurewebsites.net/KDD%20Tutorial%20T39.pdf.

[2] *Google Scholar*, 2016. https://scholar.google.com.

[3] A. Abu-Jbara, J. Ezra, and D. R. Radev. Purpose and polarity of citation: Towards nlp-based bibliometrics. In L. Vanderwende, H. D. III, and K. Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 596–606. The Association for Computational Linguistics, 2013.

[4] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI/MIT Press, 1996.

[5] W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. Ha, R. Kinney, S. Kohlmeier, K. Lo, T. Murray, H. Ooi, M. E. Peters, J. Power, S. Skjonsberg, L. L. Wang, C. Wilhelm, Z. Yuan, M. van Zuylen, and O. Etzioni. Construction of the literature graph in semantic scholar. *CoRR*, abs/1805.02262, 2018.

[6] T. Baldwin and S. N. Kim. Multiword expressions. In N. Indurkhya and F. J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*, pages 267–292. Chapman and Hall/CRC, 2010.

[7] H. Bast and C. Korzen. A benchmark and evaluation for text extraction from PDF. In *2017 ACM/IEEE Joint Conference on Digital Libraries, JCDL 2017, Toronto, ON, Canada, June 19-23, 2017*, pages 99–108. IEEE Computer Society, 2017.

[8] Ø. R. Berg, S. Oepen, and J. Read. Towards high-quality text stream extraction from PDF. technical background to the ACL 2012 contributed task. In R. E. Banchs, editor, *Proceedings of the Special Workshop on Rediscovering 50 Years of Discoveries@ACL 2012, Jeju Island, Korea, July 10, 2012*, pages 98–103. Association for Computational Linguistics, 2012.

[9] D. Blei and M. Jordan. Variational inference for dirichlet process mixtures. *Journal of Bayesian Analysis*, 2006.

[10] D. M. Blei and J. D. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*, pages 147–154, 2005.

[11] D. M. Blei and J. D. Lafferty. Visualizing topics with multi-word expressions. *arXiv:0907.1013*, 2009.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[13] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. Sung, B. Strope, and R. Kurzweil. Universal sentence encoder. *CoRR*, abs/1803.11175, 2018.

[14] M. Danilevsky, C. Wang, N. Desai, J. Guo, and J. Han. KERT: automatic extraction and ranking of topical keyphrases from content-representative document titles. *CoRR*, abs/1306.0271, 2013.

[15] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[16] Y. Ding, G. Zhang, T. Chambers, M. Song, X. Wang, and C. Zhai. Content-based citation analysis: The next generation of citation analysis. *J. Assoc. Inf. Sci. Technol.*, 65(9):1820–1833, 2014.

[17] Y. Dong, H. Ma, Z. Shen, and K. Wang. A century of science: Globalization of scientific collaborations, citations, and innovations. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 1437–1446. ACM, 2017.

[18] A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han. Scalable topical phrase mining from text corpora. *Proc. VLDB Endow.*, 8(3):305–316, 2014.

[19] S. Ganguly and V. Pudi. Competing algorithm detection from research papers. In *CODS '16*, 2016.

[20] Y. Gao, J. Liang, B. Han, M. Yakout, and A. Mohamed. *Building a Large-scale, Accurate and Fresh Knowledge Graph*, 2016. https://kdd2018tutorialt39.azurewebsites.net/KDD%20Tutorial%20T39.pdf.

[21] C. L. Giles, K. D. Bollacker, and S. Lawrence. Citeseer: An automatic citation indexing system. In *Proceedings of the 3rd ACM International Conference on Digital Libraries, June 23-26, 1998, Pittsburgh, PA, USA*, pages 89–98. ACM, 1998.

[22] T. R. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101:5228 – 5235, 2004.

[23] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In W. Chen, J. F. Naughton, and P. A. Bernstein, editors, *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*, pages 1–12. ACM, 2000.

[24] M. D. Hoffman, D. M. Blei, C. Wang, and J. W. Paisley. Stochastic variational inference. *CoRR*, abs/1206.7051, 2012.

[25] T. Hofmann. Probabilistic latent semantic indexing. In F. C. Gey, M. A. Hearst, and R. M. Tong, editors, *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*, pages 50–57. ACM, 1999.

[26] S. Huang and X. Wan. Akminer: Domain-specific knowledge graph mining from academic literatures. In X. Lin, Y. Manolopoulos, D. Srivastava, and G. Huang, editors, *Web Information Systems Engineering - WISE 2013 - 14th International Conference, Nanjing, China, October 13-15, 2013, Proceedings, Part II*, volume 8181 of *Lecture Notes in Computer Science*, pages 241–255. Springer, 2013.

[27] D. Jurgens, S. Kumar, R. Hoover, D. A. McFarland, and D. Jurafsky. Measuring the evolution of a scientific field through citation frames. *Trans. Assoc. Comput. Linguistics*, 6:391–406, 2018.

[28] S. Kataria, P. Mitra, C. Caragea, and C. L. Giles. Context sensitive topic models for author influence in document networks. In T. Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2274–2280. IJCAI/AAAI, 2011.

[29] H. D. Kim, D. H. Park, Y. Lu, and C. Zhai. Enriching text representation with frequent pattern mining for probabilistic topic modeling. In *Information, Interaction, Innovation: Celebrating the Past, Constructing the Present and Creating the Future - Proceedings of the 75th ASIS&T Annual Meeting, ASIST 2012, Baltimore, MD, USA, October 26-30, 2012*, volume 49 of *Proceedings of the Association for Information Science and Technology*, pages 1–10. Wiley, 2012.

[30] S. Klampfl, M. Granitzer, K. Jack, and R. Kern. Unsupervised document structure analysis of digital scientific articles. *Int. J. on Digital Libraries*, 14(3-4):83–99, 2014.

[31] M. Ley. The DBLP computer science bibliography: Evolution, research issues, perspectives. In A. H. F. Laender and A. L. Oliveira, editors, *String Processing and Information Retrieval, 9th International Symposium, SPIRE 2002, Lisbon, Portugal, September 11-13, 2002, Proceedings*, volume 2476 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2002.

[32] M. Lipinski, K. Yao, C. Breitinger, J. Beel, and B. Gipp. Evaluation of header metadata extraction approaches and tools for scientific PDF documents. In J. S. Downie, R. H. McDonald, T. W. Cole, R. Sanderson, and F. Shipman, editors, *13th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL '13, Indianapolis, IN, USA, July 22 - 26, 2013*, pages 385–386. ACM, 2013.

[33] R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 404–411. ACL, 2004.

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[35] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. 2013.

[36] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[37] R. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In Y. Li, B. Liu, and S. Sarawagi, editors, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 542–550. ACM, 2008.

[38] C. Ramakrishnan, A. Patnia, E. H. Hovy, and G. A. P. C. Burns. Layout-aware text extraction from full-text PDF of scientific articles. *Source Code Biol. Medicine*, 7:7, 2012.

[39] A. Ritchie. *Citation context analysis for information retrieval.* PhD thesis, University of Cambridge, UK, 2009.

[40] F. Sclano and P. Velardi. Termextractor: a web application to learn the shared terminology of emergent web communities. In R. Jardim-Gonçalves, J. P. Müller, K. Mertins, and M. Zelm, editors, *Enterprise Interoperability II - New Challenges and Industrial Approaches, Proceedings of the 3th International Conference on Interoperability for Enterprise Software and Applications, IESA 2007, March 27-30, 2007, Funchal, Madeira Island, Portugal*, pages 287–290. Springer, 2007.

[41] Z. Shen, H. Ma, and K. Wang. A web-scale system for scientific knowledge exploration. In F. Liu and T. Solorio, editors, *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, pages 87–92. Association for Computational Linguistics, 2018.

[42] K. Shubhankar, A. P. Singh, and V. Pudi. An efficient algorithm for topic ranking and modeling topic evolution. In A. Hameurlain, S. W. Liddle, K. Schewe, and X. Zhou, editors, *Database and Expert Systems Applications - 22nd International Conference, DEXA 2011, Toulouse, France, August 29 - September 2, 2011. Proceedings, Part I*, volume 6860 of *Lecture Notes in Computer Science*, pages 320–330. Springer, 2011.

[43] J. Tang, R. Jin, and J. Zhang. A topic modeling approach and its integration into the random walk framework for academic search. *2008 Eighth IEEE International Conference on Data Mining*, pages 1055–1060, 2008.

[44] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In Y. Li, B. Liu, and S. Sarawagi, editors, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 990–998. ACM, 2008.

[45] F. Tao, F. Murtagh, and M. M. Farid. Weighted association rule mining using weighted support and significance framework. In L. Getoor, T. E. Senator, P. M. Domingos, and C. Faloutsos, editors, *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, pages 661–666. ACM, 2003.

[46] S. Teufel, A. Siddharthan, and D. Tidhar. Automatic classification of citation function. In D. Jurafsky and É. Gaussier, editors, *EMNLP 2006, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 22-23 July 2006, Sydney, Australia*, pages 103–110. ACL, 2006.

[47] X. Wan and J. Xiao. Single document keyphrase extraction using neighborhood knowledge. In D. Fox and C. P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 855–860. AAAI Press, 2008.

[48] X. Wan and J. Xiao. Exploiting neighborhood knowledge for single document summarization and keyphrase extraction. *ACM Trans. Inf. Syst.*, 28(2):8:1–8:34, 2010.

[49] C. Wang, X. Liu, Y. Song, and J. Han. Towards interactive construction of topical hierarchy: A recursive tensor decomposition approach. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 1225–1234, 2015.

[50] J. Wang, X. Hu, X. Tu, and T. He. Author-conference topic-connection model for academic network search. In *CIKM '12*, 2012.

[51] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. KEA: practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM conference on Digital Libraries, August 11-14, 1999, Berkeley, CA, USA*, pages 254–255. ACM, 1999.