Scalable, Semi-Supervised Extraction of Structured Information 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

KRITIKA AGRAWAL
201502061
kritika.agrawal@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
February 2021
Copyright © KRITIKA AGRAWAL, 2020
All Rights Reserved
INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Scalable, Semi-Supervised Extraction of Structured Information from Scientific Literature” by KRITIKA AGRAWAL, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. Vikram Pudi
To my father!
Acknowledgments

I would like to take this opportunity to express my heartfelt gratitude towards IIIT Hyderabad community at large. My entire journey here has been a wonderful and perhaps the most memorable experience of all.

I am thankful to IIIT Hyderabad as an institute for providing me an awesome atmosphere to carry out research. The courses, assignments and seminar talks provide ample opportunity to students for pursuing research. In today’s world, proper computational infrastructure is a requisite for carrying out research in computer science, for solving problems involving big-data. I am much obliged to have a lot of computational resource just at my finger-tips to carry out my experiments.

This thesis would not be possible without the guidance of Prof. Vikram Pudi, who gave me his full support and complete independence to work on the problems I wanted to. He also encouraged me to keep on working at the initial stages of research when results were not obvious. In spite of having such a busy schedule, he always took out time to meet me regularly for discussions about ongoing work progress and future ideas. I would also like to thank him for allowing and encouraging me to collaborate with people outside our lab and letting me work at my comfortable pace. Working at the Data Science and Analytics lab with him changed everything for me because, during my time in the lab, I not only gained valuable domain language from him, but also learned several research skills like exploration, communication and presentation, necessary to thrive in a collaborative research environment.

Without the continued support and sacrifices from my family, I would not have completed my masters on time. Particularly, I extend my gratitude to my parents for letting me carve my own career path. I am lucky to have a large pool of friends ranging from juniors to seniors, from B.tech to PhD who made my stay in IIIT quite memorable.

I’m also very thankful to the mentors that were not directly a part of this thesis but have played a considerable role in making a better researcher.

Thank you all for all the support!
Abstract

Scientific knowledge is one of the greatest assets of humankind. This knowledge is recorded and disseminated in scientific publications, and the body of scientific literature is growing at a tremendous rate. As scientific communities grow and evolve, there is a high demand for improved methods for finding relevant papers, comparing papers on similar topics and studying trends in the research community. Whenever researchers start working on a problem, they are interested to know if the problem has been solved previously, methods used to solve this problem, the importance of the problem and the applications of that problem. Automatic methods of processing and cataloging that information are necessary for assisting scientists to navigate this vast amount of information and facilitating automated reasoning, discovery, and decision-making on that data. All these tasks involve the common problem of extracting structured information from scientific articles. This leads to the requirement to find automatic ways of extracting such structured information from the vast available raw scientific literature, which can help summarize the research paper and the research community.

This thesis focuses on processing scientific articles and creating structured repositories such as knowledge graphs to find new information and make scientific discoveries. In this thesis, we propose a novel, scalable, semi-supervised method for extracting relevant structured information from the vast available raw scientific literature. We extract the fundamental concepts of aim, method, and result from scientific articles and use them to construct a knowledge graph. The algorithm makes use of domain-based word embedding and the bootstrap framework. Our approach also makes use of citation context apart from title and abstract on which most of the work relied till now. We show how the extracted concepts and the available citation graph can be used to represent the research community as a knowledge graph. We demonstrate our method on a sizeable multi-domain dataset built with the help of the DBLP citation network. Our experiments show the domain independence of our algorithm and that our system achieves precision and recall compared to state of the art.

The tremendous amount of research publications available online aims to solve a lot of interesting problems. Some of the fields have been studied well and research problems have been solved with time. However, there are few problematic research problems which are yet not solved entirely and interests many researchers. In this thesis, we also aim to find research fields that are saturated and research fields that need to be explored yet by performing temporal analysis on top of the knowledge graph formed.
# Contents

| Chapter | Page |
|---------|------|
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Contribution | 3 |
| 1.3 Organization of the thesis | 4 |
| 2 Dataset | 5 |
| 2.1 Requirements | 5 |
| 2.2 Creation | 5 |
| 2.2.1 Source | 5 |
| 2.2.2 Extraction of Title, Abstract and Citations | 6 |
| 3 Concept Extraction | 8 |
| 3.1 Problem Definition | 8 |
| 3.2 Proposed Method | 8 |
| 3.3 Definitions | 9 |
| 3.4 Algorithm | 10 |
| 3.5 Experimental Setup | 12 |
| 3.5.1 Phrase embeddings | 12 |
| 3.5.2 Seed Feature Generation | 13 |
| 3.5.3 Other Experimental Details | 14 |
| 3.5.4 Evaluation | 14 |
| 3.6 Results and Discussions | 15 |
| 4 Graph Construction | 17 |
| 4.1 Definitions | 17 |
| 4.2 Bringing phrases with same meaning together | 17 |
| 4.3 Neo4j interface | 18 |
| 4.4 Results and Discussion | 19 |
| 5 Temporal Analysis | 21 |
| 5.1 Objectives | 21 |
| 5.2 Definitions | 21 |
| 5.3 Approach | 22 |
| 5.3.1 Finding Saturated Problems | 22 |
| 5.3.2 Finding Grand Problems | 22 |
CONTENTS

5.4 Results and Discussion ..................................................... 23

6 Related Work ................................................................. 27
  6.1 Concept Extraction ....................................................... 27
  6.2 Literature Graph .......................................................... 28
  6.3 Temporal Analysis of Scientific Literature .......................... 28

7 Conclusion and Future Work ................................................ 29
  7.1 Conclusion ................................................................. 29
  7.2 Future Work ............................................................... 29

8 Related Papers ................................................................. 31

Bibliography ......................................................................... 32
List of Figures

| Figure | Page |
|--------|------|
| 1.1 Proposed Graphical Representation | 2 |
| 3.1 Proposed Method | 9 |
| 3.2 Algorithm | 10 |
| 3.3 Classification of phrase | 11 |
| 3.4 Classify Phrase | 12 |
| 3.5 Generating Feature Set | 13 |
| 4.1 Sample from our constructed graph. Green nodes correspond to research papers and brown nodes correspond to extracted phrase entities. | 19 |
| 5.1 Word Cloud for Grand Challenges | 25 |
| 5.2 Word Cloud for Saturated Problems | 26 |
List of Tables

| Table | Page |
|-------|------|
| 3.1 f1, precision & recall score for AIM concept | 15 |
| 3.2 f1, precision & recall score for METHOD concept | 15 |
| 3.3 Comparison with state-of-the-art for METHOD Concept | 16 |
| 3.4 Comparison with state-of-the-art for AIM Concept on DBLP dataset | 16 |
| 3.5 Comparison with state-of-the-art for Method Concept on DBLP dataset | 16 |
| 5.1 Top 10 Grand Challenges and Saturated Problems | 23 |
Chapter 1

Introduction

1.1 Motivation

With the tremendous amount of research publications available online, there is an increasing demand to automatically process this information to facilitate smooth navigation through this enormous literature for researchers. Whenever researchers start working on a problem, they are interested to know if the problem has been solved previously, methods used to solve this problem, the importance of the problem and the applications of that problem. This leads to the requirement of finding automatic ways of extracting such structured information from the vast available raw scientific literature, which can help summarize the research paper as well as the research community and can help in finding relevant papers. Organizing scientific information into structured knowledge bases requires information extraction (IE) about scientific entities and their relationships. However, the challenges associated with scientific information extraction are more significant than for a generic domain. General methods of information extraction cannot be applied to research papers due to their semi-structured nature and also the new and unique terminologies used in them. Secondly, the annotation of scientific text requires domain expertise, which makes annotation costly and limits resources.

In this work, we propose constructing a scientific knowledge graph from the vast available scientific literature by extracting a set of phrases categorized as aim, method, and result from scientific articles in a semi-supervised manner.

Graph representations like knowledge graph that link the information of a large body of publications can reveal patterns and lead to the discovery of new information that would not be apparent from the analysis of just one publication. Analysis on top of these representations can lead to new scientific insights and discovery of trends in a research area. They can also facilitate some other tasks like assigning reviewers, recommending relevant papers, or improving scientific search engines. Therefore, we propose to build graphical representation by extracting phrases representing the concepts Aim, Method and Result from scientific publications. We introduce these phrases as additional nodes and connect them to their corresponding paper nodes in the citation graph. We argue that the citation network is an integral part of the scientific knowledge graph and the proposed representation can adequately summarize
the research community. Proposed graph is shown in [1.1] Our main motivation behind creating this structure is:

1. To provide a visual summary of the scientific literature
2. Enhanced Question Answering on raw research papers. E.g., Which papers in the year 2019 published in NAACL have used Machine Translation as a tool to enhance their results?
3. Can be used by search engines for advanced search.
4. Can provide surveys at a glance of research topics. E.g., What are the latest developments in the field related to Word Embedding Generation?
5. The knowledge graph can be used to understand the evolution of techniques, ideas and algorithms as they transform from being the “problem at hand” to “subroutine used”. E.g., How Named Entity Recognition (NER) has changed from being a challenging problem in 2010 to a subroutine used to add linguistic information to the text in current scientific articles?
6. Knowledge graphs can help identify deeper relationships and dependencies. E.g., How the development of machine translation accelerated with improvements in word embeddings generation?

**Figure 1.1** Proposed Graphical Representation

1. To provide a visual summary of the scientific literature
2. Enhanced Question Answering on raw research papers. E.g., Which papers in the year 2019 published in NAACL have used Machine Translation as a tool to enhance their results?
3. Can be used by search engines for advanced search.
4. Can provide surveys at a glance of research topics. E.g., What are the latest developments in the field related to Word Embedding Generation?
5. The knowledge graph can be used to understand the evolution of techniques, ideas and algorithms as they transform from being the “problem at hand” to “subroutine used”. E.g., How Named Entity Recognition (NER) has changed from being a challenging problem in 2010 to a subroutine used to add linguistic information to the text in current scientific articles?
6. Knowledge graphs can help identify deeper relationships and dependencies. E.g., How the development of machine translation accelerated with improvements in word embeddings generation?
A consistently thriving global research community has, over decades, produced a colossal amount of research papers that are published online, which makes it crucial to organize this vast bulk of information systematically so that upcoming researchers can navigate through efficiently and continue to push boundaries of scientific research. Such an organization over intellectual information will not only boost the rate of further research work but also augment researchers with a better holistic view of development in research and the directions in which it is evolving into. One of the first elementary steps we take as researchers are to figure out which problems to focus on solving, and structured analysis on present research status will help researchers identify critical problems and also give insight about how they developed across time. Due to this, it will be easier to realize if a particular problem has got no recent improvement in the recent past and has moved into a thriving application and so on. Analysis is the foundation of the organization of cumulative knowledge garnered by the research community in decades, and this paper deals with this first step in the direction.

1.2 Contribution

In this work, we propose a new technique to extract key concepts from the research publications. Our main insight is that a paper cites another paper either for its aim, or method, or result. Therefore, key contributions of paper in the research community can be best summarized by its aim, the method used to solve the problem and the final result. We define these concepts as:

- **Aim**: Target or primary focus of the paper.
- **Method**: Techniques used to achieve the aim.
- **Result**: Well-defined output of the experiments or contribution which can be directly used by the research community.

Example,

“Automatically describing the content of an image is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. In this paper, we present a generative model (RESULT) based on a deep recurrent architecture (METHOD) that combines recent advances in computer vision and machine translation and that can be used to generate natural sentences describing an image (AIM).”

**Contributions**: Our key contributions are:

1. We propose a novel, scalable, semi-supervised, and domain-independent method for extracting concepts, aim, method, and result from the vast available raw scientific literature by using domain-based word embeddings and data mining techniques. Our approach also takes citation context into account apart from title and abstract on which most of the work relied till now.
2. We experimentally validate our approach and show statistically significant improvements over existing state-of-the-art models.

3. We show how the extracted concepts and the available citation graph can be used to represent the research community as a knowledge graph.

4. We demonstrate our method on a large multi-domain dataset built with the help of the DBLP citation network. Our dataset consists of 332,793 papers and 1,508,560 links between them.

5. We propose temporal analysis models on top of the graph representation formed to mine saturated problems and grand challenges in the complete computer science domain.

1.3 Organization of the thesis

The thesis is organized as follows. In Chapter 2, we describe our process of creating a dataset for solving the problem. In Chapter 5, we discuss the approach, experiments, and results for solving the subproblem of extracting concepts from scientific papers. In Chapter 4, we present our constructed scientific knowledge graph. In Chapter 5, we present our work on temporal analysis on the formed knowledge graph by leveraging the graph to find grand challenges and saturated problems in the computer science research domain. In Chapter 6, we introduce the various existing approaches for extracting concepts from scientific articles and existing literature graphs. We finally conclude the thesis and discuss future directions in Chapter 7.

In 1.1, we understand the need and application of extracting structural information from scientific literature. We then summarized the problem statement and the contributions in 1.2 along with the thesis organization in this section.
Chapter 2

Dataset

2.1 Requirements

For the purpose of our experiments, we required a dataset which satisfies the following criteria:

1. It contains research papers from various domains. This is to make an algorithm which is domain independent.

2. It contains citation information or citation network because we need to extract citation contexts for each paper. Citation context is the text around the citation marker. We also want to preserve the citation graph in our graph representation of the scientific literature to allow more detailed analysis.

3. For each paper, we require:
   - Authors Information
   - Year of publication
   - Venue of publication
   - Title
   - Abstract
   - Citation Contexts

2.2 Creation

2.2.1 Source

We used the DBLP Citation Network (version 7) dataset as our basis. This dataset is an extensive collection of computer science papers. It contains 2,244,021 papers and 4,354,534 citation relationships. DBLP only provides citation-link information and paper titles. For the full text of these papers,
we refer to a recent research by Zhou et al[28], which has been crawled from CiteSeerX and is publicly available. This dataset is partly noisy with some duplicate paper information, and there is a lack of unique one-to-one mapping from the DBLP paper ids to the actual text of that paper. We either pruned out ambiguous papers or manually resolved the conflicts during the creation of our final dataset. We came up with a final set of 429,373 papers from the DBLP corpus for which we have full text available. In this set, only 224,836 papers are connected by citations because most of the other cited links are outside DBLP (not from the computer science domain), and hence full text is not available. Since we need papers connected via citation relations, we prune our dataset by taking only the largest connected component in the citation network while considering the links to be bidirectional. We get 332,793 papers having 1,508,560 citation links.

2.2.2 Extraction of Title, Abstract and Citations

We used ParsCit[20] for extracting different sections from the papers. It extracts the following from each paper:

- Different sections and subsections
- Abstract
- Weblinks
- Author List
- Affiliation
- Citations with corresponding context

Parscit gives output in XML format. It produced zero length output for 82,992 papers and corrupted or incomplete XML for 10,742 papers.

DBLP is a computer science bibliography website. The whole DBLP dataset[2] is available as one big XML file. It contains all bibliographic records, including title, authors, year, abstract, conference for each paper in the dataset. We have the DBLP ids for each paper from DBLP Citation Network. Due to less confidence in Parscit output, we take the title, abstract, year and conference from the DBLP dataset for each paper. Papers for which fields were missing in the DBLP dataset, we extract those fields from Parscit output.

We also need the citation context for each paper. Parscit extracts citations/references from each paper. We had to map the citation context to the paper it belongs to. For this, we used the entity-citation linking algorithm [8]. The matching function iterates over entities and citations to get their closeness score. After the scoring step, a two-step pruning is performed. It first takes all the citations and keeps a list of the closest entity per citation. Then it takes the remaining entities and keeps only the closest

[https://dblp.uni-trier.de/xml/](https://dblp.uni-trier.de/xml/)
citations per entity. Finally, we get a list of tuples where each element contains a unique entity matched with its citation. Only the entities which are present in this list of tuples are considered as candidate phrases.
Chapter 3

Concept Extraction

3.1 Problem Definition

Given a target document \( d \), the objective of the concept extraction task is to extract a list of words or phrases which best represent the aim, method and result of document \( d \). Prior work has solved the problem of extracting keyphrases and relations between them as a sequence labelling task. However, due to the nonavailability of large annotated data for this purpose limits this approach. Also this approach does not take advantage of the fact that more than 96 percent of phrases that form aim, method and result are noun phrases [3]. Since we already have a defined set of candidates for the key phrases, we attempt this problem as a multiclass classification problem. Given a document, we classify its phrases as Aim, Method, Result.

3.2 Proposed Method

We extract these concepts from title, abstract and citation contexts of a research paper. These sections can be accurately automatically extracted from research papers. Title and abstract work as a short and to the point summary of work done in the paper. They are an essential place to find the exact phrases for these concepts without the introduction of too much noise. Citation context is the text around the citation marker. This text serves as “micro summaries” of a cited paper and phrases in this text are important candidates for the aim, method or result of the cited paper. We combine data mining and natural language techniques to solve the problem scalably in a semi-supervised manner.

Our approach is built on the observation that the semantics of the sentence of document \( d \) containing a phrase belonging to any of the concept types is similar across research papers. To capture this semantic similarity, we use \( k \) nearest neighbour classifier on top of state-of-the-art [6] domain based word embeddings. We start by extracting features from a small set of annotated examples and use bootstrapping [11] for extracting new features from unlabeled dataset. 3.1 shows our pipeline.
3.3 Definitions

Following are some of the terminologies which will be used throughout the paper that follows:

- **Candidate phrases**: Phrases present in the target document $d$ which will be considered for labeling.
- **Concept mention**: Phrases labeled as either aim, method or result in the labeled dataset.
- **Parent sentence of a phrase $p$**: The original sentence in target document to which the candidate phrase/concept mention $p$ belongs to.
- **Left context phrase $(S, p)$**: The part of the parent sentence $S$ before the occurrence of the candidate phrase $p$ or concept mention.
- **Right context phrase $(S, p)$**: The part of the parent sentence $S$ after the occurrence of the candidate phrase $p$ or concept mention.
- **Left Context Vectors($p$)**: Vector representations of left context phrase $p$.
- **Right Context Vectors($p$)**: Vector representations of right context phrase $p$.
- **Feature Vectors**: Tuple of Left and Right Context Vectors which is being used as features to label candidate phrases.
- **Feature Score**: Each feature vector has an associated feature score between 0 and 1 that represents the confidence of it being a representative of the class. Seed features have a feature score of 1.
- **Support Score of candidate phrase $p$ for class $c$**: Every phrase is assigned a support score for all classes that represents the confidence that the phrase belongs to that class.
3.4 Algorithm

Seed Feature Extraction: In this step, we extract features for each of the concept type using the small set of annotated examples. For each concept mention in the annotated examples, we construct left context vector $lev$ and right context vector $rcv$. These $lev$ and $rcv$ then form part of the features for the class to which concept mention belongs to. Phrase embeddings are generated using a pre-trained BERT model [6] fine-tuned on DBLP research papers dataset. Details of BERT training and datasets used for seed feature extraction are given in the Experiments Section.

Candidate Phrase Extraction: To limit the search space of phrases, we propose to use noun phrases present in the Title and Abstract of document $d$ as candidate phrases. For citation contexts, named entities form a better set of candidates as shown by [8]. However different named entities can be linked to different papers cited in the same citation context. So it becomes essential to first identify which entity $e$ corresponds to which cited paper $cp$ and then use the proposed algorithm to classify $e$ as aim/method/result for the corresponding paper $cp$. For the above purpose, we use entity-citation linking algorithm [8]. The matching function iterates over entities and citations to get their closeness score. After the scoring step, a two-step pruning is performed. It first takes all the citations and keeps a list of the closest entity per citation. Then it takes the remaining entities and keeps only the closest citations per entity. Finally, we get a list of tuples where each element contains a unique entity matched with its citation. Only the entities which are present in this list of tuples are considered as candidate phrases.

Labeling Candidate Phrases: For labeling candidates in iteration $i$, we use $k$ nearest neighbour algorithm ($k-NN$). The algorithm for labeling candidate phrases is presented in Algorithm 1.
1. For each sentence $s$ in document $d$ in the dataset, $p \leftarrow$ unlabeled Phrase in sentence $s$.
2. Let $lcv$ be the left context vector and $rcv$ be the right context vector corresponding to phrase $p$ in sentence $s$.
3. Find nearest neighbours of $lcv$ and $rcv$ from the feature vectors that are at most distance $r$. Let the nearest neighbours corresponding to $lcv$ be $lnn$ or left nearest neighbours and $rcv$ be $rnn$ or right nearest neighbours.
4. If the size of both $lnn$ and $rnn$ is less than the minimum number of neighbours required for classification $k$ then the phrase can not be labeled in this iteration and we move to the next phrase.
5. Else we take $k$ nearest neighbours for both the $lcv$ and $rcv$ and the support score of the phrase for class $c$ is calculated as follows:

$$N = \{ n | n \in \text{Top k Neighbours of } lcv \text{ or } rcv \text{ and } \text{label}(n) = c \}$$

$$\text{supportScore}(p, c) = \sum_{n \in N} \text{featureScore}(n)$$

6. Then the predicted class for phrase $p$ is $\arg \max_c \text{supportScore}(p, c)$.

Algorithm 1: Label Candidate Phrases

Finally after $T$ iterations, unlabeled candidate phrases are discarded.

![Diagram](image)

Figure 3.3 Classification of phrase
Extraction of new features: For each phrase \( p \) assigned class \( c \) in any of the iterations, we generate context vectors \( lcv \) and \( rcv \). We define the feature score corresponding to the context vectors of phrase \( p \) labeled as class \( c \) as:

\[
featureScore(p) = \frac{\text{supportScore}(p, c)}{\sum_{c'} \text{supportScore}(p, c')}
\]

For each class, the context vectors are sorted based on their feature score and top 5000 are taken as feature vectors.

Final Selection: For each document, we take top \( t \) phrases (based on their \( \text{supportScore} \)) for each class as the final output of our system.

3.5 Experimental Setup

3.5.1 Phrase embeddings

For vector representation of a phrase, we use BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding as proposed in [6]. We use the pre-trained model BERT-Base, Uncased: 12-layer, 768-hidden, 12-heads, 110M parameters available publicly. We fine tune the model on our DBLP research paper dataset. Complete text of papers after cleaning has been used for the purpose of
fine tuning. The model is fined tuned on total of 20,970,300 sentences with max sequence length as 128 and learning rate as $2 \times 10^5$. For generating the phrase embedding, we use second last layer as the pooling layer with pooling strategy as reduced mean.

3.5.2 Seed Feature Generation

For the purpose of seed feature generation we use the following two publicly available datasets:

1. SemEval 2017 Task 10 dataset [3]: It contains 500 scientific paragraphs from physics, material science and computer science domain, each marked with keyphrases and each keyphrase is labelled as TASK, PROCESS and MATERIAL. The concepts of TASK and PROCESS in this dataset closely relates to our definition of AIM and METHOD. This complete dataset is used for seed feature extraction.

2. [11] introduced a dataset of titles and abstracts of 474 research publications from ACL Anthology annotated with phrases corresponding to FOCUS, TECHNIQUE and DOMAIN. Their definitions

![Figure 3.5 Generating Feature Set](image)
FOCUS and TECHNIQUE closely relate to our definitions of AIM and METHOD respectively. We divided this data into two parts- one is used as training data for seed features extraction having 277 papers and another as test data for evaluation purposes having 197 papers. These two datasets helped to build seed features for AIM and METHOD category. We removed the papers from SemEval dataset which overlapped with [11].

3. For RESULT, we manually annotated titles and abstracts of 100 research publications in the computer science domain.

### 3.5.3 Other Experimental Details

1. While generating vector encoding for context phrases, we limit the length of the context phrase to 25 in-order to handle very long sentences. We used cosine distance to measure distance between vector representation of the phrases.

2. It may be possible that there are more than one concept mention in a sentence. To nullify the effect of other concept mentions, we generated the seed features list in two ways:

   - Take the left context phrase and right context phrase and generate their vector representation. This is called as unmasked feature list.
   - We mask the other candidate phrases $C$ in the left and right context phrase of candidate $c_i$ before generating their embedding. This is called as masked feature list.

   Experiments were done for masked and unmasked feature lists separately.

3. As the number of phrases added per iteration decreased substantially after iteration 5, we ran only 5 iterations of bootstrapping algorithm for all the experiments.

4. We experimented with different values of distance $r$ and $k$. We observed that in general precision increases with increase in value of $k$ and recall increases with decrease in value of $r$.

### 3.5.4 Evaluation

For evaluating our results, we use the labeled dataset made available by [11]. We used 197 out of 474 papers for evaluation purpose. We calculate precision, recall and $f - 1$ score for each class. However, as Result phrases were not annotated in that dataset, we could evaluate only for Aim and Method. We compare our proposed approach with [25] which ran the bootstrapping algorithm for a similar problem but used n-gram based features. They reported results for ACL Anthopology Network(AAN) Corpus [22]. We ran their algorithm on our dataset with parameter tuning as mentioned by them.
### 3.6 Results and Discussions

We got the best results for parameter values, $r = 0.65$ and $k = 60$. Our bootstrapping algorithm gave output for 332,242 out of 332,793 papers. In Table 3.1 we report the top five scores for Aim for different parameters. Top ten scores for both aim and method concept were given by unmasked feature list. Therefore mask feature list results have not been shown. In Table 3.2 we report the top five scores for Method on different parameters. 3.3 and 3.4 compares our scores with that of [11] and [25]. 3.5 compares our scores with the score computed for [25] approach on our dataset. Our proposed algorithm was able to extract phrases from scientific articles in a large dataset in semi-supervised manner with $f_1$ score comparable to the state-of-the-art. Our $f_1$ score was lower as compared to [11] [25]. However, our recall was consistently higher. Our precision was perhaps low as we were considering only the noun phrases whereas such limitation was not there while annotating the test corpus. They [11] [25] used hand crafted features for AAN Corpus whereas our features were extracted algorithmically starting from a small annotated dataset containing multiple domains such as physics, material science and computer science. 3.5 shows the scalability of our approach. [25] bootstrapping algorithm could not give a decent score when ran on our multi-domain dataset because phrases could not be extracted for most of the papers.

| k  | r  | t | $f_1$ score | precision | recall |
|----|----|---|-------------|-----------|--------|
| 30 | 0.65 | 3 | 40.66 | 46.04 | 36.41 |
| 60 | 0.65 | 3 | 40.47 | 52.60 | 32.88 |
| 40 | 0.65 | 3 | 40.38 | 48.65 | 34.51 |
| 40 | 0.60 | 4 | 40.06 | 47.12 | 34.84 |
| 30 | 0.75 | 4 | 38.38 | 41.95 | 35.37 |

**Table 3.1** $f_1$, precision & recall score for AIM concept

| k  | r  | t | $f_1$ score | precision | recall |
|----|----|---|-------------|-----------|--------|
| 40 | 0.85 | 20 | 32.58 | 22.65 | 58.1 |
| 30 | 0.75 | 17 | 30.81 | 21.12 | 56.89 |
| 30 | 0.90 | 14 | 30.87 | 23.78 | 44 |
| 30 | 0.80 | 25 | 31.16 | 20.72 | 62.77 |
| 30 | 0.65 | 15 | 30.69 | 21.35 | 54.6 |

**Table 3.2** $f_1$, precision & recall score for METHOD concept
| Approach     | f1 score | precision | recall |
|--------------|----------|-----------|--------|
| GM [10]      | 30.5     | 46.7      | 36.9   |
| [25]         | 48.2     | 48.8      | 48.5   |
| Our Approach | 32.58    | 22.65     | 58.1   |

Table 3.3 Comparison with state-of-the-art for METHOD Concept

| Approach | f1 score | precision | recall |
|----------|----------|-----------|--------|
| [25]     | 8.26     | 31.37     | 4.761  |
| Our Approach | 40.66 | 46.04     | 36.41  |

Table 3.4 Comparison with state-of-the-art for AIM Concept on DBLP dataset

| Approach | f1 score | precision | recall |
|----------|----------|-----------|--------|
| [25]     | 18.0     | 50.70     | 10.94  |
| Our Approach | 32.58 | 22.65     | 58.1   |

Table 3.5 Comparison with state-of-the-art for Method Concept on DBLP dataset
Chapter 4

Graph Construction

4.1 Definitions

We build a graphical representation by using the extracted concepts and citation graph. Our graph has the following types of nodes and edges:

**Paper nodes**: These are the original paper nodes in the citation graph. Each paper node has metadata related to the paper like dblp id, title, authors, conference, year of publication.

**Entity nodes**: These nodes are the phrases extracted in the concept extraction step. Cited by relation: A cited by relation is defined between paper nodes \( p_i \) and \( p_j \) if paper \( p_i \) has cited \( p_j \).

**Aim relation**: Aim relation is defined between a paper node \( p_i \) and entity node \( e_i \) if \( e_i \) was extracted as aim concept for \( p_i \).

**Method relation**: A method relation is defined between a paper node \( p_i \) and entity node \( e_i \) if \( e_i \) was extracted as method concept for \( p_i \).

**Result relation**: A result relation is defined between a paper node \( p_i \) and entity node \( e_i \) if \( e_i \) was extracted as a result concept for \( p_i \).

4.2 Bringing phrases with same meaning together

A major challenge in the construction of graph using phrases extracted in concept extraction step is merging of phrases with the same meaning. For the purpose of entity node merging, we do the following:

1. We group the papers according to the conference in which they were published. Then \( \forall \) papers in the same group, we cluster their extracted phrases by running DBSCAN [7] over vector space representations of these phrases. The clusters are created based on lexical similarity which is captured by cosine distance between phrase embeddings. The intuition behind clustering phrases conference wise is that the research papers in a conference have same domain and thus phrases with high lexical similarity belonging to a particular conference are much more likely to mean the same as compared to phrases across conferences. This helps to avoid error as in the example:
‘real time intrusion detection’ in security domain and ‘real time object detection’ in computer vision domain are very different from each other but they may be clustered together by DBSCAN algorithm based on lexical similarity if DBSCAN is run on all the papers in the dataset at once.

2. **Clusters merging across conferences**: A cluster $i$ belonging to conference $c_1$ and a cluster $j$ belonging to conference $c_2$ are merged if they have any common phrase. This is done to capture the fact that there can be more than one conference on the same domain and hence some of their clusters should be merged if they correspond to the same term or phrase. For example, both NAACL and ACL have papers on machine translation and therefore the individual clusters of these conferences corresponding to machine translation should be merged.

Finally we get clusters such that phrases in each cluster have the same meaning. We add only one entity node to the graph for each cluster.

We define the relation type between a paper node and an entity node based on the label of the entity (phrase inside the entity node) for the corresponding paper as identified in the Concept Extraction step.

### 4.3 Neo4j interface

Neo4j [27] is a native graph database, built from the ground up to leverage not only data but also data relationships. Unlike traditional databases, which arrange data in rows, columns and tables, Neo4j has a flexible structure defined by stored relationships between data records. With Neo4j, each data record, or node, stores direct pointers to all the nodes it’s connected to. Because Neo4j is designed around this simple, yet powerful optimization, it performs queries with complex connections orders of magnitude faster, and with more depth, than other databases.

We imported the graph formed in the last step 4.2 in Neo4j. We created two types of nodes - paper nodes and phrase nodes. Paper nodes have the following information - title, authors, year and unique identifiers for each paper node. Phrase nodes contain the representative phrase from each cluster formed in 4.2. Phrase nodes have attributes - unique identifier for each cluster, representative name, list of all the phrases in the cluster. We define four types of relationships:

- **has_aim**: directed edge from paper node to phrase node which represents that the corresponding phrase is aim concept in the corresponding paper. Stores unique identifiers for phrase node and paper node it connects.

- **has_result**: directed edge from paper node to phrase node which represents that the corresponding phrase is result concept in the corresponding paper. Stores unique identifiers for phrase node and paper node it connects.

- **has_result**: directed edge from paper node to phrase node which represents that the corresponding phrase is result concept in the corresponding paper. Stores unique identifiers for phrase node and paper node it connects.
• Cited by: directed edge from paper node 1 to paper node 2 which represents that paper 2 was cited by paper 1.

Figure 4.1 Sample from our constructed graph. Green nodes correspond to research papers and brown nodes correspond to extracted phrase entities.

4.4 Results and Discussion

Total number of unique phrases produced by the proposed algorithm are 565,031. Using DBSCAN we form 63,638 clusters having 266,015 phrases. Our final graph contains 332,242 paper nodes, 362,654 entity nodes, 483,899 aim relations, 982396 relations and 661 result relations.

We can see that result relations are quite few as compared to method and aim relations. This is mainly because of less number of seed features for Result due to less annotated data as compared to Aim and Method.

The constructed graph can summarize the research community in the following way:

1. All the papers on a particular topic can be accessed by just finding the entity node corresponding to the topic in the graph. The associated papers can also be differentiated on the basis of whether
the topic appears as aim or method or result in the paper. This can also help in academic search and recommendation.

2. A field can be summarized by finding all the methods used in the field and applications of the field by finding all the aims where the field has been used as a method.

3. Trend Analysis, conference proceedings summarization, or summarization of a particular author’s work can be done using the meta data in the paper node.
Chapter 5

Temporal Analysis

5.1 Objectives

One of the first elementary steps we take as researchers is to determine which problems to focus on solving, and structured analysis on present research status will help researchers identify critical problems and also give insight about how they developed across time. Due to this it will be easier to realize if a particular problem has got no recent improvement in the recent past and has moved into a thriving application and so on. In this section, we take the preliminary steps towards temporal analysis of scientific literature. Using the knowledge graph generated using the techniques in Chapter 3 and 4, we aim to find saturated fields and grand challenges. We define saturated fields as those research problems which have been studied to a great extent and nothing much has left to achieve in them. On the other hand grand challenges are defined as those problems which have been tried to solve over a large period of time and are still worked upon extensively.

5.2 Definitions

**Saturated Problems:** Problems which were very actively studied in the yesteryears and are now solved to a great extent. Example, parts of speech tagging in NLP.

**Grand Challenges:** Problems which were defined in yesteryears and are still worked upon extensively. Example, machine translation in NLP. Research during the 1980s typically relied on translation through some variety of intermediary linguistic representation involving morphological, syntactic, and semantic analysis. In current times, research has focused on moving from domain specific systems to domain independent translation systems.
5.3 Approach

From the first step, we have research problems which have been studied as AIM for the last fifty years. We also have techniques or METHOD used to solve these problems over these years. We first extract data for each research field, \( p \), and find the number of times paper published on them for each of the years in the range 1971 to 2013.

5.3.1 Finding Saturated Problems

- Count vs year plot for such problems should show a steep decline in the current years.
- Based on exploratory data analysis we came up with the following rules for finding saturated problems from the data collected above
- We list a problem \( p \) as a saturated problem if:
  - \( T_1 \) is the first year when the problem appeared in the literature. \( T_2 \) is the last time when the problem appeared in the literature.
  - Count of \( p \) appearing as aim in \( T_2 \) should be less than the count of \( p \) appearing as aim in \( T_1 \)
  - Peak of count vs year plot should have occurred much before 2013.
  - Suppose problem \( p_1 \) has peak at time \( t_1 \) and problem \( p_2 \) has peak at time \( t_2 \). \( P_1 \) is a better candidate for saturated problem than \( p_2 \) if the difference between \( T_2 \) of \( p_1 \) and \( t_1 \) is more than the difference between \( T_2 \) of \( p_2 \) and \( t_2 \).

5.3.2 Finding Grand Problems

- Count vs year plot for such problems should start from yester years and be consistent over the time. Peaks should be current years as well as yesteryears.
- Based on exploratory data analysis we came up with the following rules for finding grand challenges from the data collected above
  - We list a problem \( p \) as a grand challenge if:
    * \( T_1 \) is the first year when the problem appeared in the literature. \( T_2 \) is the last time when the problem appeared in the literature.
    * \( T_1 \) for problem \( p \) to be classified as a grand challenge should be before 2000 and \( T_2 \) after 2010. Time span between \( T_1 \) and \( T_2 \) should be more than 10 years.
    * Count of \( p \) appearing as an aim in \( T_2 \) should be more than some threshold. This is to rule out the edge cases where there is occurrence of few counts in current years.
  - We rank these problems based on the following formula:
* To capture the fact that more the span of the problem over the years, more likely it is a grand challenge; we propose rank to be directly proportional to the number of years it spans to.

* To capture the fact the count needs to be consistent over the years; we propose rank to be inversely proportional to \( \sum_{i=1}^{n} (\text{count}[i] - \text{count}[i-1]) \) where \( i \) iterates over all the years in which a problem \( p \) occurs.

\[
Rank(p) \propto \frac{n}{\sum_{i=1}^{n} (\text{count}[i] - \text{count}[i-1])}
\]  

(5.1)

Where \( i \) iterates over all the years in which problem \( p \) occurs, starting from the second entry and \( n \) is the total number of years.

5.4 Results and Discussion

- All experiments were done on the final graph generated from section 4. These papers range from 1936 to 2013. However, for the period 1936 - 1971, the number of papers available were relatively very less for temporal analysis. So we pruned the data further and worked on papers from 1971 to 2013.

- We got a total of 555,383 problems in the first step. Out of these, our algorithm classified 599 as saturated problems and 1052 as grand challenges. To analyse the results, we extracted top 100 problems in both the categories. We represent our results as word clouds where the font and color of each word is proportional to rank of that problem as extracted by our algorithm.

| Grand Challenges                        | Saturated Problems          |
|-----------------------------------------|-----------------------------|
| speech recognition                      | disk arrays                 |
| computer vision                         | schema integration          |
| kolmogorov complexity                   | abductive reasoning         |
| real-time applications                  | reconfigurable mesh         |
| human-computer interaction              | loop transformations        |
| query language                          | non-monotonic reasoning     |
| automatic parallelization               | claw-free graphs            |
| stereo vision                           | facility location problem   |
| java                                    | one-way function            |
| xml                                     | robot learning              |

Table 5.1 Top 10 Grand Challenges and Saturated Problems.
Discussion on Results:

1. Speech recognition has a rich history that precedes Internet era. In 1952, three Bell lab researchers made “Audrey” which recognized formats in power spectrum of each word. Investment in research in this area amplified during 1970s with DARPA marking funding for understanding speech. IEEE speech groups were setup. In 1990s CMU led research funded Sphinx system which dominated DARPA 1992 evaluation. In 2005 Siri came into life under Apple. From 2012 there was major breakthrough in research and HMM models which were industry standard till then were replaced by DNN. In 2014 end-to-end speech training was new paradigm that caught winds within DNN. In 2016 CMU and Google collectively introduced idea of “Attention” in training. In past three years there has been work on language agnostic ASR and more notable improvements kept on pressing. With importance of digital assistance, industry support has further expedited constant improvements every month over month till date. Clearly its a field with surreal active development and its not a surprise that our Model has correctly predicted this model as a Grand Challenge.

2. Human Computer interaction is defined as a discipline concerned with the design and evolution of interactive computing systems for human use. HCI surfaced in the 1980s with the advent of personal computing, just as machines such as the Apple Macintosh, IBM PC 5150 started turning up in homes and offices. HCI soon became the subject of intense academic investigation. Initially, HCI researchers focused on how easy computers are to learn and use which has now also included to support the vision of personalized, adaptive, responsive, and proactive services, adaptation and personalization methods and techniques that will need to consider how to incorporate AI and big data [24].

3. In algorithmic information theory, the Kolmogorov complexity of an object, such as a piece of text, is the length of a shortest computer program that produces the object as output. Research on this started in 1970s and is still going on.

4. The exact solution of facility location problem is known to be hard. And there are many approximation algorithms. No new research have been done on this problem. So clearly it is a saturated problem.

5. A one-way function is easy to compute on every input, but hard to invert. Although, The existence of true one-way functions is an open conjecture. In practice many functions such as those based on discrete Log are assumed to be work well since no polynomial time algorithm is known to invert them. Also, their existence would prove that the complexity classes P and NP are not equal.

6. Loop optimization is the process of increasing execution speed and reducing overhead of loops. This problem is fairly solved and many modern compilers already use loop optimization techniques like Fission, Fusion, Inversion, Parallelisation etc.
Figure 5.1 Word Cloud for Grand Challenges
Figure 5.2 Word Cloud for Saturated Problems
Chapter 6

Related Work

6.1 Concept Extraction

There has been growing interest in studying automatic methods of information extraction from scientific articles. Our work maps to mainly two types of problems - Extracting key phrases, concepts, and relations between them and extracting structured information in the form of a knowledge graph from the scientific literature. Keyphrase extraction, specifically from scientific articles, started with SemEval 2010 Task 5 [13], which was focused on automatic keyphrase extraction from scientific articles and prepared a dataset of 284 articles marked with key phrases. [9] Gollapalli and Caragea studied the keyphrase extraction problem in an unsupervised setting. They extracted candidates from the title, abstracts, and citation contexts and used Page Rank [18] to give a score to the candidates. [11] Gupta and Manning first proposed a task that defines scientific terms for 474 abstracts from the ACL anthology [22] into three aspects: domain, technique, and focus. They applied template-based bootstrapping on the title and abstract of articles to tackle the problem. They used handcrafted dependency-based features. Based on this study, [25] improved the performance by introducing hand-designed features to the bootstrapping framework. Our system beats their systems in terms of recall for both aim and method concepts. Also, we worked on a larger multi-domain dataset. SemEval 2017 Task 10 [3] focused on mention level keyphrase identification and their classification into three categories - TASK, PROCESS, and MATERIAL. They prepared an annotated dataset consisting of 500 papers from Material Science and Computer Science journals. Many systems [2] [26] solved the problem in a supervised manner. Top system [2] modeled the problem as a sequence labeling problem. [26] Tsujimura trained LSTM-ER on that dataset. However, these supervised systems require a large amount of training data, in the absence of which they tend to overfit. Our semi-supervised method can work using a small set of annotated documents for original features.

In work by M. Liakata [14], a discourse-driven content model has been used to summarise scientific articles. In their approach, full papers are first automatically annotated using CoreSC scheme [15] [16], capturing 11 content-based concepts such as Hypothesis, Result, Conclusion etc. A content model is then used to provide the skeleton of the summary, making a distinction between dependent and indepen-
dent categories. Another work by E. Collins[5] introduces a new dataset for summarization of scientific publications and later introduces a method HighlightROUGE to automatically extend this dataset. Later, they also introduce AbstractROUGE, which can be used to extract summaries by exploiting the abstract of a paper. In a paper by F. Peng and A. McCallum [19], they try to improve the accuracy for the task of Information Extraction from Research Papers by employing Conditional Random Fields (CRFs). On a standard benchmark data set, they achieved new state-of-the-art performance, reducing error in average F1 by 36%, and word error rate by 78% compared with the previous best SVM results. A Cohan and N. Goharian [4] propose a summarization approach for scientific articles that takes advantage of citation-context and the document discourse model. Their method overcomes the inconsistency between the citation summary and the article’s content by providing context for each citation.

6.2 Literature Graph

There is also ongoing work on constructing knowledge graphs from the scientific literature. [23] Sinha builds a complex graph consisting of six types of entities: field of study, author, institution (the affiliation of the author), paper, venue (journal and conference series) and event. [1] Ammar focused on constructing literature graph consisting of papers, authors, entities nodes, and various interactions between them (e.g., authorship, citations, entity mentions). [17] Luan developed a unified framework for identifying entities, relations, and coreference clusters in scientific articles with shared span representations. They used supervised methods by creating a dataset that included annotations for scientific entities, their relations, and coreference clusters for 500 scientific abstracts from AI conferences proceedings.

6.3 Temporal Analysis of Scientific Literature

[10] first proposed a task that defines scientific terms for 474 abstracts from the ACL anthology [22] into three aspects: domain, technique, and focus. They applied template-based bootstrapping on the title and abstract of articles to tackle the problem. They used handcrafted dependency-based features. Based on this study, [25] improved the performance by introducing hand-designed features to the bootstrapping framework. They both tried to study the influence of different scientific communities over some time. However, their work was limited to the computational linguistics field.

[21] studied temporal analysis on ACL anthology network. They particularly did a temporal analysis of citations by observing the change in citation purpose and polarity over time. [12] provides an approach to understand the changing roles of a publication characterized by its citation contexts in the full text of publications. They proposed approaches for representing the changing citation contexts of cited publications in different periods as sequences of vectors by training temporal embedding models. They evaluated their work in the biomedical domain.
Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we propose semi-supervised techniques for identifying Aim, Method and Result concepts from scientific articles. We show how these concepts can be introduced in the citation graph to graphically summarize the research community and the various applications of the graphical representation thus formed. We show the domain-independence of our approach as Seed features from one domain (physics, material science from SemEval dataset) were used to extract concepts for another domain (computer science papers from DBLP dataset).

We also experimentally show the scalability of our approach and compared the results with the state-of-the-art. We went on to do the temporal analysis on top of the knowledge graph formed by finding grand challenges and saturated problems in the computer science domain.

7.2 Future Work

Scope of improvement:

- Better heuristics to select the best candidates generated by the bootstrapping method.
- Better selection of candidate phrases; not just noun phrases.
- Better node merging process.

Some of the future directions and research problems which emerge out of this thesis are as follows:

Finding Transition Time: We plan to further do temporal analysis by finding transition time for problems where transition time is defined as the period where a problem starts occurring as a method instead of aim. We also propose to do temporal analysis for citation purposes.

Scientific literature summarization: Knowledge Graph representation formed and imported in Neo4j can be directly used to summarize research fields automatically.
Influence of communities on each other: Due to different subfields of computer science domain present in our dataset, we can study the dependence between different communities.

Popularity of communities: Popularity of different communities can be studied with time.

Other tasks: Final Graph representation formed can be used to study existing interesting problems like citation link prediction and personalized scientific recommendation system.
Chapter 8

Related Papers

1. Kritika Agrawal, Aakash Mittal, and Vikram Pudi. Scalable, semi-supervised extraction of structured information from scientific literature. In Proceedings of the Workshop on Extracting Structured Knowledge from Scientific Publications, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

2. Kritika Agrawal and Aakash Mittal. Iiit-h @ clscisumm-18. In BIRNDL@SIGIR, 2018.

3. Kritika Agrawal and Vikram Pudi. Temporal Analysis of Scientific Literature to find Saturated Problems and Grand Challenges. In BIRDS@SIGIR 2020.
Bibliography

[1] 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.

[2] W. Ammar, M. Peters, C. Bhagavatula, and R. Power. The ai2 system at semeval-2017 task 10 (science): semi-supervised end-to-end entity and relation extraction. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 592–596, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[3] 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 (SemEval-2017), pages 546–555. Association for Computational Linguistics, 2017.

[4] A. Cohan and N. Goharian. Scientific article summarization using citation-context and article’s discourse structure. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 390–400, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics.

[5] E. Collins, I. Augenstein, and S. Riedel. A supervised approach to extractive summarisation of scientific papers. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pages 195–205, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics.

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

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD’96, pages 226–231. AAAI Press, 1996.

[8] S. Ganguly and V. Pudi. Competing algorithm detection from research papers. In Proceedings of the 3rd IKDD Conference on Data Science, 2016, CODS ’16, pages 23:1–23:2, New York, NY, USA, 2016. ACM.
[9] S. D. Gollapalli and C. Caragea. Extracting keyphrases from research papers using citation networks. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI’14, pages 1629–1635. AAAI Press, 2014.

[10] S. Gupta and C. Manning. Analyzing the dynamics of research by extracting key aspects of scientific papers. In Proceedings of 5th International Joint Conference on Natural Language Processing, pages 1–9. Asian Federation of Natural Language Processing, 2011.

[11] S. Gupta and C. D. Manning. Improved pattern learning for bootstrapped entity extraction. In CoNLL, 2014.

[12] J. He and C. Chen. Temporal representations of citations for understanding the changing roles of scientific publications. Frontiers in Research Metrics and Analytics, 3:27, 2018.

[13] S. N. Kim, O. Medelyan, M.-Y. Kan, and T. Baldwin. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval ’10, pages 21–26, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[14] M. Liakata, S. Dobnik, S. Saha, C. Batchelor, and D. Rebholz-Schuhmann. A discourse-driven content model for summarising scientific articles evaluated in a complex question answering task. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 747–757, Seattle, Washington, USA, Oct. 2013. Association for Computational Linguistics.

[15] M. Liakata, S. Teufel, A. Siddharthan, and C. Batchelor. Corpora for the conceptualisation and zoning of scientific papers. In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10), Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[16] M. Liakata, P. Thompson, A. de Waard, R. Nawaz, H. Pander Maat, and S. Ananiadou. A three-way perspective on scientific discourse annotation for knowledge extraction. In Proceedings of the Workshop on Detecting Structure in Scholarly Discourse, pages 37–46, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[17] Y. Luan, L. He, M. Ostendorf, and H. Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. CoRR, abs/1808.09602, 2018.

[18] L. PAGE. The pagerank citation ranking : Bringing order to the web. http://www-db.stanford.edu/backrub/pageranks.ps, 1998.

[19] F. Peng and A. McCallum. Accurate information extraction from research papers using conditional random fields. In HLT-NAACL, 2004.

[20] A. Prasad, M. Kaur, and M.-Y. Kan. Neural parscit: a deep learning-based reference string parser. International Journal on Digital Libraries, 19(4):323–337, Nov 2018.

[21] D. Radev and A. Abu-Jbara. Rediscovering ACL discoveries through the lens of ACL anthology network citing sentences. In Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries, pages 1–12, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[22] D. Radev, P. Muthukrishnan, V. Qazvinian, and A. Abu-Jbara. The acl anthology network corpus. Language Resources and Evaluation, pages 1–26, 2013.
[23] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-J. P. Hsu, and K. Wang. An overview of microsoft academic service (mas) and applications. In Proceedings of the 24th International Conference on World Wide Web, WWW ’15 Companion, pages 243–246, New York, NY, USA, 2015. ACM.

[24] C. C. Stephanidis, G. Salvendy, M. of the Group Margherita Antona, J. Y. C. Chen, J. Dong, V. G. Duffy, X. Fang, C. Fidopiastis, G. Fragomeni, L. P. Fu, Y. Guo, D. Harris, A. Ioannou, K. ah (Kate) Jeong, S. Konomi, H. Krömker, M. Kurosu, J. R. Lewis, A. Marcus, G. Meiselwitz, A. Moallem, H. Mori, F. F.-H. Nah, S. Ntoa, P.-L. P. Rau, D. Schmorrow, K. Siau, N. Streitz, W. Wang, S. Yamamoto, P. Zaphiris, and J. Zhou. Seven hci grand challenges. International Journal of Human–Computer Interaction, 35(14):1229–1269, 2019.

[25] C.-T. Tsai, G. Kundu, and D. Roth. Concept-based analysis of scientific literature. In Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, CIKM ’13, pages 1733–1738, New York, NY, USA, 2013. ACM.

[26] T. Tsujimura, M. Miwa, and Y. Sasaki. Tü-coin at semeval-2017 task 10: Investigating embeddings for end-to-end relation extraction from scientific papers. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 985–989, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[27] J. Webber and I. Robinson. A Programmatic Introduction to Neo4J. Addison-Wesley Professional, 1st edition, 2018.

[28] T. Zhou. Academic Papers (Researches on Big Scholarly Dataset), 2016. http://zhou142.myweb.cs.uwindsor.ca/academicpaper.html