
# Automatic Generation of Descriptive Related Work Reports

**AHMED GHASSAN TAWFIQ ABURA'ED**

TESI DOCTORAL UPF / 2020

Thesis Director:

Prof. HORACIO SAGGION
LaSTUS lab - TALN Group
Dept. of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona, Spain

Dissertation submitted to the Department of Information and Communication Technologies of Universitat Pompeu Fabra in partial fulfillment of the requirements for the degree of

DOCTOR PER LA UNIVERSITAT POMPEU FABRA




TALN (http://taln.upf.edu), Department of Information and Communication Technologies (http://www.upf.edu/dtic), Universitat Pompeu Fabra (http://www.upf.edu), Barcelona, Spain.

The doctoral defense was held on ..... at the Universitat Pompeu Fabra and scored as .....

**Prof. Horacio Saggion**
(Thesis Supervisor)
Universitat Pompeu Fabra (UPF), Barcelona, Spain

**Dr. Leila Kosseim**
(Thesis Committee Member)
Concordia University, Montreal, Canada

**Dr. Elena Lloret**
(Thesis Committee Member)
Universidad de Alicante, Alicante, Spain

**Dr. Iria da Cunha**
(Thesis Committee Member)
UNED, Madrid, Spain

*To my Parents: Ghassan and Sounia, my Brother: Hamza and my Sister: Dalia.*

This thesis has been carried out at the Large Scale Text Understanding Systems Lab (LaSTUS) at the Natural Language Processing Research Group (TALN) of Universitat Pompeu Fabra in Barcelona, Spain, from Oct. 2015 to Sep. 2020. It is supervised by Prof.Horacio Saggion.

# Acknowledgments

I believe that a PhD is not just a journey that you go through for a few years and then you are done, it is a life changing event that sticks with you forever modifying your core, I am for sure a different person. Beyond learning the scientific methods, learning what questions to ask, how to tackle problems, how to find answers and even how to think, I always faced new situations which required constant adaptation and grasping new sets of skills. I would like to express my deepest appreciation to the people that I encountered during my PhD, they have shown audacity, kindness, and wiseness and I am deeply thankful to them for their guidance, support, and friendship.

I'm deeply indebted to my supervisor: Prof. Horacio Saggion, I could not have done it without him. He offered me this PhD and he mentored me throughout the entire process. I am forever grateful for what he taught me, just to name a few: multi-tasking, time management, keeping it cool during hard times, keeping a warm heart, communication, investigation and the list goes on. He is a true leader who taught me how to have my own thread of thoughts and he changed the way I look at things, for sure there was some babysitting involved, and I am really grateful for his patient with me.

I cannot begin to express my thanks to Lydia Garcia who was very helpful thought all the paper work and I appreciate her emotional and motivational support. I also would like to thank the entire administration staff, I am an international student and it took a tremendous amount of bureaucracy and paper works to keep me at UPF, to mention a few: Jana, Luis, Vanessa, Ruth, Montse.

I also would like to thank my research group: TALN led by Prof. Leo Wanner, I appreciate the insightful conversations. I closely collaborated with Aleksandr Shvets, Alex Bravo and Francesco Ronzano. I have also collaborated with Montserrat Marimon, Alicia Burga and Beatriz Fisas Elizalde for annotation tasks and I am very grateful for that. Thanks to my lunch and tea gang (DA inner-circle) Alexandre Peiro Lilja, Kim Cheng Sheang, Giorgia Cistola, Paula Fortuna and Janine Kleinhans. Thanks for Roberto Carlini for all the brain stormign sessions and advice. Thanks for Laura Pérez Mayos and Carla Ten for being an awesome office mates. Than ks to Seda Mut, Pablo Accuosto, Guillermo Cambara Ruiz for all the conversations and tech advice. Thanks for Joan Codina and Jens Grivolla for the technical support. Thanks for Monica Dominguez for organizing the internal seminars. Thanks

for Mireia Farrús, Simone Mille and Sándor Dés Alcalá for the nice conversations and the rest of the gang as well: Juan Soler Company, Gerard Casamayor Del Bosque, Alexandra-catalina Matreata, Francesco Barbieri, Miguel Ballesteros, Luis Espinosa Anke, Hamdi Alp Öktem, Sara Rodriguez-Fernandez and Joan Pere Sánchez Pellicer. I want to also thank my co-authors Luis Chiruzzo, Ahmet Fadhil and Daniel Ferrés.

Huge thanks for my friends from the NLP community: Venelin, Mahmoud, Marina, Tatiana, Dasha, Amir, Khalil, Sanja, Irina, Jelke, Stefania, Syrielle and Taha. My friends in MTG: Swapnil, Marios, Sankalp and Sertan. Huge thanks to my flatmates over the years: Melis, Mehdi, Giada, Svetlana, Gulce, Alicia, Vaida and Elena.

As for my research journey, I would like to thank the University of Sheffield (Sheffield, UK), for allowing me to be part of their SoBigData Project and hosting me for around a month. I also would like to thank the Concordia University (Montreal, Canada) for allowing me to do a research visit for around six months.

I am also deeply thank full for the Able to Include project and the people behind it, I was honored to be part of it. Moreover, I would like to thank Aurelio Ruiz Garcia and the María de Maeztu program for all the financial support that allowed me to continue my research.

Finally, I would like to thank all my family and friends. You know you are guys.

# Abstract

A related work report is a section in a research paper which integrates key information from a list of related scientific papers providing context to the work being presented. Related work reports can either be **descriptive** or **integrative**. Integrative related work reports provide a high-level overview and critique of the scientific papers by comparing them with each other, providing fewer details of individual studies. Descriptive related work reports, instead, provide more in-depth information about each mentioned study providing information such as methods and results of the cited works. In order to write a related work report, scientist have to identify, condense/summarize, and combine relevant information from different scientific papers. However, such task is complicated due to the available volume of scientific papers. In this context, the automatic generation of related work reports appears to be an important problem to tackle. The automatic generation of related work reports can be considered as an instance of the multi-document summarization problem where, given a list of scientific papers, the main objective is to automatically summarize those scientific papers and generate related work reports. In order to study the problem of related work generation, we have developed a manually annotated, machine readable data-set of related work sections, cited papers (e.g. references) and sentences, together with an additional layer of papers citing the references. We have also investigated the relation between a citation context in a citing paper and the scientific paper it is citing so as to properly model cross-document relations and inform our summarization approach. Moreover, we have also investigated the identification of explicit and implicit citations to a given scientific paper which is an important task in several scientific text mining activities such as citation purpose identification, scientific opinion mining, and scientific summarization. We present both extractive and abstractive methods to summarize a list of scientific papers by utilizing their citation network. The extractive approach follows three stages: scoring the sentences of the scientific papers based on their citation network, selecting sentences from each scientific paper to be mentioned in the related work report, and generating an organized related work report by grouping the sentences of the scientific papers that belong to the same topic together. On the other hand, the abstractive approach attempts to generate citation sentences to be included in a related work report, taking advantage of current sequence-to-sequence neural architectures and resources that we have created specifically

for this task. The thesis also presents and discusses automatic and manual evaluation of the generated related work reports showing the viability of the proposed approaches.

# Resum

Una secció d'antecedents o estat de l'art d'un articulo científic resumeix la informació clau d'una llista de documents científics relacionats amb el treball que es presenta. Per a redactar aquesta secció de l'article científic l'autor ha d'identificar, condensar / resumir i combinar informació rellevant de diferents articles. Aquesta activitat és complicada per causa del gran volum disponible d'articles científics. En aquest context, la generació automàtica d'aquestes seccions és un problema important a abordar. La generació automàtica d'antecedents o d'estat de l'art pot considerar-se com una instància del problema de resum de documents. Per estudiar aquest problema, es va crear un corpus de seccions d'estat de l'art d'articles científics manualment anotat i processat automàticament. Així mateix, es va investigar la relació entre citacions i l'article científic que es cita per modelar adequadament les relacions entre documents i, així, informar el nostre mètode de resum automàtic. A més, es va investigar la identificació de citacions implícites a un article científic que és un problema important en diverses activitats de mineria de textos científics. Presentem mètodes extractius i abstractius per resumir una llista d'articles científics utilitzant el conjunt de citacions de cada article. L'enfoc extractiu segueix tres etapes: càlcul de la rellevància de les oracions de cada article en funció de les seves citacions, selecció d'oracions de cada article científic per a integrar-les en el resum i generació de la secció de treballs relacionats agrupant les oracions per tema. Per un altre costat, l'enfoc abstractiu implementa la generació de citacions per a incloure-les en un resum que utilitza xarxes neuronals i recursos que hem creat específicament per a aquest tasca. La tesi també presenta i discuteix l'avaluació automàtica i el manual dels resums generats automàticament, demostrant la viabilitat dels mètodes proposats.

(*Translated from English by Professor Horacio Saggion*)

# Resumen

La sección de trabajos relacionados de un artículo científico resume e integra información clave de una lista de documentos científicos relacionados con el trabajo que se presenta. Para redactar esta sección del artículo científico el autor debe identificar, condensar/resumir y combinar información relevante de diferentes artículos. Esta tarea es complicada debido al gran volumen disponible de artículos científicos. En este contexto, la generación automática de tales secciones es un problema importante a abordar. La generación automática de secciones de trabajo relacionados puede ser considerada como una instancia del problema de resumen de documentos múltiples donde, dada una lista de documentos científicos, el objetivo es resumir automáticamente esos documentos científicos y generar la sección de trabajos relacionados. Para estudiar este problema, hemos creado un corpus de secciones de trabajos relacionados anotado manualmente y procesado automáticamente. Asimismo, hemos investigado la relación entre las citaciones y el artículo científico que se cita para modelar adecuadamente las relaciones entre documentos y, así, informar nuestro método de resumen automático. Además, hemos investigado la identificación de citaciones implícitas a un artículo científico dado que es una tarea importante en varias actividades de minería de textos científicos. Presentamos métodos extractivos y abstractivos para resumir una lista de artículos científicos utilizando su red de citaciones. El enfoque extractivo sigue tres etapas: cálculo de la relevancia las oraciones de cada artículo en función de la red de citaciones, selección de oraciones de cada artículo científico para integrarlas en el resumen y generación de la sección de trabajos relacionados agrupando las oraciones por tema. Por otro lado, el enfoque abstractivo intenta generar citaciones para incluirlas en un resumen utilizando redes neuronales y recursos que hemos creado específicamente para esta tarea. La tesis también presenta y discute la evaluación automática y manual de los resúmenes generados automáticamente, demostrando la viabilidad de los enfoques propuestos.

(*Translated from English by Professor Horacio Saggion*)

# Contents

# List of Figures

# List of Tables

# Chapter 1
# Introduction

This research emphasizes the topic of *automatic generation of descriptive related work reports*. We provide the reader with the problem statement, objectives, rationale and research questions that have triggered this research. As well, prominent work in the area is critically reviewed. Later, several methods are proposed to describe the process of *automatic descriptive related work generation* as well as the results of those experiments against a set of baselines.

## 1.1 Context, Motivation and Objectives

Nowadays scientific fields of study are rapidly growing, the number of publications is increasing exponentially making it problematic for scholars to keep up with knowledge in their field, related domains or even new developments in new sectors that are appearing due to such exponential growth Larsen & von Ins (2010). de Solla Price & Page (1961) stated that by 1950 the number of journals in existence sometime between 1650 and 1950 was about 60,000 and with the known growth rate: about 5.6% per year which cause it to double in 15 years, the number would be about 1 million in year 2000. Moreover, current estimates indicate that the number of scientific publications grows at unprecedented rates: between 0.7 and 1.5 million new papers are published every year (Bornmann & Mutz, 2015; Jinha, 2010). Such exponential growth in the number of publications from all disciplines is increasing the overlap between different fields due to the progressively interrelated nature of real-world tasks leading to situations that often require specialists in one domain to rapidly learn about other areas in a short amount of time. This rapid growth makes it more and more challenging to get rapidly familiar with the new advances in a new area. In addition, during the last decade the amount of scientific information available on-line increased at an unprecedented rate with recent estimates reporting a new paper published every 20 seconds (Munroe,

2013). PubMed, the most important reference in biomedicine, includes more than 25M papers with a growth rate of about 1,370 new articles per day. Elsevier' Scopus and Thomson Reuther's ISI Web of Knowledge respectively contain more than 57 and 90 million papers. The Cornell University Library arXiv initiative provides access to over 1M e-prints from various scientific domains. At the same time, well recognized conferences (ACL, LREC, etc.) are making their contents freely available through dedicated archives even before the conference takes place allowing scientist to scrutinize relevant papers before their peers publicly present their findings. This context of the exponential growth of scientific papers alongside the amount of records we already have is causing what scholars refer to as an "information overload" that urges the need to investigate scientific text summarization.

On the other hand, scientific publications are permanent records of what has been discovered so far, they contain most of humanity's knowledge, including information as relevant as how to cure diseases, invent life saving machines and create drugs (Kuhn & Hawkins, 1963). Scientific research is a collective activity. The work of researchers depends on knowledge accumulated by scientists and scholars over years of research. Therefore, an author often needs to provide related previous works for his or her readers to help them understand the context of his or her contributions in an area of research, and also to facilitate any form of comparison between the current and previous works. Similarly identifying background information in scientific articles can help scholars understand major contributions in an area of research more easily. Also, scientists are often called upon to review papers in a wide range of areas, some of which may be unfamiliar to scholars. Thus, they must learn about a new discipline "on the fly" to be able to produce such reviews. Additionally, authors of journal articles and books must write accurate surveys of previous work, ranging from short summaries of related research to in-depth historical notes. It can be seen that all those issues motivate the need for designing a methodology to automatically summarize scientific articles and generates related work reports. Such reports can be later used as a related work section in a scientific paper or a review of a related work in a study field which serve as a review for scholars.

A related work summary is a text summary which describes briefly the main ideas of previous or recent works, indicating their relevant aspects in the context of the current paper's topics. A specific related work example extracted from (Kong et al., 2014) is shown in Figure 1.1.

This related work section introduces previous related works for a paper on Argument Labeling with Joint Inference in Discourse Parsing. From Figure 1.1, we can have a glance at the structure of related work sections. Related

For argument labeling in discourse parsing on the PDTB corpus, the related work can be classified into two categories: **locating parts of arguments** and **labeling full argument spans**.
As a representative on **locating parts of arguments**, ***Wellner and Pustejovsky (2007)*** proposed several machine learning approaches to identify the head words of the two arguments for discourse connectives. Following this work, ***Elwell and Baldridge (2008)*** combined general and connective specific rankers to improve the performance of labeling the head words of the two arguments. ***Prasad et al. (2010)*** proposed a set of heuristics to locate the position of the Arg1 sentences for inters entence cases.
. . . .. . . .. . . .
In comparison, **labeling full argument spans** can provide a complete solution to argument labeling in discourse parsing and has thus attracted increasing attention recently, adopting either a **subtree extraction approach** (***Dinesh et al. (2005)***, ***Lin et al. (2014)***) or a **linear tagging approach** (***Ghosh et al. (2011)***).
As a representative **subtree extraction approach**, ***Dinesh et al. (2005)*** proposed an automatic tree subtraction algorithm to locate argument spans for intra-sentential subordinating connectives.
. . . .. . . .. . . .
Instead, ***Lin et al. (2014)*** proposed a two-step approach. First, anargument position identifier was employed . . . .
. . . .. . . .. . . .
As a representative **linear tagging approach**, ***Ghosh et al. (2011)*** cast argument labeling as a linear tagging task using conditional random fields. ***Ghosh et al. (2012)*** further improved the performance with integration of the n-best results.

**Figure 1.1:** A sample related work section.

work sections usually discuss several different topics, such as “subtree extraction” and “linear tagging” approaches shown in the Figure 1.1. Besides the knowledge of previous works, the author often compares his own work "constituent-based" approach with the previous works. The advantages and disadvantages are generally mentioned. The example in Figure 1.1 also indicates this phenomenon.

Finally, in this research we are going to investigate methods in order to summarize scientific papers and automatically generate related work reports that mention and describe those research papers on given topics.

Several studies in the field of Automatic Text Summarization (ATS) apply

methods to summarize scientific papers. However, only few studies have tackled the task of automatic generation of related work reports by summarizing a list of scientific papers to be mentioned in these reports. These few studies that directly investigate automatic related work generation either limit the scope of the scientific paper's content by taking into consideration only a couple of sections from each scientific paper discarding the rest of its content, or they have limited the scope of the related work reports generation process. In this context, the automatic generation of related work reports appears to be an important problem to tackle, which leaves the door open for scholars to further investigate this topic.

Given a list of scientific papers our main objective is to automatically summarize those scientific papers and generate an organized related work report, by grouping the sentences of the scientific papers that belong to the same topic together. Many steps are needed to generate a related work report including: understanding and analyzing the structure of related work sections, finding relevant documents, identifying important sentences of these documents in relation to the current work worth summarizing by scoring and ranking their sentences, determining the order in which the sentences can be grouped together and presented, and finally generating the final related work report.

## 1.2 Research Context

Related work reports or literature reviews offer already digested information ready to be used by researchers interested in getting a gist of the state of the art. Automatically generating this type of text, that is selecting and combining key information from a set of articles, could greatly help researchers in coping with the problem of scientific information overload. The automatic generation of related work sections can be considered an instance of the multi-document summarization problem.

Summarization is the task of condensing a piece of text to a shorter version that contains the main information from the original. There are two general approaches to summarization: **extractive** and **abstractive**. Extractive methods put together summaries exclusively from sentences taken directly from the source text, while abstractive methods may generate novel words and phrases not featured in the source text – as a human-written abstract usually does. The extractive approach is easier, because copying large amounts of text from the source document ensures baseline levels of correct grammar. On the other hand, sophisticated abilities that are crucial to high-quality summarization, such as paraphrasing, generalization, or the incorporation of real-world know-

ledge, are possible only in an abstractive framework (See et al., 2017). Finally, extractive summarization is a selection problem, while abstractive summarization requires a deeper semantic and discourse understanding of the text, as well as a novel text generation process.

Generic text summarization techniques may not work well in specialized genres such as the scientific genre and domain specific techniques (Saggion & Lapalme, 2002a; Teufel & Moens, 2002). Scientific papers are characterized by several structural, linguistic and semantic peculiarities. Articles include common structural elements (title, authors, abstract, sections, figures, tables, citations, bibliography) that often require specific text processing tools. Additionally, scientific papers have specific discourse structure (Teufel et al., 2009; Liakata et al., 2010). Moreover, scientific papers are not isolated units, but they are inter-connected by means of co-citation relations or citation networks which are useful to quantitatively understand the value of a piece of scientific work. However, citation networks are limited in that they do not provide information about why a paper is being cited or what part of the reference paper the citing paper is referring to. This qualitative information is very important in order to allow fine-grained automatic analysis of scientific works.

Considering the urgent need for new, automated approaches to browse and aggregate scientific information, a number of natural language processing challenges have been proposed in recent years: the Biomedical Summarization (BioSumm) 2014 Task carried out in the context of the Text Analysis Conferences provided a forum for researchers interested in exploring the summarization of clusters of documents where one of the documents is a reference paper and the rest of the documents in the cluster are citing papers which cite the reference paper. Several studies (Qazvinian & Radev, 2008a, 2010a; Abu-Jbara et al., 2013) have proposed to take advantage of the scientific paper's citation network to approach scientific literature summarization. Researchers tend to cite the major contributions of a scientific paper. Therefore, utilizing the citation network between the scientific paper and the papers that are citing it will provide an insight of what those researchers consider an important context in the scientific paper. As an extractive way to summarize scientific papers, researcher try to identify which sentences of a scientific paper have been cited and then they consider these sentences as a summary of the scientific paper. However, identifying which sentences of a reference paper contain the information being referred to by a set of citing papers is a difficult task in part due to the short context provided by the explicit citation, so it becomes necessary to look beyond this explicit citation for other information in the citing paper that might be relevant. The identification of explicit and implicit citations to a given reference paper is important for numerous scientific

text mining activities such as citation purpose identification, scientific opinion mining, and scientific summarization.

Furthermore, in the past few years some works have proposed to cast summarization as a mapping problem between an input sequence and a summary sequence. Recent studies such as (Rush et al., 2017; Nallapati et al., 2016a) have shown that the RNN encoder-decoder performs remarkably well in summarizing short text. Such seq2seq approaches offer a fully data-driven solution to both semantic and discourse understanding and text generation. Though these systems are promising, they exhibit undesirable behavior such as inaccurately reproducing factual details, an inability to deal with out-of-vocabulary (OOV) words, and repeating themselves. While seq2seq offers a promising route for abstractive summarization, using the methodology to other tasks, such as the summarization of a scientific article, is not trivial. Scientific articles are too long to be processed entirely via seq2seq. Moving from one or two sentences, to several sentences or several paragraphs, introduces additional levels of compositionality and richer discourse structure. Deep learning approaches depend heavily on good quality, largescale data sets. Collecting source-summary data pairs is difficult, and data sets are scarce outside of the newswire domain. Therefore, we also investigate to generate related work reports through abstractive summarization using only the title and the abstract of the scientific papers we want to mention in the related work report. To further enhance our neural network and to exploit the fact that we only need a title and an abstract as as source text alongside a target summary, we use many available resources that provide such information to enlarge enormously our data size and train seq2seq models based on such information.

In this context we investigate both extractive and abstractive summarization of scientific papers through utilizing their citation networks to generate related work reports. Extractive summarization of scientific articles has been the focus in the recent past while abstractive summarization remains a challenge to this day due to the length of the scientific papers.

During our research, we faced the following research questions:

- Which information from a given list of related scientific papers should be extracted to produce a related work report?
- How information should be organised in the final related work report?
- How can a related work report be evaluated?

## 1.3 Related Work Reports Analysis

As mentioned at Section 1.1 scientific research is a collective activity. The work of researchers depends on knowledge accumulated by scientists and scholars over years of research. Therefore, an author often needs to describe related previous works for the readers to help them understand the context of his or her contributions in an area of research, also facilitating any form of comparison between the current and previous works.

In this context, every scientific paper should include a related work section providing, in a well organized and condensed form, the key information from a carefully selected list of publications which contextualize and ground the research being presented by an author (Rowley & Slack, 2004). Moreover, their relevance is critical for quality assessment since journals pay particular attention to related work sections where evaluation of manuscripts is of concern (Maggio et al., 2016).

One way of having a brief overview of a research field is by reading related work reports, which usually contain, in condensed form, key information on a topic drawn from different sources. It is a text summary which describes briefly the main ideas of previous or recent works, indicating their relevant aspects in the context of that topic.

Good related work sections are difficult to produce since they require the author to select, contrast, and organize key information from several sources. Khoo et al. (2011); Jaidka et al. (2013) stated that related work sections or literature reviews can either be **descriptive** or **integrative**. Integrative literature reviews provide a high-level overview and critique of the recent work by comparing related papers against each other. Integrative literature reviews focus on the ideas and results extracted from a number of research papers and provide fewer details of individual papers/studies, hence they focus mainly on the Conclusion sections. Finally, they provide critical summaries of topics and methodologies. While descriptive literature reviews provide more in-depth information about each mentioned study, they have a significantly greater number of method, result and interpretation elements embedded within each study. Hence, a descriptive report will summarize individual papers providing information such as methods and results in citation sentences making use of cut-and-paste summarization strategies (Jaidka et al., 2013) which are typical of abstracting a document (insertion, deletion, substitution, etc.) (Endres-Niggemeyer et al., 1995; Saggion, 2011). Finally, all literature reviews encompass integrative and descriptive elements in different proportions.

In our work we are concerned with the automatic production of descriptive related work sections from a set of selected Reference Papers. Summarizing

each Reference Paper with more attention to details will provide a good insight about its contributions. We do not attempt to generate integrative reviews since they will require knowledge difficult to encode in an automatic process. Moreover, recommending a pre-selection set of scientific papers to be included in the report is outside the scope of our work. To further investigate possible ways of compiling a list of scientific papers to cite, see (McNee et al., 2002).

## 1.4 Contributions

The thesis makes several contributions including the creation of two data sets for the study of scientific text summarization, one of them annotated by humans, a state of the art system for citation context identification, a state of the art extractive summarizer for scientific papers, and the first sequence to sequence abstractive model to produce citation sentences.

During our research we made several contributions directly related to automatic generation of related work reports and scientific papers' summarization. We have created several resources and developed many tools that we share with the research community in the hope that such resources could be useful for scholars working on the same topic.

In order to allow the study of this specific problem, we have developed a manually annotated, machine readable data set of related work sections[1], cited papers (e.g. references) and sentences, together with an additional layer of papers citing the references. We additionally present experiments on the identification of cited sentences, using as input citation contexts. The corpus alongside the gold standard are made available for use by the scientific community.

We have also presented experiments on the identification of implicit citations in scientific papers[2] by relying on an annotated data set of explicit and implicit citation sentences, we cast the problem as classification, evaluating several machine learning algorithms trained on a set of task-motivated features.

Moreover on summarization of scientific papers we presented several systems[3,4,5] (over the span of three years) developed to participate in the Computational Linguistics Scientific Document Summarization Shared challenge which addresses the problem of summarizing a scientific paper taking advantage of its citation network (i.e., the papers that cite the given paper). Our

[1] http://taln.upf.edu/sciencecorpus/
[2] https://github.com/AhmedAbuRaed/CitationContextExtension
[3] https://github.com/AhmedAbuRaed/SciSumm2016Testing
[4] https://github.com/AhmedAbuRaed/CL-SciSumm2017
[5] https://github.com/AhmedAbuRaed/CLSciSumm2018

systems are based on both supervised (Convolutional Neural Networks) and unsupervised techniques taking advantage of word embeddings representations and features computed from the linguistic and semantic analysis of the documents.

We also develop a state of the art system that automatically generates descriptive related work reports by performing extractive summarization of scientific papers. The system first scores the sentences of the scientific papers to be mentioned in the related work report, then selects a specific number of sentences with the highest scores. Finally, it applies topic modeling to organize the selected sentences in a comprehensible way that respect the flow of topics and sentences.

We have also automatically generated related work sections using abstractive summarization of scientific papers[6] through a neural sequence learning process which produces citation sentences to be included in a related work section of an article. We train the neural architecture using a novel scientific data set of citation sentences that we have created.

Finally, we have also collaborated with other scholars on related topics to NLP, like producing an auto-trainable morphological generation system[7], the development of a system for Complex Word Identification[8], scientific text sentiment analysis on social media and finally, introducing an Arabic conversational agent that assists physicians and supports patients with the care process.

The contributions can be listed as follows:

- A manually annotated multi-level scientific corpus (3 annotators) that is automatically processed;
- A state of the art system for citation context identification;
- Identifying which sentences in a Reference Paper has been cited by a citation context;
- Multiple supervised and unsupervised methods have been implemented to participate in the CL-SciSumm shared task;
- The design and evaluation of a state of the art related work reports generation system using extractive summarization of scientific papers;
- The design and evaluation of a state of the art related work reports generation system using abstractive summarization of scientific papers;
- A new data set of over 15K pairs of articles and citation sentences to train sequence-to-sequence models;
- The data, the software and instructions on how to reproduce our work

[6] https://github.com/AhmedAbuRaed/SPSeq2Seq
[7] https://github.com/AhmedAbuRaed/SpanishMorphologicalRealizer
[8] https://github.com/AhmedAbuRaed/CWISharedTask2018

are available for the community [9,10,11,12,13,14,15];

### 1.4.1 Publications

During our work on the thesis we have published several scientific papers in the field as follows:

Thesis Related Publications

- AbuRa'ed, Ahmed, Horacio Saggion, and Alexander Shvets, Àlex Bravo. "Automatic Related Work Section Generation: Experiments in Scientific Document Abstracting."
  Accepted at Scientometrics Journal (Q1). 2020.
- AbuRa'ed, Ahmed, Horacio Saggion, and Luis Chiruzzo.
  "A Multi-level Annotated Corpus of Scientific Papers for Scientific Document Summarization and Cross-document Relation Discovery."
  In Proceedings of The $12_{th}$ Language Resources and Evaluation Conference. 2020.
- AbuRa'ed, Ahmed, Luis Chiruzzo, and Horacio Saggion.
  "Experiments in Detection of Implicit Citations."
  WOSP 2018, $7_{th}$ International Workshop on Mining Scientific Publications. 2018.
- AbuRa'ed, Ahmed, Àlex Bravo, Luis Chiruzzo, and Horacio Saggion.
  "LaSTUS/TALN+ INCO@ Cl-Scisumm 2018-Using Regression and Convolutions for Cross-document Semantic Linking and Summarization of Scholarly Literature."
  BIRNDL 2018, $3_{rd}$ Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries. 2018.
- AbuRa'ed, Ahmed, Luis Chiruzzo, and Horacio Saggion.
  "What Sentence are you Referring to and Why? Identifying Cited Sentences in Scientific Literature."
  RANLP 2017, International Conference Recent Advances in Natural Language Processing. 2017.
- Abura'ed, Ahmed, Luis Chiruzzo, Horacio Saggion, Pablo Accuosto, and Àlex Bravo.

[9] http://taln.upf.edu/sciencecorpus
[10] https://github.com/AhmedAbuRaed/CitationContextExtension
[11] https://github.com/AhmedAbuRaed/SciSumm2016Testing
[12] https://github.com/AhmedAbuRaed/CL-SciSumm2017
[13] https://github.com/AhmedAbuRaed/CLSciSumm2018
[14] https://github.com/AhmedAbuRaed/RWRG
[15] https://github.com/AhmedAbuRaed/SPSeq2Seq

"LaSTUS/TALN @ CLSciSumm-17: Cross-Document Sentence Matching and Scientific Text Summarization Systems."
In Proceedings of the Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017). 2017.

- Saggion, Horacio, Ahmed AbuRa'ed, and Francesco Ronzano.
"Trainable Citation-enhanced Summarization of Scientific Articles."
In Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016). 2016.

Other Publications

- Fadhil, Ahmed, and Ahmed AbuRa'ed.
"OlloBot-Towards A Text-Based Arabic Health Conversational Agent: Evaluation and Results."
In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019). 2019.
- Chiruzzo, Luis, Ahmed AbuRa'ed, Àlex Bravo, and Horacio Saggion.
"LaSTUS-TALN+ INCO@ CL-SciSumm 2019."
In BIRNDL@ SIGIR. 2019.
- AbuRa'ed, Ahmed, and Horacio Saggion.
"LaSTUS/TALN at Complex Word Identification (CWI) 2018 Shared Task."
In Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, ACL. 2018.
- Ferrés, Daniel, Ahmed AbuRa'ed, and Horacio Saggion.
"Spanish Morphological Generation with Wide-Coverage Lexicons and Decision Trees."
Procesamiento del Lenguaje Natural. 2017.
- Ferrés, Daniel, Montserrat Marimon, and Horacio Saggion.
"YATS: Yet Another Text Simplifier."
In International Conference on Applications of Natural Language to Information Systems. 2016.
- Ronzano, Francesco, Luis Espinosa Anke, and Horacio Saggion.
"TALN at SemEval-2016 task 11: Modelling Complex Words by Contextual, Lexical and Semantic Features."
In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). 2016.
- Bella, Gabor, Fausto Giunchiglia, and Fiona McNeill.
"A Multilingual Ontology Matcher."
In Proceedings of the $10_{th}$ Workshop on Ontology Matching. 2015.

## 1.5 Thesis Outline

There are eight chapters in this thesis, each of these chapters contains an introduction, the main body and a summary of the key results and conclusions. A significant amount of the content in these chapters is derived from our publications (Saggion et al., 2016b; Abura'ed et al., 2017; AbuRa'ed et al., 2017; Abura'ed et al., 2018; AbuRa'ed et al., 2018; AbuRa'ed et al., 2020a). Most of the work in these papers is done in collaboration with other researchers, which is duly indicated wherever required.

In Chapter 2 we describe the related work. Section 2.1 describes related works in the field of automatic text summarization (ATS) for the domain of scientific texts. Section 2.2 describes related works directly tackling the automatic related work generation. Finally, section 2.3 describes related works to our experiments on generating abstractive summaries using Seq2Seq.

In Chapter 3 we describe a multi-level annotated corpus of scientific papers for scientific document summarization and cross-document relation discovery. Section 3.1 introduces our corpus that was created specifically for our research, but it is made available for the research community in order to provide them with means of fair comparisons of various summarization approaches. Section 3.2 presents a data set upon which we base our corpus creation. Section 3.4 describes our basic processing and analysis of the scientific papers. Section 3.5 describes the annotation process alongside the annotators' agreement. Section 3.6 presents the enrichment of the corpus and how it could benefit the community. Section 3.7 introduces several automatic systems we implemented to test how viable is our corpus. Section 3.8 presents the results across the automatic systems. Finally, section 3.9 summarizes our contributions.

In Chapter 4 based on an existing annotated data set of explicit and implicit citation sentences, we present experiments to identify implicit citations in scientific papers. Section 4.1 defines implicit citations and their purpose. Section 4.2 introduces the corpus our work is based on. Section 4.3 describes the experiments we ran to identify implicit citations. Section 4.4 provides the results of our system against the baseline.

In Chapter 5 we present several systems developed to participate in the Computational Linguistics Scientific Document Summarization Shared challenge which addresses the problem of summarizing a scientific paper taking advantage of its citation network (i.e., the papers that cite the given paper). Section 5.1 describes the shared task and the motivation behind it. Section 5.2 describes the data provided by the organizers of the shared task. Section 5.3, section 5.4 and section 5.5 present our participation in the first, second and

third iterations of the shared task respectively. Finally, section 5.6 summarises our contributions in the Computational Linguistics Scientific Document Summarization.

In Chapter 6 we present an extractive scientific papers' summarization system that we implemented to automatically generate related work reports. Section 6.1 motivates the need for related work sections and introduces our approach to generate them. Section 6.3 describes how we determine the most important sentences from a scientific paper. Section 6.4 presents our approach of ordering the selected sentences from the scientific papers to furnish the final related work report. Section 6.5 and section 6.6 presents our experiments against several baselines and their results, and finally section 6.7 summarizes our contributions in producing related work reports.

In Chapter 7 we compare different automatic methods to produce "descriptive" related work sections given as input the set of papers which have to be described. Section 7.1 we introduce our seq2seq approach over the data described in section 7.2. Our method is described in section 7.3 and it is based on pointer–generator neural networks with copy-attention technique and coverage mechanism (See et al., 2017; Wu et al., 2016). Section 7.4 implements the systems and section 7.5 presents the results against several baselines. Finally, section 7.6 presents the summary of our contributes.

# Chapter 2

# State of the Art

Traditional summarization (e.g., summarizing the news) is different from Automated related work summarization in many aspects. Automated related work summarization deals with a specific scientific domain which has specific features which makes it different from other types of summarization. Moreover, automated related work summarization must be written in the same way related works are written, therefore it should have a more regular and formalized structure than domain independent and generic summarization.

In contrast with generic summarization, state of the art generation/summarization has not been extensively explored. Key works in the area are: (Hoang & Kan, 2010) and (Hu & Wan, 2014), making (Hoang & Kan, 2010) to be the first to generate related work sections from a hierarchical topic-biased tree, and (Agarwal et al., 2011) who deal with multi-document scientific article summarization. Other studies investigate mainly single document scientific article summarization. In respect to that we will cover two main types of related works: Automated related work summarization and Automatic Text Summarization (ATS) in the domain of scientific texts.

## 2.1 Automatic text summarization (ATS) in the domain of scientific texts

Although research in summarization can be traced back to the 50s Luhn (1958) and even though a number of important discoveries have been produced in this area, automatic text summarization still faces many challenges given its inherent complexity. Scientific text summarization is of paramount importance and scientific texts were automatic summarization's first application domain Luhn (1958); Edmundson (1969). Several methods and techniques have already

been reported in the literature to produce text summaries by automatic means (Lloret & Palomar, 2012; Saggion & Poibeau, 2013).

Summarization of scientific documents has been addressed from different angles: in (Teufel & Moens, 2002) summarization is treated as a rhetorical classification task and Saggion & Lapalme (2002a) addressed the summarization problem as one of information extraction and text generation.

Teufel & Moens (2002) proposed a single-document summarization methodology to summarize scientific articles through rhetorical status of the article sentences. It selects material for summaries that can show the new contribution of the source article and situate it with respect to earlier work. They also provide a gold standard substantial corpus of conference articles in computational linguistics annotated with human judgments of the rhetorical status and relevance of each sentence in the articles. Several experiments measuring the judges' agreements on the annotations were made. They also presented a machine learning system for the classification of sentences by relevance and by rhetorical status. The algorithm of the system selects content from unseen articles and classifies it into a fixed set of seven rhetorical categories. Finally, a single-document summary is produced which can be used as a standalone summarization or to generate task-oriented and user-tailored summaries designed to give users an overview of a scientific field.

Saggion & Lapalme (2002a) presented a single document indicative-informative summarization system (SumUM) for technical text. Topics of the documents are identified (indicative part) and expanded according to the reader's interest (informative part). SumUM performs shallow syntactic and semantic analysis, concept identification, and text regeneration in order to motivate topics, describe entities, and define concepts, proposing the concept of dynamic summarization. To create a corpus, they used manually written abstracts by professionals provided from the journals: Library & Information Science Abstracts (LISA), Information Science Abstracts (ISA), and Computer & Control Abstracts and then they extracted the source documents for those abstracts from journals of Computer Science (CS) and Information Science (IS). They used human judgments in addition to Microsoft Word 97's Autosummarize and Extractor (Turney, 2002) for evaluation of both parts: indicativeness and informativeness in addition to acceptability of automatic summaries. Finally, the results of SumUM performed well when compared to the other summarization technologies.

The idea of using citing sentences to create data sets for paraphrase extraction was initially suggested by (Nakov et al., 2004b) who proposed an algorithm that extracts paraphrases from citing sentences using rules based on automatic

named entities annotation and the syntactic dependencies between them, such as gene / protein names. To improve the semantic interpretation and the retrieval of text for biomedical articles, they coined the term *citances* to refer to citing sentences, or the sentences that contain a citation. They also assume that it will be a gold mine of data for training algorithms to perform semantic analysis in the bioscience domain, and will improve the results of querying the domain literature. Finally, they show preliminary results on the problem of regularizing the different ways that the same concepts are articulated within a set of citances, by means of and refining existing techniques in automatic paraphrase generation.

da Cunha & Wanner (2005) studied a corpus of medical articles and their abstracts to detect the kind of information that should be selected for a summary taking into account numerous specialized information of the medical domain. They looked into enhancing the representation of text by applying different types of linguistic criteria to generalize the summarization developed model. They found linguistic clues that come from five sources: textual, lexical, discursive, syntactic and communicative structures. The system applies two types of lexical rules: increasing score rules and elimination rules. The increasing score rules increase the score of sentences with words from the title, first plural person verbal forms, words of list containing verbs or nouns that can be relevant and finally any numerical information in the Patients and Methods and results sections. On the other hand, eliminating sentences rules eliminate unnecessary information: references to tables and figures, references to computational aspects, references to previous work and finally references to definitions. Afterwards (Da Cunha et al., 2007) stated that the medical domain professionals use similar strategies to summarize their texts and they always extract the same content for their summaries. Therefore, evaluations of medical domain summaries can be done from the summary of the author of a medical article.

Qazvinian & Radev (2008b) proposed a model which uses a clustering approach to summarize a single article, which can be further used to summarize an entire topic. The main contribution is to use citation summaries and network analysis techniques which yield a summary of a single scientific article as a framework for future research on topic summarization. A corpus for studying clusters is constructed by mining small clusters from ACL Anthology Network (AAN) data (Joseph & Radev, 2007). Each cluster consists of a set of articles, in which the topic phrase is matched within the title or the content of papers in AAN. In particular, the five clusters that we collected this way, are: Dependency Parsing (DP), Phrased Based Machine Translation (PBMT), Text Summarization (Summ), Question Answering (QA), and Tex-

tual Entailment (TE). Finally, their model outperforms the current state of the art multi-document summarizing algorithms, Lexrank on this particular problem.

Mei & Zhai (2008) utilized citation information for summaries of a single scientific article in the computational linguistics domain. They proposed language modelling methods to incorporate features such as authority and proximity to estimate impact. The models exploit both the citation context and original content of a paper to generate impact-based summary which is used for facilitating the exploration of literature, it also helps to generate query. Experiment results on a SIGIR publication collection show that the proposed methods are effective for generating impact-based summaries.

Mohammad et al. (2009) performed preliminary experiments of the helpfulness of citation text to automatically generate technical surveys. Three types of input were used (full papers, abstracts and citation texts). They suggested utilizing citation information to generate surveys of scientific paradigms. Moreover, highlighting the importance of using citations from articles that in the framework of multi-document survey creation, citation texts can play a crucial role. Finally, they stated that summaries based on citation texts comprise important survey-worthy material that is not obtainable or hard to extract from abstracts and the full texts of papers. Likewise, they demonstrate that author abstracts contain information not present in citation texts and full texts. A similar study was carried out by (Ronzano & Saggion, 2016) using data from the BioSumm 2014 Challenge.

Oftentimes a work is cited in one sentence and then implicitly referred to again in later sentences. Finding all sentences that refer to a specific reference without using a formal reference notation is called implicit citation. Qazvinian & Radev (2010b) believed that implicit citations can be used by academics to understand major contributions in their research area more easily. They implemented probabilistic inference using a tuned Markov Random Field (MRF) (Metzler & Croft, 2005) model to propose a general framework that extracts such context from scientific papers. They model the sentences in an article and their lexical similarities. They used a tuned Markov Random Field to detect the patterns that context data create, and they also employ a Belief Propagation mechanism to detect likely context sentences. Finally, they tackled the problem of generating surveys of scientific papers. They used 10 published papers from the ACL Anthology Network (AAN) in various areas of NLP to create their data. Afterward they annotated the data to distinguish explicit and implicit citations between paper reference pairs. They started by looking at the explicit citations distribution each reference has received in a paper. Next they investigated the distance (number of sentences) between context

sentences and the closest citation. Such investigation proved the existence of a gap (the number of sentences between a context sentence and the closest context sentence or explicit citation to it) between sentences describing a cited paper. Moreover, they noticed that the majority of context sentences directly fell after or before a citation or another context sentence. As for the proposed method each sentence is characterized with a node and is given two scores (context, non-context), and such scores are updated to be in harmony with the neighbors' scores in the Markov Random Field (MRF) model. The assessment is done by comparing their results with the gold standard annotated data from their corpus. Qazvinian & Radev (2010b) also demonstrate the usefulness of the context sentences in generating surveys of scientific literature. Subsequently the experiments on generating surveys for the topics "Question Answering" and "Dependency Parsing" display how surveys generated using such (implicit) context information along with citation sentences have higher quality than those built using (explicit) citations alone.

Agarwal et al. (2011) tackled the multi-document summarization of scientific articles problem by an original unsupervised method, in which the source document cites a list of papers (also known as a co-citation). From each co-cited article, a topic based clustering of fragments were mined and ranked using a query produced from the context surrounding the co-cited list of papers. An overview was created by this analysis from the co-cited papers that relate to the context. They applied this approach to the 2008 ACL Anthology and called it SciSumm. They also evaluated the summarization system for appropriate content selection using gold standard summaries. Evaluation with gold standard summaries proves that their system summaries outperforms the ones by MEAD (Radev et al., 2004). Agarwal et al. (2011) discovers the comparable attributes of the co-cited articles using Frequent Term Based Clustering (Beil et al., 2002). The clusters generated in this process contain a set of topically related text fragments called TILES, which are extracted from the set of co-cited articles. The system pipeline operates as follows: first, the Text Tiling (Hearst, 1997) module produces tiles of text related to the citation context. Next, the clustering module is utilized to generate labelled clusters using the text tiles extracted from the co-cited papers. Finally, the summary presentation module is used to show the ranked clusters obtained from the ranking module.

Abu-Jbara & Radev (2011) have presented citation-based summaries in three stages: preprocessing, extraction, and post processing. Their experimentations demonstrate that their approach generates better summaries than several baseline summarization systems in which they started the pipeline with preprocessing by determining which pieces of text (sentences or fragments of

sentences) should be considered for selection during the extraction and which ones should be excluded. For the extraction step they used LexRank (Erkan & Radev, 2004) (a network based ranking algorithm equivalent to PageRank) to recognize the most salient sentences within clusters. LexRank first summarizes multi-documents and builds a cosine similarity graph of all the candidate sentences. Then it discovers the most central sentences by performing a random walk on the graph. LexRank sets each citation sentence as a node and links sentences with weighted edges where weights are the similarity between the sentences (measured with cosine). The most central papers are selected based on the main facts of the corresponding cluster (e.g., representative sentences). Finally, post processing aims to refine the selected sentences and create the final summary by avoid repeating the author's names and the publication year in every sentence and then replace the reference with a suitable personal pronoun for the sentences. To evaluate their results they used ROUGE (Lin, 2004), a widely used and recognized automated summarization evaluation method. The results produced show that their approach outperforms all the baseline techniques (MEAD (Radev et al., 2004), LexRank (Erkan & Radev, 2004), citation-based summarizer(QR08) (Qazvinian & Radev, 2008b), randomly selected sentences from the set of citation sentences and another three baselines were variations of their system produced by removing one component from the pipeline at a time). Their system attains higher ROUGE score in the testing set. Moreover, they stated that sentence filtering has a significant impact on the results. It also shows that the classification and clustering components both improve the extraction quality.

Nanba et al. (2011) utilize citances as features for classifying papers into topics. They also propose to use citances as part of a support system for writing review articles on specific topics. Given a document, their system finds the citances originating from other papers. They analyze citation sentences and automatically categorize citations into three groups using 160 predefined phrase-based rules. The three groups are: citations that show other researcher's theories and methods for the theoretical basis, citations to point out problems or gaps in the related works, and finally, a group for citations that do not belong to the first two groups. This categorization is then used to build a tool for survey generation. Their aim is to automatically generate review articles in a specific subject domain using citation types as the foundation for the classification of papers. Counting on two main citation categories (works that provide a supporting basis for the Citing Paper, works that have a contrasting or 'negative' relationship), but also add a third 'others' category to designate some form of unspecified relationship exists between the citing and cited papers. Groups of 'cue phrases' (including discourse markers, lex-

ical usage, specific phrases), are used to classify citations into the different categories but these cues are heuristically motivated rather than theoretically based. Finally, the outcome of their approach based on bibliographic coupling seems more effective than others. Moreover, they evaluated their approach with manually classified papers.

Qazvinian et al. (2013) proposed C-LexRank, a graph-based summarization method. This method models a set of citing sentences as a network in which vertices are sentences and edges characterize their lexical similarity. They recognized vertex communities (clusters) in this network to produce summaries, by mining representative sentences from the citation summary network. Therefore, a good sentence selection set from the citation summary network will include vertices that are similar to many other vertices and which are not very similar to each other. On the other hand, a bad selection set can include sentences that represent only a small set of vertices in the graph. They summarized 30 single scientific articles selected from 6 different topics in the ACL Anthology Network (AAN). Using bibliometric lexical link mining that exploits the structure of citations and summarization techniques they compared and contrasted the usefulness of abstracts and of citations from multiple research papers in automatically generating a technical summary on a given topic. Finally, the authors compared C-LexRank with the state-of-the-art summarization systems where this method outperforms a leverage diversity method (Mei et al., 2010), a random summarization method (Erkan & Radev, 2004) and LexRank (Zajic et al., 2007).

Jha et al. (2013) implemented a system that can summarize a topic starting from a query as input. The articles are retrieved from that query using a simple heuristic called Restricted Expansion and the system then select sentences from these articles to generate a survey of the topic. Jha et al. (2013) describe an evaluation corpus they generated by manually extracting factoids, or information units, from 47 gold standard documents on seven topics in Natural Language Processing. They also manually annotated 2,625 sentences with these factoids (around 375 sentences per topic) to build an evaluation corpus. They experimented on three context models: *Centroid*: The centroid of a set of documents is a set of words that are statistically important to the cluster of documents creating centroid-based summarization. *Lexrank* (Erkan & Radev, 2004) and *C-Lexrank* (Qazvinian & Radev, 2008b). Given a set of sentences, it first creates a network using these sentences and then runs a clustering algorithm to partition the network into smaller clusters that represent different aspects of the paper. Finally, for each summary, they determine two evaluation metrics. Pyramid score (Nenkova & Passonneau, 2004) is calculated by treating the factoids as Summary Content Units (SCUs). The second

is the Unnormalized Relative Utility score (Radev & Tam, 2003) which is calculated by using the factoid scores of sentences based on the method proposed in (Yih & Qazvinian, 2012). They find that the Lexrank method beats other sentence selection methods on both evaluation metrics.

Jaidka et al. (2013) performed studies in the domain of multi-document summarization, and established a literature review framework on a deconstruction of human-written literature review sections in information science research papers. They studied scientific papers to be able to compare them, to identify new problems, to place a work inside the current literature and to elaborate new research propositions. The first part of the study offers the results of a multi-level discourse analysis to examine their discourse and content features. They created a framework for literature reviews focusing on macro-level document structure and the sentence-level templates, as well as the information summarization strategies. The second part of this study debates visions from this analysis, and how the framework can be adapted to automatic summaries resembling human written literature reviews. Summaries are evaluated against human written summaries and evaluators comments are discussed to express recommendations for future work.

Cohan & Goharian (2017) proposed a summarization approach for scientific articles which takes advantage of citation-context and the document discourse model. They also leverage the inherent scientific article's discourse for producing better summaries. Their proposed method effectively improves over existing summarization approaches (greater than 30% improvement over the best performing baseline) in terms of ROUGE scores on TAC2014 scientific summarization dataset. The dataset they use for evaluation is in the biomedical domain.

Hashimoto et al. (2017) examined a problem of automatically generating a synthesis matrix for scientific literature review. A synthesis matrix is a table that summarizes various aspects of multiple documents. They formulate the task as multi document summarization and question-answering tasks given a set of aspects of the review based on an investigation of system summary tables of NLP tasks. Their system consists of two steps: sentence ranking and sentence selection. In the sentence ranking step, the system ranks sentences in the input papers by regarding aspects as queries. They use LexRank and also incorporate query expansion and word embedding to compensate for tersely expressed queries. In the sentence selection step, the system selects sentences that remain in the final output. Specifically emphasizing the summarization type aspects, they treated this step as an integer linear programming problem with a special type of constraint imposed to make summaries comparable. Finally, they evaluated their system using a dataset they created from the ACL

Anthology. The results of manual evaluation demonstrated that their selection method using comparability improved performance.

The semantic link network is a semantics modeling method for effective information services. Sun & Zhuge (2018) suggested that performing reinforcement ranking on the Semantic Link Network of various representation units within a scientific paper (word, sentence, paragraph and section) can significantly improve extractive summarization of paper. They claimed that it also verifies the significance of Semantic Link Network in representing and understanding the content of a paper. The proposed approach creates stability in single document summarization on both scientific papers and short news text in DUC 2002 test documents and performs better when documents have more structural information modelled by Semantic Link Network.

Zhang et al. (2019) proposed an approach to generate automatic summarization based on 5W1H (who, what, whom, when, where, how) event structure. Each scientific paper is treated as a scientific research event, whose elements are distributed in the full text of scientific paper. When, where, who, and what are corresponding to publication time, publication venue, authors, and title, respectively, which are essential parts of metadata of scientific literature. Sentences in literature are classified and selected for different elements of events by relevance, and then, the importance of each candidate sentence is calculated. Top-k relevant and important sentences are selected to formulate event-based summarization. They compared with existing summarization results or abstracts given by authors. Experimental results contain more detailed information with the 5W1H event structure, which is convenient for researchers to search and browse the brief description of scientific and technical information distributed in massive scientific literature.

Xu et al. (2019) employed a hierarchical attention model to learn document structure from all the papers for summarization. They utilized attention mechanism to capture relations among document structure and to learn semantic information on document discourse levels and judge the importance of each sentence according to its surroundings by the information obtained. Moreover, they automatically constructed a scientific literature data set consisting of surveys and their references. Finally, they evaluated their proposed model on that dataset with ROUGE metrics.

Erera et al. (2019) presented a system (named: IBM Science Summarizer) providing summaries for Computer Science publications. Through a qualitative user study, they identified the most valuable scenarios for discovery, exploration and understanding of scientific documents. Based on these findings, they built a system that retrieves and summarizes scientific documents for a

given information need, either in form of a free-text query or by choosing categorized values such as scientific tasks, datasets and more. IBM Science Summarizer summarizes the various sections of a paper independently, allowing users to focus on the relevant sections for the task at hand. In doing so, the system exploits the various entities and the user's interactions, like the user query, in order to provide a relevant summary. Their system ingested 270,000 papers, and its summarization module aims to generate concise and detailed summaries. They validated their approach with human experts.

### 2.1.1 CL-SciSumm Shared Task

Recent studies proposed to take advantage of the scientific paper's citation network mainly to approach scientific literature summarization. Therefore, new generations of scientific summarization approaches have emerged which take advantage of the citations as we mentioned at Section 2.1.

In the CL-SciSumm challenge, given a cluster of $n$ documents where one is a Reference Paper (RP) and the $n-1$ remaining documents are papers (i.e., Citing Papers (CPs)) citing the reference paper, participants of the challenge have to develop automatic procedures to simulate the following tasks: Task 1: to identify which reference paper sentences have been cited and also tried to identify the discourse facet of the reference sentence. Task 2: generating a structured summary of the reference paper with up to 250 words from the cited text spans. Many systems reported their methods to approach this challenge. In the following we review key systems in each edition of the challenge.

#### 2.1.1.1 The CL-SciSumm Shared Task (2016)

Li et al. (2016) used an SVM classifier with a topical lexicon to identify the best-matching reference spans for a citance, using IDF similarity, Jaccard similarity and context similarity. They finally submitted six system runs, each following a variant of similarity measures and approaches: fusion (combination of all methods), Jaccard Cascade, Jaccard Focused, SVM and two other voting methods.

Conroy & Davis (2015) attempted to solve Task 2 with an adaptation of a system developed for the TAC 2014 BioMedSumm task [16]. They provided the results from a simple vector space model, wherein they used a TF representation of the text and non negative matrix factorization (NNMF) to estimate the latent weights of the terms for scientific document summarization. They

[16] https://tac.nist.gov/2014/BiomedSumm/

also provide the results from two language models based on the distribution of words in human-written summaries.

Moraes et al. (2016) used SVM with the subset tree kernel, a type of convolution kernel. Computed similarities between three tree representations of the citance and reference text formed the convolution kernel. Their setup scored better than their TF.IDF baseline. They submitted three system runs with this approach.

Cao et al. (2016), for Task 1A, use SVM rank with lexical and document structural features to rank reference text sentences for every citance. Task 1B was tackled using a decision tree classifier. They modeled summarization as a query-focused summarization task with citances as queries. They generate summaries (Task 2) by improvising on a manifold ranking method.

Table 2.1 shows the top performing systems at Task 2 of the Cl-Scisumm 2016 shared task.

| Team | Approaches |
|---|---|
| Li et al. (2016) | Fusion, Jaccard Cascade, Jaccard Focused, SVM and two other voting methods. |
| Conroy & Davis (2015) | TF-representation with non-negative matrix factorization (NNMF) |
| Moraes et al. (2016) | SVM with the subset tree kernel |
| Cao et al. (2016) | Query-focused summarization task with citances as queries |

**Table 2.1:** Top performing systems in Task 2 at the CL-SciSumm 2016

#### 2.1.1.2 The CL-SciSumm Shared Task (2017)

Li et al. (2017) followed an approach similar to their 2016 system submission (Jaidka et al., 2013). Lauscher et al. (2017a) also participated in all of the tasks. For Task 1A, they used supervised learning to rank paradigm to rank the sentences in the reference paper using features such as lexical similarity, semantic similarity, entity similarity and others. They formulated Task 1B, as a one-versus-all multi-class classification. They used an SVM and a trained Convolutional Neural Network (CNN) for each of the five binary classification tasks. For Task 2, they clustered the sentences using single pass clustering algorithm using a Word Mover's similarity measure and sorted the sentences in each cluster according to their Text Rank score. Then they ranked the clusters according to the average Text Rank score. Top sentences were picked from the clusters and added to summary until the word limit of 250 words was reached.

Ma et al. (2017) participated in all of the tasks (Tasks 1A, 1B and 2). For Task 1A, they used a weighted voting-based ensemble of classifiers (linear Support Vector Machines (SVM), SVM using a radial basis function kernel, Decision Tree and Logistic Regression) to identify the reference span. For Task 1B, they created a dictionary for each discourse facet and labeled the reference span with the facet if its dictionary contained any of the words in the span. For Task 2, they used bisecting K-means to group sentences in different clusters and then used maximal marginal relevance to extract sentences from each cluster and combine into a summary.

Dipankar Das & Pramanick (2017) participated in all of the tasks. For Task 1A, they defined a cosine similarity between texts. The reference paper's sentence with the highest score is selected as the reference span. For Task 1B, they represent each discourse facet as a bag of words of all the sentences having that facet. Only words with the highest TF.IDF values are chosen. To identify the facet of a sentence, they calculated the cosine similarity between a candidate sentence vector and each bag's vector. The bag with the highest similarity is deemed the chosen facet. For Task 2, a similarity score was calculated between pairs of sentences belonging to the same facets. If the resultant score is high, only a single sentence of the two is added to the summary.

Table 2.2 shows the top performing systems at Task 2 of the Cl-Scisumm 2017 shared task.

| Team | Approaches |
|---|---|
| Li et al. (2017) | Fusion, Jaccard Cascade, Jaccard Focused, SVM and two other voting methods. |
| Lauscher et al. (2017a) | Single pass clustering algorithm using a Word Mover's similarity measure. |
| Ma, S. (2017) | K-means to group sentences in different clusters with the maximal marginal relevance. |
| Dipankar Das & Pramanick (2017) | A similarity score of sentences belonging to the same facets. |

**Table 2.2:** Top performing systems in Task 2 at the CL-SciSumm 2017

#### 2.1.1.3 The CL-SciSumm Shared Task (2018)

Ma et al. (2018) participated in all of the tasks (Tasks 1A, 1B and 2). For Task 1A, they used the same method they used in their 2017 participation, a weighted voting-based ensemble of classifiers (linear Support Vector Machines (SVM), SVM using a radial basis function kernel, Decision Tree and

Logistic Regression) to identify the reference span. For Task 1B, they used a dictionary for each discourse facet, a supervised topic model, and XGBOOST. For Task 2, they grouped sentences into three clusters (motivation, approach and conclusion) and then extracted sentences from each cluster to combine into a summary.

Ma et al. (2018) developed models based on their 2017 system (previous year participation). For Task 1A, they adopted Word Movers Distance (WMD) and improve LDA model to calculate sentence similarity for citation linkage. For Task 1B they presented both rule-based systems, and supervised machine learning algorithms such as: Decision Trees and K-nearest Neighbor. For Task 2, in order to improve the performance of summarization, they also added WMD sentence similarity to construct new kernel matrix used in Determinantal Point Processes (DPPs).

Debnath et al. (2018) participated in all of the tasks (Tasks 1A, 1B and 2). For task 1A and 1B they extracted each Citing Papers (CP) text span that contains citations to the Reference Paper (RP). They used cosine similarity and Jaccard Similarity to measure the sentence similarity between CPs and RP, and picked the reference spans most similar to the citing sentence (Task 1A). For Task 1B, they applied rule based methods to extract the facets. For Task 2, they built a summary generation system using the OpenNMT tool.

Table 2.3 shows the top performing systems at Task 2 of the Cl-Scisumm 2018 shared task.

| Team | Approaches |
|---|---|
| Ma et al. (2018) | Grouped sentences into three clusters (motivation, approach and conclusion) |
| Ma, S. (2017) | Added WMD sentence similarity to construct new kernel matrix. |
| Dipankar Das & Pramanick (2017) | Summary generation system using the OpenNMT tool |

**Table 2.3:** Top performing systems in Task 2 at the CL-SciSumm 2018

## 2.2 Automated related work summarization

Hoang & Kan (2010) and Vu (2010) presented a related work summarization system that creates a topic-biased summary of related work for a target paper given multiple scientific articles together with a topic hierarchy tree as an input. Hoang & Kan (2010) and Vu (2010) essentially envisioned a Natural Lan-

guage Processing application named ReWoS – Related Work Summarization – that assists in creating a related work summary. This summary then takes in a set of keywords arranged in a hierarchical fashion that describes a target paper's topics then generates an extractive summary identifying the appropriate sentences for general topics as well as detailed ones. The initial results show an improvement over general multi-document summarization baselines in a human evaluation. Hoang & Kan (2010) stated that solving the previous problem involves the following three tasks: 1) Finding relevant documents; 2) Identifying the relevant aspects of these documents in relation to the current work worth summarizing; and 3) Generating the final topic-biased summary. However, they only focused on the third task - the construction of a related work section, given a structured input of the topics as the previous two were already tackled by existing works in Natural Language Processing and recommendation systems communities. Specifically, the work done by (Nallapati et al., 2008) on Citation prediction is a growing research area with interest in individual paper citation patterns and at foreseeing citation growth over time within a community. Subsequently an automatic key phrase extraction task from scientific articles was introduced in SemEval-2, partially addressing Task 11. Also, the work of (Mohammad et al., 2009) on automatic survey generation is becoming a growing field within the summarization community.

Hoang & Kan (2010) also stated that three things should be considered to generate a summary. First, a mandatory input is needed for the summarization process identified as a high-level rhetorical structure in a form of a topic tree. Second, summaries can be seen as transitions along the topic hierarchy tree. Third, sentences either describe generic or specific topics. Generic topics are often characterized by background information. This include definitions or descriptions of a topic's purpose. In contrast, detailed information forms the substance of the summary and often describes key related work that is attributable to a specific author. In order to study the problem, they created a dataset (called RWSData) based on twenty articles at cherished Natural Language Processing venues. Their approach consists of two modules: Specific Content Summarization (SCSum) that provides general background sentences and the General Content Summarization (GCSum) which furnishes specific contributions of authors. Sentences containing pronouns or indication phrases (e.g., "we", "this approach") as a description of own work will be classified and directed to Specific Content Summarization (SCSum); otherwise they will be directed to the GCSum workflow. GCSum extracts sentences holding valuable background information. Such general content sentences were divided into two groups: indicative and informative. Informative sentences stretch detail on a specific facet of the problem. They often give definitions, purpose

or application of the topic and it is best if they are extracted from the source articles themselves. In contrast, indicative sentences are inserted to make the topic transition clear and rhetorically sound; such sentences can be easily generated by templates. On the other hand, SCSum intends to extract sentences that hold detailed information about a specific writer's effort that is relevant to one of the input topics from the set of sentences that exhibit the author-as-agent. Finally, they evaluated ReWoS in contrast to two baseline systems: LEAD and MEAD. The LEAD baseline represents each of the cited articles with an equal number of sentences. First *n* sentences of each processed article are extracted. MEAD (Radev et al., 2004) offers a set of different features that can be parameterized to create resulting summaries. They utilized two features of *centroid* and that similarity to centroid was measured with cosine. Furthermore they performed a human evaluation to expand extra fine-grained assets of their system, after which they asked eleven human judges to evaluate [Correctness, Novelty, Fluency and Usefulness] of the generated summary. Automatic evaluation was performed on four summaries with ROUGE (Lin, 2004). Summaries came from a LEAD-based system, MEAD system, proposed ReWoS system without (ReWoS-WCM) and with (ReWoS-CM) the context modeling in Specific Content Summarization (SCSum). The idea behind context modeling is to add five to seven sentences into the Specific Content Summarization (SCSum) module to enrich the context of each sentence classified to that module. The results of their system showed that the MEAD baseline system outperforms both LEAD baseline and ReWoS–WCM (without context modeling). Solitary ReWoS–CM (with context modeling) is better than others in relation to all ROUGE variants. The motivation behind context modeling is derived from the belief that they should also select nearby sentences within a contextual window of five sentences to better represent the information meant to be described.

Hu & Wan (2014) investigated on the task of producing a related work section for a target paper, provided a set of Reference Papers along with a target academic paper which has no related work section as input. They believed the generated text can be used as a draft to continue a related work section that describes the related works and addresses the relationship between the target paper and the Reference Papers presented as input. Automatic Related Work Generation system (ARWG) exploits the Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999) to solve this problem. They used the PLSA model to divide the sentence set of the given papers into different topic-biased parts, and then applies regression models to learn the standing (ranking) of the sentences. Finally, it utilizes an optimization framework to produce the related work section. Their evaluation results on a test set of 150 target papers

sideways with their Reference Papers show that ARWG can indeed generate related work sections with improved quality than those of baseline methods. The baseline methods are: MEAD (Radev et al., 2004) and LexRank (Erkan & Radev, 2004). MEAD is an open-source extractive multi-document summarizer. LexRank is a multi-document summarization system which is based on a random walk on the similarity graph of sentences. A user study is also carried on to demonstrate that ARWG can achieve a development over generic multi-document summarization baselines. The need for the target paper itself, without the related work section, is motivated by the belief that the abstract and introduction sections have valuable information which contribute in generating the related work section. As for the Reference Papers, they only study and extract the abstract, introduction, related work and conclusion sections, since other sections corresponding to method and evaluation sections always describe in too much details of the specific work and they are not suitable for this task. In comparison (Hoang & Kan, 2010) did not require a target paper as part of the input, but, requiring a topic tree as part of the input as they did is considered a limitation.

## 2.3 Sequence to Sequence Summarization

Recent approaches to abstractive summarization include the following. Cohan et al. (2018) developed an abstractive model for summarizing scientific papers. The model includes a hierarchical encoder, capturing the discourse structure of the document and a discourse-aware decoder that generates the summary. The decoder attends to different discourse sections and allows the model to more accurately represent important information from the source resulting in a better context vector. They introduce two large-scale datasets of long and structured scientific papers obtained from arXiv and PubMed to support both training and evaluating models on the task of scientific paper summarization. Finally, their model outperforms two abstractive seq2seq baselines alongside three extractive baselines including LexRank.

Bražinskas et al. (2019) has addressed opinions summarization in which they analyze multiple reviews from users over different products and businesses and then created text summaries that reflect subjective information expressed in these reviews. To overcome and rely on large quantities of document-summary pairs as used in supervised abstractive summarization which are expensive to acquire, they used an unsupervised approach which uses a hierarchical variational auto-encoder (VAE) model and utilizes two sets of latent variables. One is a continuous variable that captures latent semantics of a group of reviews and the other is a continuous variable to encode latent se-

mantics of each individual review in the group. The final summaries are produced by the decoder that uses the information stored at the second continuous variable. Chu & Liu (2018) also utilized an unsupervised abstractive summarization model that uses an auto-encoder where the mean of the representations of the input reviews (i.e. mean over the hidden and cell states of all the input reviews) decodes to a reasonable summary-review while not relying on any review-specific features. They implemented variants of the proposed architecture and analyzed the different variants. Finally, Baziotis et al. (2019) also used an unsupervised abstractive model to develop a sequence-to-sequence-to-sequence autoencoder ($SEQ^3$), where the first sequence is the input, the second sequence is the compressed sentence and the last sequence consists of reconstructed sentences. $SEQ^3$ consists of two chained encoder-decoder pairs, with words used as a sequence of discrete latent variables.

Zhang et al. (2018) proposed a latent variable extractive model that views labels of sentences in a document as binary latent variables. The latent model maximizes the likelihood of human summaries given selected sentences where loss comes directly from gold summaries. They modeled instances of sequence labeling in which a document is viewed as a sequence of sentences and the model is expected to predict a *true* or *false* label for each sentence, where *true* indicates that the sentence should be included in the summary. Their system has three parts: a sentence encoder to convert each sentence into a vector, a document encoder to learn sentence representations given surrounding sentences as context, and a document decoder to predict sentence labels based on representations learned by the document encoder. Finally, they use CNN/Dailymail data set (Hermann et al., 2015) for their experiments and they compare their system with other extractive and abstractive systems.

Lastly, a hybrid method for summarization of multiple related work sections of scientific articles has recently been proposed (Altmami & Menai, 2018). In this work a semantic graph-based approach is used to handle the redundancy of citation sentences by reducing the sentence graph while preserving its properties. Using cross-document structure theory (CST) to analyze multi-documents i.e. related work section, they discover semantic relations to further reduce redundancy in the set of citation sentences.

# Chapter 3

# A Corpus for Scientific Document Summarization

In this chapter we will describe a multi-level annotated corpus of scientific papers for scientific document summarization and cross-document relation discovery. The corpus was created specifically for our research, but it is made available for the research community in order to provide means for fair comparisons of various summarization approaches.

## 3.1 Introduction

Related work sections contain information that can link scientific papers in a citation network, in order to take advantage of the scientific paper's citation network to approach scientific literature summarization we have developed a manually annotated, machine readable data-set of related work sections, cited papers (e.g. references) and sentences, together with an additional layer of papers citing the references. Additionally, we present experiments on the identification of cited sentences, using as input citation contexts. The corpus alongside the gold standard are made available for use by the scientific community.

Good related work sections are difficult to produce since they require the author to select, contrast, and organize key information from several sources. Although there have been a number of studies and guidelines on their functions, types and forms (Khoo et al., 2011; Jaidka et al., 2013; Pautasso, 2013), our understanding of what is a good related work section is still limited.

There is a number of corpora related to the our work. A large-scale, human-annotated scientific papers corpus is provided by Yasunaga et al. (2019a). It provides over 1,000 papers in the ACL anthology with their citation networks

(e.g. citation sentences, citation counts) and their comprehensive, manual summaries. There is also a data-set which has been created for the Computational Linguistics Scientific Document Summarization Shared Task which started in 2014 as a pilot project (Jaidka et al., 2014a) and which is now a well developed challenge in its fourth year (Jaidka et al., 2017d,c). The shared task provided training data structured in clusters of reference and Citing Papers together with manual annotations indicating, for each citance, the text span(s) in the Reference Paper that best represent the citance, as well as their corresponding facets. One of the main problems with the data-set is the lack of agreed manual annotations since only one annotator was in charge of annotating each cluster. Those previously mentioned data-sets are considered the closest to our corpus. However they are only equivalent to what we name Level 2 of our corpus and they provide no link between a target paper with a segmented related work section that explicitly mentions a set of Reference Papers.

There are also corpora for the study of scientific text mining and summarization. (Saggion & Lapalme, 2002b) have aligned 200 abstracts produced by professional abstractors to their source documents to investigate how to produce non-extractive indicative abstracts. (Fisas Elizalde et al., 2016) have created a multi-layered annotated corpus from 40 articles in the domain of Computer Graphics. Sentences are annotated with respect to their role in the argumentative structure of the discourse. It specifies the purpose of each citation in the scientific papers and it identifies special features of the scientific discourse such as advantages and disadvantages. In addition, a grade is allocated to each sentence according to its relevance for being included in a summary. Athar & Teufel (2012b) created a citation context corpus from the ACL Anthology Network (AAN) which consists of 852 papers that are citing 20 papers. The corpus contains 1,034 paper–reference pairs and 203,803 sentences. It is manually annotated by identifying the sentences in the citation context. It also contains a sentiment annotation as well (negative, positive, objective/neutral). Teufel (2006) created a corpus based on 80 Argumentative Zoning-annotated conference articles in the Computational Linguistics domain. The corpus was created to research classifying academic citations in scientific articles according to author claims.

Finally, based on the SAPIENT tool (Liakata et al., 2009) and an annotation guideline (Liakata & Soldatova, 2008) a corpus of 225 papers was created and manually annotated with CISP (Core Information about Scientific Papers) concepts. These papers cover topics in physical chemistry and biochemistry. The Corpus was developed to add value to scientific papers through semantic markup.

Our corpus expands considerably the data-set of related work sections used

in (Hoang & Kan, 2010) by providing: (i) related work sections, (ii) a manually annotated layer of cited papers and sentences, (iii) Citing Papers referring to the cited papers in the related work section, and (iv) a layer of rich linguistic, rhetorical, and semantic annotations computed automatically. While the manually identified cited sentences are useful to support the study of sequence to sequence models in scientific summarization, the new layer of Citing Papers facilitates the test of citation-based summarization approaches (Qazvinian & Radev, 2008b; Jaidka et al., 2014b) which rely on citation networks to assess sentence relevance. We organize the documents in: Target Papers, Reference Papers, and Citing Papers forming a two-level network. Level 1 contains Target Papers with their related work sections in which we are interested and, which cite a set of Reference Papers. Level 2 extends the corpus by adding a layer representing a set of scientific papers explicitly citing the Reference Papers in Level 1.

The contributions of this chapter are the following:

- The corpus has been manually annotated (3 annotators) and automatically processed;
- We also present experiments to assess several text representation mechanisms (e.g. lemmas, embeddings, synsets) for the retrieval of sentences likely to be cited by scientific papers comparing system results to the gold standard annotations;
- The corpus is available for research and development purposes in two versions[17]; one version contains the manual annotations (agreed cited sentences) and the other contains the full machine readable corpus with the automatic analysis just described;

## 3.2 RWSData Dataset

The RWSData data-set (Hoang & Kan, 2010) is a publicly available resource that includes twenty articles from sources such as the Special Interest Group on Information Retrieval (SIGIR), the Association for Computational Linguistics (ACL), the North American Chapter of the Association for Computational Linguistics (NAACL), the Empirical Methods for Natural Language Processing (EMNLP) and the International Conference on Computational Linguistics (COLING). Hoang & Kan (2010) extracted the related work sections directly from those research articles as well as several references cited in the related work sections (references to books and Ph.D. theses were removed). All the scientific papers provided in the RWSData are in PDF format with no

[17] http://taln.upf.edu/sciencecorpus

further analysis. Moreover, the data-set provides no mapping between the related works section citations and the sentences in the Reference Papers that are being cited making it challenging to use such data-set for scientific papers summarization. An example of a segmented related work section of a Target Paper can be seen at Table 3.1. Venugopal et al. (2009) have cited six Reference Papers in their related work section. They also made some claims on their own. The claims are considered the Target Paper's authors' opinion while all the citations to the references reflect the opinions of the Reference Paper's authors.

| # | Text | RP or Claim |
|---|---|---|
| 1 | there have been significant efforts in ... | claim |
| 2 | we survey the work most closely related to our approach . | claim |
| 3 | (May and Knight 2006) extract nbest lists containing unique ... while (Kumar and Byrne 2004) use the minimum bayes risk ... | (May and Knight 2006) (Kumar and Byrne 2004) |
| 4 | (Tromble et al. 2008) extend this work to lattice structures . | (Tromble et al. 2008) |
| 5 | all of these approaches only marginalize ... | claim |
| 6 | ... work by (Blunsom et al. 2007) propose a ... | (Blunsom et al. 2007) |
| 7 | (Matsusaki et al. 2005) and (Petrov et al. 2006) propose automatically learning annotations that add information ... | (Matsusaki et al. 2005) (Petrov et al. 2006) |
| 8 | in our work , we focused on approximating ... | claim |
| 9 | the methods described above might improve .... | claim |

**Table 3.1:** An Example of the related work section of (Venugopal et al., 2009) in the corpus.

## 3.3 Corpus Extension over the RWSData Dataset

We extracted the same twenty Target Papers considered in (Hoang & Kan, 2010), then for each paper we collected the Reference Papers mentioned in its related work section. Afterwards, for each Reference Paper we collected multiple scientific papers citing it. This extra layer would allow us to experiment with the citation networks which could allow citation network summarization systems to be implemented over the extended corpus. Figure 3.1 shows in details how our data is organized, a Target Paper containing a related work section alongside the Reference Papers which it cites and, in turn, for each Reference Paper, a set of scientific papers citing it. The RWSData data-set is the raw data on level 1 while our extension added the Citing Papers for the Reference Papers and the (manually identified) links between citing and cited sentences.

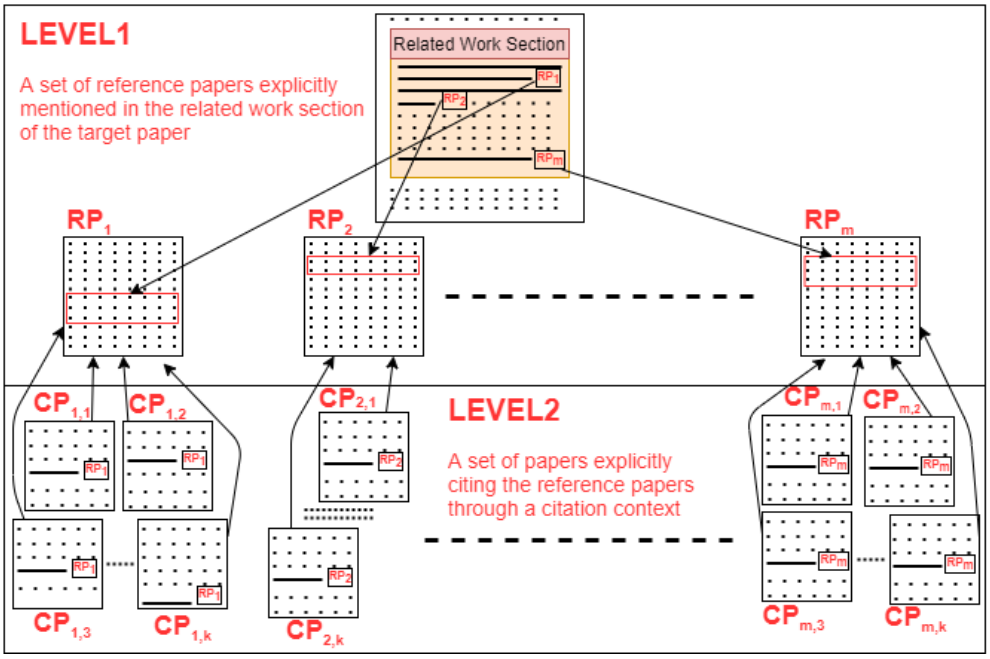


**Figure 3.1:** Our corpus outline presenting a target paper, a set of reference papers (Level 1) and for each reference paper a set of citing papers (Level 2)

### 3.3.1 Data Collection

The extension of the corpus was done by adding Citing Papers for each Reference Paper. The Citing Papers were collected from Microsoft Academic Graph (MAG) (Tang et al., 2008; Sinha et al., 2015; Wade, 2015; Herrmannova & Knoth, 2016), Semantic Scholar (Xiong et al., 2017; Valenzuela et al., 2015) and the ACL Anthology Network (AAN) (Radev et al., 2013). We queried the APIs of both Semantic scholar and Microsoft Academic Graph in order to obtain detailed information for the scientific papers. Microsoft Academic Graph (MAG) (Tang et al., 2008) is a diverse graph containing scientific publication records, citation relationships between those publications, as well as metadata. Semantic Scholar (Valenzuela et al., 2015) is a publicly available search service with millions of indexed articles. Semantic Scholar identifies citations where the cited publication has a significant impact on the citing publication, making it easier to understand how publications build upon and relate to each other. It also has what is named "influential citations" which are determined by using a machine-learning model analyzing a number of factors including the number of citations to a publication, and the surrounding context for each citation (Valenzuela et al., 2015).

The ACL Anthology Network (AAN) (Radev et al., 2013) is a wide-range manually curated networked database of citations and summaries in the field of Computational Linguistics. AAN provides citation and collaboration networks of the articles included in the ACL Anthology (Bird et al., 2008) (excluding book reviews). The data sources were collected starting with Semantic Scholar, then MAG and, finally, ACL. The Citing Papers were collected from the same source as the Reference Paper. We kept the most cited or most influential papers depending on the source from where the papers were collected. Overall, we collected up to 15 Citing Papers for each Reference Paper (with an average of 12 per Reference Paper). Each one of these sources has a different representation of the scientific papers and stores the meta-data in a different way. We stored the documents with the same ID they have on the equivalent source and we indexed the meta-data of all the papers provided from semantic scholar and the ACL anthology using Elasticsearch (Gormley & Tong, 2015). Table 3.2 presents examples of Citing Papers names. Scientific papers provided by MAG has been named with a numeric number of around 10 digits (i.e. Citing Paper ID). ACL has the format of LDD-DDDD (L: letter and D: digit). Finally, semantic scholar has a mixture of letters and digits. This will allow any mapping between a Citing Paper in the corpus and its source. Finally, the Target Papers and the Reference Papers were named the same as (Hoang & Kan, 2010)'s corpus.

| Source | ID |
|---|---|
| ACL | C08-1013 [18] |
| Semantic Scholar | 5dbf9d4c177a1cd207ccf205c7e223b90d0d867b [19] |
| MAG | 2055543848 [20] |

**Table 3.2:** Examples of names we adopt for the citing papers in the corpus.

[18]Callison-Burch, Chris, Trevor Cohn, and Mirella Lapata. "Parametric: An automatic evaluation metric for paraphrasing." Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008). 2008.

[19]Zhao, Shiqi, et al. "Extracting paraphrase patterns from bilingual parallel corpora." Natural Language Engineering 15.4 (2009): 503-526.

[20]Virga, Paola, and Sanjeev Khudanpur. "Transliteration of proper names in cross-lingual information retrieval." Proceedings of the ACL 2003 workshop on Multilingual and mixed-language named entity recognition-Volume 15. Association for Computational Linguistics, 2003.

## 3.4 Corpus Basic Data Processing

We converted the PDF documents for the entire corpus into GATE documents (Maynard et al., 2002) using three converters; Grobid (Lopez, 2009), PDF Digest (Ferrés et al., 2018) and PDFX (Constantin et al., 2013). The three converters provide basic information about each scientific paper contents including: title, authors, affiliations, abstract and paper sections. Finally, we also identified the sentences of each scientific paper by annotating a sentence ID for the GATE documents. This ID was used to help map the sentences during the annotation process. Table 3.3 provides information about the different paper types: Target Papers (TP), Reference Papers (RP) and Citing Papers (CP). It shows the number of papers, sentences and tokens alongside their averages.

| Paper Type | # | #Sentences | avg#Sentences | #Tokens | avg#Tokens |
|---|---|---|---|---|---|
| TP | 20 | 8,151 | 407.55 | 148,732 | 7,436.60 |
| RP | 222 | 73,225 | 329.84 | 1,285,168 | 5,789.04 |
| CP | 2,216 | 829,003 | 374.10 | 15,073,031 | 6,810.90 |

**Table 3.3:** Corpus statistics presenting information about the different paper types: Target Papers (TP), Reference Papers (RP) and Citing Papers (CP). It presents the number of papers, sentences and tokens as well was their respective averages.

## 3.5 Annotation Process

In order to perform cross document linking between the Target Paper and the Reference Papers cited in the related work section we relied on experts to do it manually. This process of document linking helps to complete the citation network of the Reference Papers. Three annotators with expertise in Computational Linguistics carried out the annotation process. Annotators were asked to identify which parts of the Reference Papers (one or more sentences) have been cited by the citing Target Papers by means of the citation sentence. We used the open-source, web-based text mining tool WARP-Text (Kovatchev et al., 2018) for the manual annotations process since it allows for annotating relationships between pairs of texts. We customized the tool to perform annotations at a sentence level. We manually annotated the relationship between the Target Papers and the Reference Papers (See the upper half of Figure 3.1). These annotations provide a mapping between the related work section and the texts fragments which are considered semantically close to the citation sentence in the Reference Papers. In order to facilitate the annotation process, we also provided the citation context computed using the state-of-the-art

approach described in (AbuRa'ed et al., 2018). We organized the annotation process in screens showing a citation context and a set of sentences representing the cited Reference Paper to choose from, see Figure 3.2. The annotator then selected which of the Reference Paper sentences best reflect the citation context of the citing Target Paper. See Figure 3.3 for a citation/cited sentence pair.

| **Citing Paper: C08-1031: Mining Opinions in Comparative Sentences** | |
|---|---|
| **Cited Paper: Fiszman-et-al-2007: Interpreting Comparative Constructions in Biomedical Text** | |
| Citation | FISZMAN ET AL (2007) studied the problem of identifying which entity has more of certain features in comparative sentences. |
| **PAGE 4 of 4** | |
| Cited paper | 55: In our sample, expressions interpreted as empty heads include those referring to drug dosage and formulations, such as extended release (the latter often abbreviated as XR).<br>56: Examples of missed interpretations are in sentences (28) and (29), where the empty heads are in bold.<br>57: These mechanisms are being incorporated into the processing for comparative structures.<br>58: 6 CONCLUSION<br>59: We expanded a symbolic semantic interpreter to identify comparative constructions in biomedical text.<br>60: The method relies on underspecified syntactic analysis and domain knowledge from the UMLS.<br>61: We identify two compared terms and scalar comparative structures in MEDLINE citations. |

**Figure 3.2:** Schematic View of the Data during the Annotation Process (on top a citation sentence in a related work section, at the bottom, sentences from the cited paper i.e. reference paper)

The annotation process was straightforward. The web page allowed multiple sentences selection and once a sentence was selected by an annotator it was highlighted. After an annotator selected all the sentences from a Reference Paper that best reflected the citation context, then the annotations were recorded. In cases where a screen presents more than a citation marker in the Citing Paper side, only the target citation would be capitalized to avoid con-

| **Citing Paper: C08-1031: Mining Opinions in Comparative Sentences** | |
|---|---|
| **Cited Paper: Fiszman-et-al-2007: Interpreting Comparative Constructions in Biomedical Text** | |
| Citation | FISZMAN ET AL (2007) studied the problem of identifying which entity has more of certain features in comparative sentences. |
| **PAGE 4 of 4** | |
| Cited paper sentences | We expanded a symbolic semantic interpreter to identify comparative constructions in biomedical text. |

**Figure 3.3:** Sentences Selected by an Annotator Matching a Citation in the Related Work Section

fusion. Finally, the scientific papers names and titles were also visible on the screens. The annotators also had access to all PDF articles which they regularly used to base their decisions. We divided the corpus into 5 batches: the first batch was aimed to get an initial feedback from the annotators. It contained only one Target Paper's related work with the references mentioned in it. The Second batch had 4 Target Papers with their references and the last 3 batches each contained 5 Target Papers with their references. The annotation process was iterative: once a batch was finished we got feedback from the annotators, we computed agreement and we improved annotation recommendations and display accordingly. For example, after the first batch, we realized that furnishing all of the sentences of a Reference Paper at once over one screen was inconvenient for annotation. Therefore, we decided to filter out non-relevant sentences and to divide the rest of the sentences of the Reference Paper over more than one screen where each screen contains a maximum of 15 sentences. We used the work done by (Abura'ed et al., 2018) to filter out unrelated sentences and keep the ones that were most similar to the citation context. All sentences were also available by consulting the original paper in PDF in case no suitable match was found.

### 3.5.1 Inter-Annotator Agreement

We used Cohen's kappa coefficient (Cohen, 1960) in order to measure the inter-annotators agreement for each Target Paper with the Reference Papers mentioned in it. During the annotation process, in all 5 batches there were some conflicts amongst the annotators. We held meetings to address the conflicts in which we presented the annotators with a list of pairs presented by the tool sorted by agreement from worst to best. We improved the annotation

process by going through the list of annotations discussing cases that could lead to any disagreement. One of the annotators was more likely to select sentences which included definitions or background information not reflected in the citations but which she considered important for her understanding of the paper. Situations like these made higher agreement levels difficult to achieve. Hence, the meetings helped to better clarify what information to search for. Table 3.4 reports the pair-wise agreement as well as the average of Cohen's kappa results over the entire corpus. The agreement level $\kappa > 0.5$ indicates moderate agreement between the annotators. The final corpus contains the cited sentences which were selected by majority agreement.

| **Citing Paper: C08-1064: Tera-Scale Translation Models via Pattern Matching** | |
|---|---|
| **Cited Paper: Dyer-et-al.-2008: Fast, Easy, and Cheap: Construction of Statistical Machine Translation Models with MapReduce** | |
| Citation | DYER ET AL. (2008) address this bottleneck with a promising approach based on parallel processing, showing reductions in real time that are linear in the number of CPUs. |
| Cited paper sentence | We have further shown that on a 20-machine cluster of commodity hardware, the Map Reduce implementations have excellent performance and scaling characteristics. |

**Figure 3.4:** An example in which all of the annotators annotated a sentence as being cited

| $A_1$ & $A_2$ | $A_1$ & $A_3$ | $A_2$ & $A_3$ | **Average** |
|---|---|---|---|
| 0.64 | 0.57 | 0.35 | 0.52 |

**Table 3.4:** Pairwise and Average Inter-annotator Agreement

An example in which all the annotators have agreed that a certain sentence has been cited in accordance to the annotation task can be seen at Figure 3.4. Examples in which two annotators agreed on a sentence while the third did not can be found at Figures 3.5, 3.6 and 3.7. Finally, examples of sentences that had no agreement from the annotators can be found at Figures 3.8 and 3.9.

## 3.6 Corpus Enrichment

Each GATE document was annotated using processing resources from the GATE system (Maynard et al., 2002; Cunningham et al., 2002), the SUMMA

| **Citing Paper: C08-1013: ParaMetric: An Automatic Evaluation Metric for Paraphrasing** | |
|---|---|
| **Cited Paper: Lin-and-Pantel-2001: DIRT – Discovery of Inference Rules from Text** | |
| Citation | LIN AND PANTEL (2001) manually judge whether a paraphrase might be used to answer questions from the TREC question-answering track. |
| Cited paper sentence | We then manually inspected the outputs and classified each extracted path as correct or incorrect. |

**Figure 3.5:** An example in which $A_2$ and $A_3$ agreed with each other and annotated a sentence as being cited

| **Citing Paper: E09-1018: EM Works for Pronoun Anaphora Resolution** | |
|---|---|
| **Cited Paper: Cherry-and-Bergsma-2005: An Expectation Maximization Approach to Pronoun Resolution** | |
| Citation | Probably the closest approach to our own is CHERRY AND BERGSMA (2005), which also presents an EM approach to pronoun resolution, and obtains quite successful results. |
| Cited paper sentence | For each pronoun, a list of antecedent candidates derived from the parsed corpus is presented to the Expectation Maximization (EM) learner. |

**Figure 3.6:** An example in which $A_1$ and $A_3$ agreed with each other and annotated a sentence as being cited

library (Saggion, 2008b), and the freely available Dr. Inventor library (DRI Framework) (Ronzano & Saggion, 2015). The tools semantically enrich the corpus by providing rhetorical annotation, causality identification, coreference, and BabelNet synsets (Navigli & Ponzetto, 2010). The SUMMA library was used to produce different normalized term vectors for each document. Vector of terms and BabelNet synsets are created using TF*IDF weighting computed from a corpus of 4K ACL scientific papers. Using 58 gazetteer lists created from the lexicons proposed by Teufel & Moens (2002), we identified scientific concepts and actions useful for text summarization.

The corpus is available for research and development purposes in two versions[21]; one version contains the manual annotations (agreed cited sentences) and the other contains the full machine readable corpus with the automatic

[21] http://taln.upf.edu/sciencecorpus

| **Citing Paper: C08-1013: ParaMetric: An Automatic Evaluation Metric for Paraphrasing** | |
|---|---|
| **Cited Paper: Bannard-and-Callison-Burch-2005: Paraphrasing with Bilingual Parallel Corpora** | |
| Citation | BANNARD AND CALLISON-BURCH (2005) replaced phrases with paraphrases in a number of sentences and asked judges whether the substitutions preserved meaning and remained grammatical. |
| Cited paper sentence | Paraphrases that were judged to preserve both meaning and grammaticality were considered to be correct, and examples which failed on either judgment were considered to be incorrect. |

**Figure 3.7:** An example in which $A_1$ and $A_2$ agreed with each other and annotated a sentence as being cited

| **Citing Paper: C08-1013: ParaMetric: An Automatic Evaluation Metric for Paraphrasing** | |
|---|---|
| **Cited Paper: Lin-and-Pantel-2001: DIRT – Discovery of Inference Rules from Text** | |
| Citation | LIN AND PANTEL (2001) manually judge whether a paraphrase might be used to answer questions from the TREC question-answering track. |
| Cited paper sentence | Inference rules are extremely important in many fields such as natural language processing, information retrieval, and artificial intelligence in general. |

**Figure 3.8:** An example in which none of the annotators annotated a sentence as being cited

analysis just described.

## 3.7 Experiments

In order to identify relevant sentences for writing a related work section, it is first important to know which sentences in a Citing Paper contain relevant information. We have implemented several automatic systems to simulate the annotators' task identifying the problem as one of retrieving sentences which better reflect the citation and its context. For this purpose, we have also enriched the corpus with annotations relevant for scientific text processing in the hope to make it easier for additional related tasks (these annotations are being

| **Citing Paper: C08-1013: ParaMetric: An Automatic Evaluation Metric for Paraphrasing** | |
|---|---|
| **Cited Paper: Barzilay-and-McKeown-2001: Extracting Paraphrases from a Parallel Corpus** | |
| Citation | For example, BARZILAY AND MCKEOWN (2001) evaluated their paraphrases by asking judges whether paraphrases were approximately conceptually equivalent. |
| Cited paper sentence | We present an unsupervised learning algorithm for identification of paraphrases from a corpus of multiple English translations of the same source text. |

**Figure 3.9:** An example in which none of the annotators annotated a sentence as being cited

made available).

### 3.7.1 Automatic Systems

We implemented several automatic systems in which we provide them with a citation context from a related work section in a citing Target Paper and retrieve the reference sentences sorted by the most similar to the citation context.

The systems are as follows:

- Google News: Using a collection of 300 dimensional `word2vec` embeddings trained over a corpus of 100 billion words from Google News[22], this heuristic calculates the centroid of each sentence in the reference and compares it to the centroid of the citing sentence, and returns the most similar ones according to cosine similarity.

- ACL: Similar to the previous case, the heuristic calculates the centroids using 100 dimensional vectors from the ACL Anthology Reference Corpus embeddings (Liu, 2017) trained over a corpus of ACL papers (Bird et al., 2008).

- Google + ACL: The same as before, but using the concatenation of Google News and ACL vectors, creating 400 dimensional vectors. When a word was not present in either of the embeddings collections, it was replaced by a null vector of equivalent size.

[22] https://code.google.com/archive/p/word2vec/

- BabelNet: This heuristic first obtains the BabelNet synsets present in each sentence using the Babelfy API[23]. It then creates an embedding for the sentence by averaging the embeddings of each synset from a BabelNet 300 dimensional embeddings collection trained over a corpus of 300 million words tagged with BabelNet synsets (Mancini et al., 2016) and then returns the sentences sorted by cosine similarity.

- SUMMA normalized vectors: In this case we model each sentence as the vector of normalized TF-IDF values for each of the terms in the sentence, calculating the frequencies in an ACL reference corpus of around 4,000 papers. We compare the citing sentence to all sentences in the reference and return the results sorted by cosine similarity.

- Modified Jaccard: This heuristic uses a metric similar to the Jaccard similarity coefficient for comparing the citing sentence to each sentence of the Reference Paper. This version of the metric (AbuRa'ed et al., 2017) considers the union and intersection of words (like the Jaccard coefficient) but also includes information about the inverted frequency to give more weight to words in the intersection that are less common.

## 3.8 Results

We used Precision at k ($P@k$) (Sujatha & Dhavachelvan, 2011) to evaluate the task of selecting the sentences in each Reference Paper that best reflects the content expressed in the citation context from the related work section of the target scientific paper. See Table 3.7 to see Precision at positions 1 to 5. The results show the automatic systems and how the precision is affected while the position increases. All of the systems have the best precision when one sentences is selected except for the ACL system which have a higher precision when two sentences are selected. Even at higher positions like the fourth and fifth ones it is not a huge difference when it comes to precision. This is a hard task due to the large number of sentences a scientific paper has and the natural difference between citing/cited papers because of the rephrasing characteristics of cited sentences. In this sense results are not surprising, it can be noticed that the BabelNet system is the best one which may indicate that comparing sentences by semantic similarity (instead of lexical) is a good option for achieving good results. The worst results are achieved by systems which use more superficial representations based on words or lemmas. Word embeddings perform better than superficial representations, still worst than semantics

[23] http://babelfy.org/guide

| **Citing Paper: Dolan-et-al.-2004** | |
|---|---|
| **Cited Paper: C08-1013** | |
| True Positive | Two techniques are employed: (1) simple string edit distance, and (2) a heuristic strategy that pairs initial (presumably summary) sentences from different news stories in the same cluster. |
| False Positive | AER measures how accurately an automatic algorithm can align words in corpus of parallel sentence pairs, with a human |

**Table 3.5:** An example of two sentences retrieved by Babelnet one matches the annotators agreement and one does not.

| **Citing Paper: Dolan-et-al.-2004** | |
|---|---|
| **Cited Paper: C08-1013** | |
| True Positive | We evaluate both datasets using a word alignment algorithm and a metric borrowed from machine translation. |
| False Positive | Two techniques are employed: (1) simple string edit distance, and (2) a heuristic strategy that pairs initial (presumably summary) sentences from different news stories in the same cluster. |

**Table 3.6:** An example of two sentences retrieved by ACL one matches the annotators agreement and one does not.

based on lexical resources, and embedding combinations shows positive improvements. Tables 3.5 and 3.6 show two sentences retrieved by Babelnet and ACL systems respectively. In those examples it can be noticed that the Babelnet system agreed with the annotators about a sentence that the ACL system disagreed on. True positive refers to a sentence that has been selected by both the annotators and the system, while a false positive sentence is a sentences that has been retrieved by a system but does not belong to any of the sentences the annotators selected.

## 3.9 Conclusion

In this chapter, we presented a corpus in the field of scientific text mining and summarization to allow the study of automatic related work text generation. The corpus provides related work sections of scientific papers, a manually annotated layer of referenced cited papers, a level of Citing Papers referring to the cited papers in the related work section, and a layer of rich linguistic,

| System | P@1 | P@2 | P@3 | P@4 | P@5 |
|---|---|---|---|---|---|
| ACL | 0.1213 | 0.1416 | 0.1388 | 0.1362 | 0.1369 |
| Babelnet | **0.1934** | **0.1844** | **0.1852** | **0.1789** | **0.1776** |
| Google | 0.1593 | 0.1361 | 0.1422 | 0.1344 | 0.1228 |
| G+ACL | 0.1653 | 0.1428 | 0.1470 | 0.1421 | 0.1321 |
| MJ | 0.0988 | 0.0957 | 0.0887 | 0.0878 | 0.0794 |
| SUMMA | 0.0609 | 0.0590 | 0.0498 | 0.0473 | 0.0478 |

**Table 3.7:** Average Precision for the automatic systems at position 1 to 5

rhetorical, and semantic annotations computed automatically.

We also presented experiments to assess several text representation mechanisms (e.g. lemmas, embeddings, synsets) for the retrieval of sentences likely to be cited by scientific papers comparing system results to the gold standard annotations. The manually annotated corpus with its automatically enriched documents is being made available for the community. We believe this dataset would be useful to the research community to provide means for fair comparisons of various summarization approaches when considering recent work in citation-based summarization. Finally, the corpus is available for research and development purposes in two versions[24]; one version contains the manual annotations (agreed cited sentences) and the other contains the full machine readable corpus with the automatic analysis just described.

[24]http://taln.upf.edu/sciencecorpus

# Chapter 4
# Implicit Citation Detection

An implicit citation is drawn from a scientific paper citing a reference scientific paper indirectly with no explicit citation markers. The identification of implicit citations is important for various scientific text mining tasks such as citation purpose identification, scientific opinion mining, and scientific summarization. Based on an existing annotated dataset of explicit and implicit citation sentences, we present experiments to identify implicit citations in scientific papers. We model the problem as a classification task, evaluating several machine learning algorithms trained on a set of task-motivated features. Our work is compared to the state of the art over the annotated dataset obtaining an improved performance. Additionally, we created a dataset which we make publicly available to validate our approach. The results on the new dataset confirm that our set of features outperforms previously published research.

## 4.1 Introduction

Analysis of citation sentiment would open up many exciting new applications in bibliographic search and in bibliometrics, i.e., the automatic evaluation of the influence and impact of individuals and journals via citations (Athar & Teufel, 2012a). Even though co-citation relations or citation networks interconnect research papers, they are limited in that they do not provide information about why a paper is being cited or what part of the reference paper the citing paper is referring to. Such information could be very important in order to allow fine-grained automatic analysis of scientific works. Finally, citation networks are mostly useful to quantitatively understand the value of a piece of scientific work with no qualitative indications. Identifying which sentences of a reference paper contain the information being referred to by a set of citing papers is a difficult task in part due to the short context provided by the explicit citation. Hence, it becomes necessary to look beyond this explicit

citation for other information in the citing paper that might be relevant. Although the detection of explicit or *formal citations* (for example in the form of author name and paper year, or using a bracketed notation) is a problem that can be resolved with high precision, papers usually contain more information about their references that is not necessarily present in a sentence containing a formal citation. We call these sentences *implicit citations*. There are many examples which highlight the need to identify implicit citations. Analysis of citations sentiment would open up many exciting new applications in bibliographic search and in bibliometrics, i.e., the automatic evaluation of the influence and impact of individuals and journals via citations. Also, it will help in detecting opinions change over the scientific paper. Finally, an extended context would lead into a better matching process between scientific papers, boosting the cross-document linking quality.

This chapter describes experiments on the detection of implicit citations in a paper, i.e. sentences that refer to the work done in another paper but do not contain an explicit citation marker. Consider the fragment shown in Figure 4.1, which is an extract from (He et al., 2008) where they cite, amongst other papers, the Pyramid method defined in (Nenkova et al., 2007).

(217) The official evaluation comprises three methods under different assumption: ROUGE [4], PYRAMID [5], and BE [3].
...
(247) In essence, Pyramid evaluation method adopts the voting idea to give the different weight for different importance Summary Content Unit (SUC).
(248) For our approach, we essentially find the stationary distribution of random walk in evolutionary manifold-ranking (...).
(249) This idea is similar to the evaluation idea of Pyramid method and more importance is that we catched the evolutionary characteristic (...).
(250) Whereas we don't do any processing of coherence and got the less linguistic quality.
...
(256) We think that ROUGE and BE are suitable to evaluate the content selection of generative summary, (...) and PYRAMID is suitable to evaluate the content selection of extractive summary (...).

**Figure 4.1:** Extract from (He et al., 2008), indicating the number of sentence in the document between parenthesis.

The explicit citation is in sentence 217: The marker "[5]" refers to (Nenkova et al., 2007). Several paragraphs later, from sentence 247 onwards, He et al. (2008) describe properties of the Pyramid method and compare it to what they

did and also to other methods. In this example, we could consider sentence 217 as a formal citation, sentences 247, 249 and 256 as implicit citations, and sentences 248 and 250 are not considered citations. Note that in these sentences, they use the name of the method defined in the reference paper instead of the author, but nonetheless they are talking about the same paper.

Authors can use several techniques for implicitly referring to a paper (Athar & Teufel, 2012b), for example: using only the name of the main author, using pronouns that could refer to the mentioned work, or using keywords that refer to a distinguishing topic in the paper (in the previous example, the Pyramid method). As well, the implicit citations can be found far from the explicit citations. The problem can be modeled as a sentence classification task: considering one sentence of the citing paper at a time, try to identify if the sentence is talking about the work done in the target reference paper, but does not contain an explicit citation to it. This task has attracted considerable attention because of its applicability in several problems in scientific literature analysis. Our approach, which is based on training a classifier with task-motivated features, improves over the state of the art in a publicly available dataset.

One of the early attempts at identifying sentences that were related to a citation but did not explicitly contain the citation marker was done by (Nanba & Okumura, 1999). In their work, they define a "reference area" which begins with the sentence that contains a citation marker and contains the following sentences that have a connection with the same subject. They use a set of cue words for identifying the sentences that belong to the reference area and use this information to build a multi-paper summarization system. (Kaplan et al., 2009) defines citation sites as the portions of text around a citation anchor in which the citation is discussed. These citation sites might be non-contiguous, but they limit the maximum distance from the anchor and they train a coreference resolution model to identify this non-contiguous fragments. (Qazvinian & Radev, 2010b) try to identify what they call "context sentences", which are sentences that contain an implicit citation. After analyzing some cases they report that those context sentences tend to occur in a small neighborhood of the explicit citation. They train a Markov Random Field model that tries to identify these context sentences, and use this information to build a summarization system by extracting keyphrases (Qazvinian et al., 2010). More recently, in (Kaplan et al., 2016) they define a similar problem of citation block determination. They train SVM and CRFs models including features such as location, topic modeling, discourse and coreference to determine if a sentence belongs to a citation block. However, they do not consider the implicit citations that might be non-contiguous to the citing sentence.

Our work follows closely the research of (Athar & Teufel, 2012c), which uses

the implicit citations in order to enrich a citation sentiment analysis system. In order to do this, they build a corpus of papers annotated with formal (explicit) citations and informal (implicit) citations, all of which are categorized as positive, negative or objective. They train a SVM model with a set of features that tries to capture relevant information for detecting implicit citations, even if they are non-contiguous, for example detecting other ways of referring to the work of an author inside a document that do not imply using the name of the author. Using this information they improve the performance of a citation sentiment classifier.

Incorporating implicit citations together with formal citations can be applied to several tasks in the context of scientific literature analysis. For example, one problem that has been studied is the automatic creation of scientific paper summaries. In addition, the task of detection of function and polarity of a citation (Athar, 2011), (Athar & Teufel, 2012c), (Li et al., 2013), (Abu-Jbara et al., 2013) can be improved using implicit citations.

The contributions of this chapter are the following:

- A novel set of features for implicit citation identification;
- A set of experiments demonstrating the improved performance of the taken approach;
- A novel data-set for the implicit citation identification task;
- The software and data developed are being made available to the research community[25].

## 4.2 Citation Context Corpus

The Citation Context Corpus (Athar & Teufel, 2012b) was created to address the problem of identifying implicit citations. As an example of the information contained in the corpus, Banerjee & Lavie (2005)'s work has been cited by Liu & Gildea (2006), the citation process involved a formal citation followed by several informal mentions, See Table 4.1. While the first sentence cites Banerjee & Lavie (2005)'s paper explicitly by using the name of the primary author along with the year of publication of the paper, the remaining sentences mentioning the same paper appear after a gap and contain an indirect and implicit reference to that paper. These mentions occur two sentences after the formal citation in the form of anaphoric *it* and the lexical hook METEOR.

Most current techniques, with the exception of Qazvinian and Radev (2010), are not able to detect linguistic mentions of citations in such forms. Ignor-

[25] https://github.com/AhmedAbuRaed/CitationContextExtension

| |
|---|
| ... In order to improve sentence-level evaluation performance,<br>several metrics have been proposed, including ROUGE-W,<br>ROUGE-S (Lin and Och, 2004) and METEOR (Banerjee and Lavie, 2005).<br>...<br>METEOR is essentially a unigram based metric,<br>which prefers the monotonic word alignment between<br>MT output and the references by penalizing crossing word alignments.<br>There are two problems with METEOR.<br>...<br>ROUGE and METEOR both use WordNet and Porter Stemmer to increase<br>the chance of the MT output words matching the reference words ... |

**Table 4.1:** Example of the use of anaphora i.e. a formal citation to a scientific paper followed by few informal citations.

ing such mentions and examining only the sentences containing an explicit citation results in loss of information about the cited paper. While this phenomenon is problematic for applications like scientific summarisation (Abu-Jbara and Radev, 2011), it has a particular relevance for citation sentiment detection (Athar, 2011). The Citation Context Corpus Athar & Teufel (2012b) consists of the full text of 852 papers (i.e. 203,803 sentences) which all cite 20 target papers from the ACL Anthology Network (AAN) corpus (Bird et al., 2008). This data is presented as a set of HTML files where each file contains all papers in the AAN which cite a specific target paper. The file contains a table where each row corresponds to a citing paper, and each cell in that row represents one sentence in the citing paper. Each sentence is marked as a formal citation, an informal citation, or no citation at all, using a color code. Figure 4.2 shows an example of one target paper[26] (HTML file[27]). The colors represent if the citations are positive, negative or netural, and the shades represent formal or informal citations.

For example the target paper is being cited by a paper [28] using a formal neutral citation:“3 The statistical model We use the Xerox part-of-speech tagger **(Cutting et al. , 1992)**, a statistical tagger made at the Xerox Palo Alto Research Center.”, followed directly by a formal positive citation: “3.1 Training The Xerox tagger is claimed **(Cutting el al. , 1992)** to be adaptable and eas-

[26]Cutting, Douglass, et al. "A practical part-of-speech tagger." Third Conference on Applied Natural Language Processing. 1992.

[27]https://cl.awaisathar.com/citation-context-corpus/A92-1018.html

[28]Chanod, Jean-Pierre, and Pasi Tapanainen. "Tagging French: comparing a statistical and a constraint-based method." Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics. Morgan Kaufmann Publishers Inc., 1995.

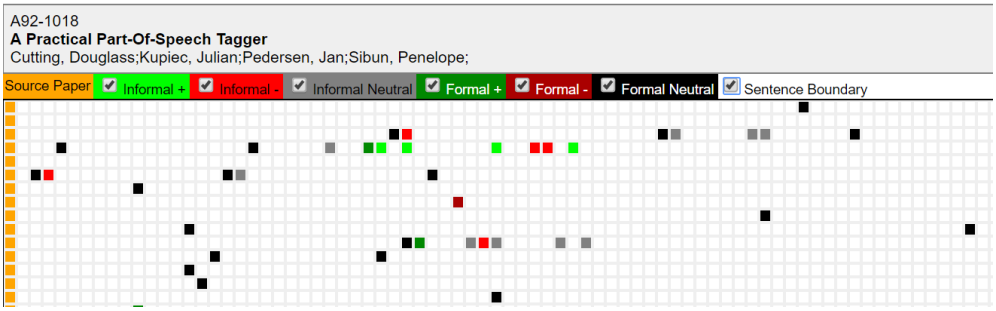


**Figure 4.2:** An example of a target paper's HTML file from Athar's Corpus

ily trained; only a lexicon and suitable amount of untagged text is required.". Afterwards, the author of the citing paper implicitly cites the target paper with three consecutive sentences as a neutral, negative and another neutral informal citations respectively as follows: "We ran the **tagger** on another text and counted the errors.", "The result was not good; 13 % of the words were **tagged** incorrectly." and "The **tagger** does not require a tagged corpus for training, but two types of biases can be set to tell the **tagger** what is correct and what is not: symbol biases and transition biases.". An example of a negative formal citation can be seen by another citing paper[29] as follows: "Brill's results demonstrate that this approach can outperform the Hidden Markov Model approaches that are frequently used for part-of-speech tagging (Jelinek, 1985; Church, 1988; DeRose, 1988; **Cutting et al. , 1992**; Weischedel et al. , 1993), as well as showing promise for other applications.". Finally, an example of an informal positive citation is: "An important aspect of this **tagger** is that it will give good accuracy with a minimal amount of manually **tagged** training data." which can be seen by a third citing paper[30].

## 4.3 Experiments

We treated the problem as a binary classification problem and we used a supervised machine learning approach to predict implicit citations. We extended the approach used by Athar & Teufel (2012b) to include task-motivated features. As the software produced by Athar & Teufel (2012b) is not available, we re-

[29]Ramshaw, Lance A., and Mitchell P. Marcus. "Exploring the statistical derivation of transformational rule sequences for part-of-speech tagging." arXiv preprint cmp-lg/9406011 (1994).

[30]Elworthy, David. "Does Baum-Welch re-estimation help taggers?." Proceedings of the fourth conference on Applied natural language processing. Association for Computational Linguistics, 1994.

implemented all the features defined in that paper and attempted to replicate the results obtained by the authors using a Support Vector Machines (SVM) classifier to compare with our approach.

### 4.3.1 Athar & Teufel's features

The original classifier described in (Athar & Teufel, 2012b) is a SVM trained using the following set of binary features:

- **Formal Citation:** Two features indicating if the citation (for example as author name followed by year) appears in the previous or in the current sentence. The feature for the current sentence is meant to help the classifier discard sentences containing formal citations, as they are not the target of their work. The feature for the previous sentence, however, could help detect sentences immediately after a formal citation that might still be talking about the same subject.
- **Author name:** A feature indicating if the author name is in the sentence, but not with the year as would happen in a formal citation. It has been shown that sometimes a paper that was already formally cited can be recalled by using only the author name.
- **Other citations:** This feature indicates if the sentence contains a citation different from the one the classifier is trying to detect.
- **Determiner and *work noun*:** Work nouns are defined in (Siddharthan & Teufel, 2007) as nouns used to indicate other people's work. This feature would capture expressions such as "the study" or "their result".
- **Third person pronoun:** This feature indicates a sentence that starts with a third person pronoun, in order to capture sentences like "They show that...".
- **Connector:** Used to mark if a sentence starts with a connector, from a list of 23 connectors such as "however" or "moreover".
- **Subsection heading:** Three features indicating if the previous, current, or next sentence starts with a subsection heading. These features could help identify a topic shift in the analyzed sentence.
- **Acronyms:** Indicates if a sentence contains an acronym mentioned near a formal citation. In the example above, "PYRAMID" is an acronym used in place of a citation.

- **Lexical substitutes for the citation (Lexical hooks):** For this feature, it is necessary to analyze all citing papers besides the one that is being classified. A lexical substitute (also referred to as Lexical hook) is defined as the most frequently capitalized phrase found around a formal citation to the reference paper. The intention is to capture other common ways of referring to a paper that do not imply the name of the author or an acronym, for example the phrase "Pyramid method" in the example above.

- ***N*-gram features:** They consider features for *n*-grams of length 1 to 3 in the sentence. Besides using these features in the final classifier, they train a classifier that only uses *n*-grams as a baseline for comparison. In our case we used the SUMMA library (Saggion, 2008c) for calculating the *n*-gram features.

According to (Athar, 2014), the most relevant features were lexical hooks, acronyms, and whether or not a formal citation is contained in the previous sentence.

### 4.3.2 Our features

After an empirical examination of the corpus and the task at hand, we defined content-based, contextual features that could incorporate novel information to the classifier:

- **Word Embeddings Cosine Similarity:** The more similar a text is to another, the more likely it is that it might be referring to it. We utilized a set of pre-trained word2vec models with 300 dimensions representing each sentence in the vector space. For this set of features we calculated the centroid of each sentence and the centroid of the reference paper abstract, and then measured the cosine similarity of these two vectors. We generated a set of three features corresponding to using three different embeddings collections: Google News embeddings [31], the ACL Anthology Reference Corpus embeddings (Liu, 2017) (trained over a corpus of ACL papers (Bird et al., 2008)), and the BabelNet embeddings (Mancini et al., 2016) (trained over a corpus of documents disambiguated using BabelNet (Navigli & Ponzetto, 2012a) synsets).

- **Context Vectors Cosine Similarity:** Using the SUMMA library (Saggion, 2008c) we calculated the TF*IDF vectors of each sentence and

[31] https://code.google.com/archive/p/word2vec/

the reference paper abstract, and we used the cosine similarity of these pairs of vectors as features. We have two features using the TF*IDF measure for lemmas and for BabelNet synsets which are extracted using BabelFy (Moro et al., 2014). The IDF tables for lemmas and synsets were computed using a subset of around four thousand ACL anthology papers.

- **Scientific Gazetteer:** Teufel's (2000) action and concept Lexicons were used to create gazetteers lists to identify scientific references (e.g. *research*: 'analyze', 'check' and 'gather'; *problem*: 'violate', 'spoil' and 'mistake', and *solution*: 'fix', 'cure' and 'accomplish'). We created two features to count how many words in the sentence belong to the "action" or the "concept" category of the gazeteers.

- **Co-reference Chains:** The Dr. Inventor (DRI) Text Mining Framework (Ronzano & Saggion, 2015) provides co-reference resolution over the scientific papers. A feature that detects cases in which a reference to an entity in a sentence containing an explicit formal citation through a co-reference chain was made.

- **Rhetorical Category:** DRI also predicts the probability of a sentence being in one of five possible rhetorical categories (i.e. Approach, Background, Challenge, Outcome and Future work). We added a feature which indicates the index of the highest probability rhetorical category which the target sentence represents. We believe that indicating the sentence rhetorical category could be informative for our classification task.

- **Cause and Effect:** DRI annotates causal relations in scientific papers. We used a feature to detect the existence of any causality relation in the sentence.

- **Citations:** The more formal citations a sentence could have the less likely it will contain an implicit citation. This feature will simply count the number of formal citations the sentence has for any related work.

- **Distance to closest formal citation** The closer a sentence is to a formal citation to the reference paper the more likely it is an implicit citation. We generate one feature to calculate the distance between the sentence and the closest formal citation to the reference paper.

- **Title tokens:** Implicit citations could contain tokens from the reference paper title. One feature has been calculated in which the value represents the number of tokens in the reference paper title appearing in the sentence.

- **POS *N*-grams:** We also added Features for part of speech *n*-grams of length 1 to 3 in the sentence.

### 4.3.3 Features analysis

In order to understand the discriminatory power of our features, we ran the information gain feature selection algorithm with the attribute ranking search method by utilizing WEKA on the set of features excluding *n*-grams over the training dataset. Such a test provides a better insight of which features are more important than others. Table 4.2 shows the top 12 features selected by the algorithm.

| | |
|---|---|
| Distance to closest formal citation* | 0.0062925 |
| Title Tokens* | 0.0054377 |
| Context Vectors cosine similarity* | 0.0049679 |
| Lexical Hooks | 0.0041353 |
| Acronyms | 0.0040977 |
| Babelnet Context Vectors cosine similarity* | 0.0040918 |
| Previous Formal Citation | 0.003768 |
| ACL Word2Vec cosine similarity* | 0.002034 |
| Google News Word2Vec cosine similarity* | 0.001385 |
| Author name | 0.000991 |
| Co-reference Chains* | 0.0007706 |
| BabelNet Word2Vec cosine similarity* | 0.000753 |

**Table 4.2:** Top 12 features ranked by the information gain algorithm. The features marked with * are new features.

What can be noticed from table 4.2 is that the majority of the top features (8 - marked as * on the table - out of 12) of the training dataset are from the newly generated features used by our system. We have evaluated several classifiers using AUTO-WEKA (Thornton et al., 2013): which simultaneously selects a learning algorithm and sets its hyper parameters. We configured AUTO-WEKA to run experiments for 7 hours in which 13 configurations were performed using 10-fold cross validation using several classifier including: Bayes net, Random Tree, SMO and Random Forest. Finally, we tested both (Athar & Teufel, 2012b) and our approach against a new test set which we annotated manually.

## 4.4 Results and Discussion

Table 4.3 shows the results of our replication of Athar's experiments (the *n*-gram baseline and the features) as well as the results using our set of features, using 10-fold cross validation over the training data. For our method we have applied a set of machine learning algorithms to check which one yields the best results using 10-fold cross validation and unlike Athar's approach which used SVM, our best model was using Random Forest algorithm.

| Experiment | Precision | Recall | F-Measure |
|---|---|---|---|
| Athar's baseline | 0.643 | 0.293 | 0.403 |
| Athar's features | 0.609 | 0.362 | 0.454 |
| Our novel features | **0.684** | **0.370** | **0.480** |

**Table 4.3:** Cross validation results

One thing we can see from the results is that the experiments using our implementation of Athar's features did not yield the same performance as reported in (Athar & Teufel, 2012b). They reported an F-measure of 0.513 for the implicit citation class, but in our case we got 0.454 for the same experiment. This is expected, as we did not use the same tools they had used to compute features, so we could not replicate exactly the same experimental setting. For both experiments, however, the classifiers beat the n-gram baseline.

### 4.4.1 Test data

In order to further validate our approach and try to compare it to the previous one, we annotated a small set of test documents. We collected five target papers from ACM Transactions on Computational Logic Journal, ACL and NAACL conferences by skimming over multiple scientific papers and choosing the ones with multiple implicit citations while skipping the ones without. For each paper we collected the papers citing it and had identified the explicit and implicit citations following the same approach as (Athar & Teufel, 2012b), but without considering the sentiment polarity of citations. The only annotations used are Formal Citation or Implicit Citation. Table 4.4 shows the composition of this small test corpus. To better understand the kind of differences between results, we tried both models over the test data. The results are shown in table 4.5.

Note that the performance of both classifiers dramatically dropped on the test data. The worst performance was for the *Blunsom2008* cluster, where both classifiers predicted no implicit citations. The best performance was for the

| Cluster | Papers | Sents | Formal | Implicit |
|---|---|---|---|---|
| Nenkova2007 | 5 | 1550 | 9 | 21 |
| Kaplan2004 | 6 | 1896 | 24 | 11 |
| Blunsom2008 | 7 | 2750 | 19 | 18 |
| BunescuPasca2006 | 12 | 5531 | 44 | 76 |
| CardieWagstaff1999 | 9 | 3895 | 20 | 24 |
| Total | 39 | 15622 | 116 | 150 |

**Table 4.4:** Composition of the test corpus

| Experiment | Precision | Recall | F-Measure |
|---|---|---|---|
| Athar's system | 0.0500 | 0.0067 | 0.0118 |
| Our system | 0.1613 | 0.0333 | 0.0552 |

**Table 4.5:** Results over test data

*Nenkova2007* cluster, where both classifiers had at least one hit. Although the classifier with our features performs a little better than the previous one, both results were very poor. One possible explanation for this is that the features used for both classifiers were too specific and could not generalize to these new examples. Another possibility is that the test clusters themselves were very hard to classify.

## 4.5 Summary and Conclusions

We presented the results of our experiments on the detection of implicit citations/references to a reseach paper, with the aim of using this method for improving the performance of a reference scope detection system. We calculated the features used in a previous work and created a new set of features that were found relevant for the classifier. We first trained an implicit citations classifier as specified in (Athar & Teufel, 2012b), and then built a new classifier using all the features. The new classifier performs better than the previous published work when evaluated with a cross-validation methodology. In both cases the results were lower than the ones reported in (Athar & Teufel, 2012b), but we consider this could have happen because our experimental setting is different. So using our features with the same experimental setting as Athar might lead to even better results. In order to further analyze the results of the classifiers, we annotated a small test set of scientific documents. On the newly created test set, the performance of both classifiers drop, still our new classifier shows better results. Finally, the software and data developed are being made avail-

able to the research community[32].

[32]https://github.com/AhmedAbuRaed/SPSeq2Seq

# Chapter 5

# Scientific Document Summarization Using Citation Networks

In this chapter, we present several systems developed to participate in the Computational Linguistics Scientific Document Summarization Shared challenge which addresses the problem of summarizing a scientific paper taking advantage of its citation network (i.e., the papers that cite the given paper). Given a cluster of scientific documents where one is a Reference Paper (RP) and the remaining documents are papers citing the reference, two tasks are proposed: (i) to identify which sentences in the reference paper are being cited and why they are cited, and (ii) to produce a citation-based summary of the reference paper using the information in the cluster.

## 5.1 Introduction

The interest in the area of citation-based scientific text summarization has motivated the development of a series of evaluation exercises in scientific summarization in the Computational Linguistics (CL) domain known as the Computational Linguistics Scientific Document Summarization Shared Task which started in 2014 as a pilot (Jaidka et al., 2014a) and which is now a well developed challenge in its fourth year (Jaidka et al., 2016, 2017a,e, 2019; Chandrasekaran et al., 2019).

The CL-SciSumm shared task aims to encourage research towards scientific paper summarization, which considers the set of citation sentences (i.e., "citances")

that reference a specific paper as a (community created) summary of a topic or paper (Qazvinian & Radev, 2008b). Citances for a Reference Paper are considered a summary of its key points and also its key contributions within an academic community (Nakov et al., 2004a).

The CL-SciSumm explores summarization of scientific research, for the Computational Linguistics research domain. It encourages the integration of new types of information in automatic scientific paper summarization, such as the use of citation networks to emphasis the use of citations written in other papers by other scholars in order to refer to the paper (Jaidka et al., 2016).

In this challenge, given a cluster of $n$ documents where one is a Reference Paper (RP) and the $n-1$ remaining documents are papers (i.e., Citing Papers (CPs)) citing the reference paper, participants of the challenge have to develop automatic procedures to simulate the following tasks:

- **Task 1A**: For each citance in the citing papers (i.e., text spans containing a citation), identify the cited spans of text in the reference paper that most accurately reflect the citance.
- **Task 1B**: For each cited text span, identify to which **discourse facet** it belongs to, among: *Aim*, *Hypothesis*, *Implication*, *Results*, or *Method*.
- **Task 2**: Finally, an optional task consists on generating a structured summary of the reference paper with up to 250 words from the cited text spans.

Additionally, the dataset provides three types of summaries for each Reference Paper:

- the abstract, written by the authors of the research paper.
- the community summary, collated from the majority of the reference spans of its citances.
- a human-written summary, written by the annotators of the CL-SciSumm annotation effort.

Participants were required to submit their system outputs from the test set to the CL-SciSumm organizers. The submissions from all the participants were evaluated automatically, for Task 1A they calculated the number of sentences from the systems output that overlap with the sentences in the human annotated reference text span. This was used to calculate precision, recall and F1 score for each system. As for Task 1B is a multi-label classification, this task

was also scored by the same metrics of precision, recall and F1 score. Finally, for the summarization task (Task 2), the ROUGE package (Lin, 2004) was used to compare the three types of gold summaries (i.e. Abstract, Comunity and Human summaries) against the system generated summaries.

The contributions of this chapter are the following:

- identifying which sentences in a Reference Paper has been cited by a citation context;
- multiple supervised and unsupervised methods have been implemented to participate in the CL-SciSumm shared task;
- The software is made available for the research community [33,34];

## 5.2 CL-SciSumm Corpus

The CL-SciSumm challenge organizers have provided training data structured in clusters. A cluster is a set of reference and citing papers together with manual annotations indicating for each citance to the reference paper, the facet of this citance and the text span(s) in the reference paper that best represents the citance.

For each cluster there are three manually created summaries of the reference paper: the author abstract, a community-based abstract created using citation sentences, and a human abstract created based on information from reference paper and citation sentences.

An example of a manual annotation provided by the organizers for Task 1 can be seen at Figure 5.1 and a visual representation of that annotation can be seen at Figure 5.2. This example shows a citing paper [35] in one of the clusters citing a reference paper [36] from the *results* section (Results Facet) using the citation marker "Sproat et al., 1996". Finally, an example of a gold human summary provided for the same cluster (Reference Paper) can be seen at Figure 5.3.

In the first year of the challenge (2016) the organizers have provided 20 clusters:

[33]https://github.com/AhmedAbuRaed/CLSciSumm2018

[34]https://github.com/AhmedAbuRaed/CL-SciSumm2017

[35]Lee, John. "A classical Chinese corpus with nested part-of-speech tags." Proceedings of the 6th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities. Association for Computational Linguistics, 2012.

[36]Sproat, Richard, et al. "A stochastic finite-state word-segmentation algorithm for Chinese." Computational linguistics 22.3 (1996): 377-404.

```
Citance Number: 60 | Reference Article:  J96-3004.xml | Citing Article:  W12-1011.xml |
Citation Marker Offset:  ['41'] | Citation Marker:  Sproat et al., 1996 | Citation Offset:
['41'] | Citation Text:  <S sid ="41" ssid = "5">Indeed, even native speakers can agree on
word boundaries in modern Chinese only about 76% of the time (Sproat et al., 1996).</S> |
Reference Offset:  ['325'] | Reference Text:  <S sid ="325" ssid = "34">The average agreement
among the human judges is .76, and the average agreement between ST and the humans is .75, or
about 99% of the interhuman agreement</S> | Discourse Facet:  Results_Citation | Annotator:
Ankita Patel |
```

**Figure 5.1:** An example of a manual annotation provided by the CL-SciSumm organizers for Task 1

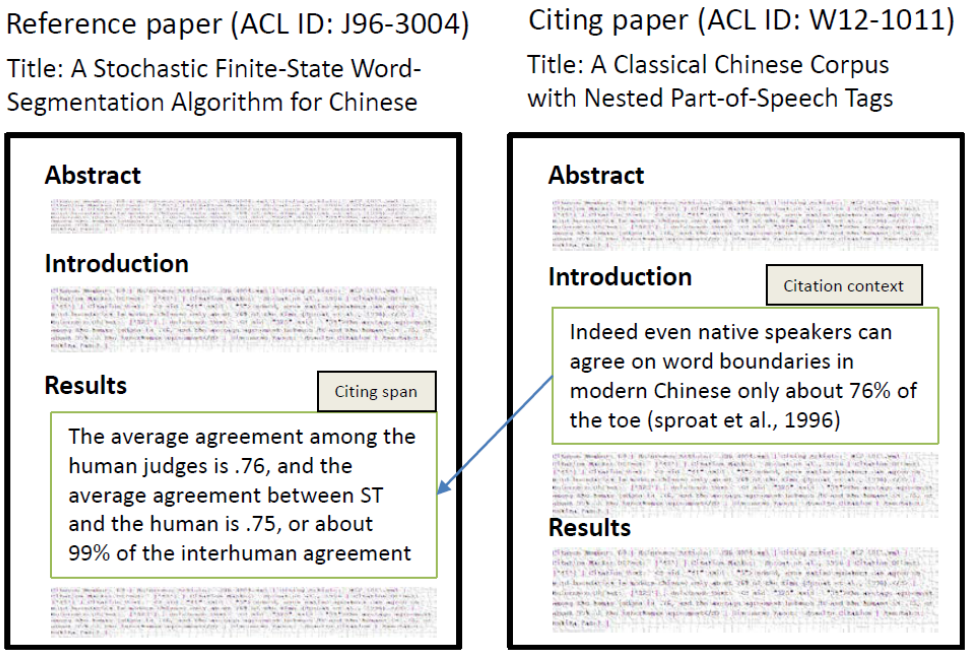


**Figure 5.2:** A visual representation of a manual annotation provided by the CL-SciSumm organizers for Task 1

In this paper the authors present a stochastic finite-state model for segmenting Chinese text into words.The model incorporates various recent techniques for incorporating and manipulating linguistic knowledge using finite-state transducers. It also incorporates the Good-Turing methodin estimating the likelihoods of previously unseen constructions, including morphological derivatives and personal names. they evaluate various specific aspects of the segmentation, as well as the overall segmentation performance. The evaluation compares the performance of the system with that of several human judges and inter-human agreement on a single correct way to segment a text.they showed that the average agreement among the human judges is .76, and the average agreement between ST(system) and the humans is .75, or about 99% of the interhuman agreement.This architecture provides a uniform framework in which it is easy to incorporate not only listed dictionary entries but also morphological derivatives, and models for personal names and foreign names in transliteration. Other kinds of productive word classes, such as company names, abbreviations,and place names can easily be handled given appropriate models.

**Figure 5.3:** An example of a gold human summary provided by the CL-SciSumm organizers for Task 2

10 for training and 10 for testing. Afterwards, throughout the years of the challenge they increased the size of the data mostly by adding the test data from a previous year into the following year's training data and adding a new test data after manually annotating it. We report such additions in each participation of our yearly additions at sections 5.4 and 5.5.

### 5.2.1 CL-SciSumm Corpus Processing

In order to properly analyze the CL-SciSumm corpus, we transformed the clusters into GATE (Maynard et al., 2002) documents. The files corresponding to reference papers were enriched with annotations covering the text spans being cited (with the information corresponding to citances). Conversely, in each Citing Paper annotations were added for the provided citances (with the information corresponding to the cited text spans). The annotations in the citing and reference papers are linked by means of a unique identifier (formed by the concatenation of citance number, reference paper id, Citing Paper id, and annotator).

Such annotations are helpful in order to retrieve the necessary information from the documents. In this way, in each Citing Paper we are able to identify for each sentence that belongs to a citance, the sentences of the corresponding reference paper that most accurately reflect the citance. Thanks to this information, we can build pairs of matching sentences (Citing Paper Sentence, Ref-

erence Paper Sentence) and associate to each pair the facet that each annotator considers that the citation is referring to. An example of the representation can be seen in Figure 5.4 where a reference paper (on the left side of the Figure) annotated with information from the citing papers (on the right side of the figure).

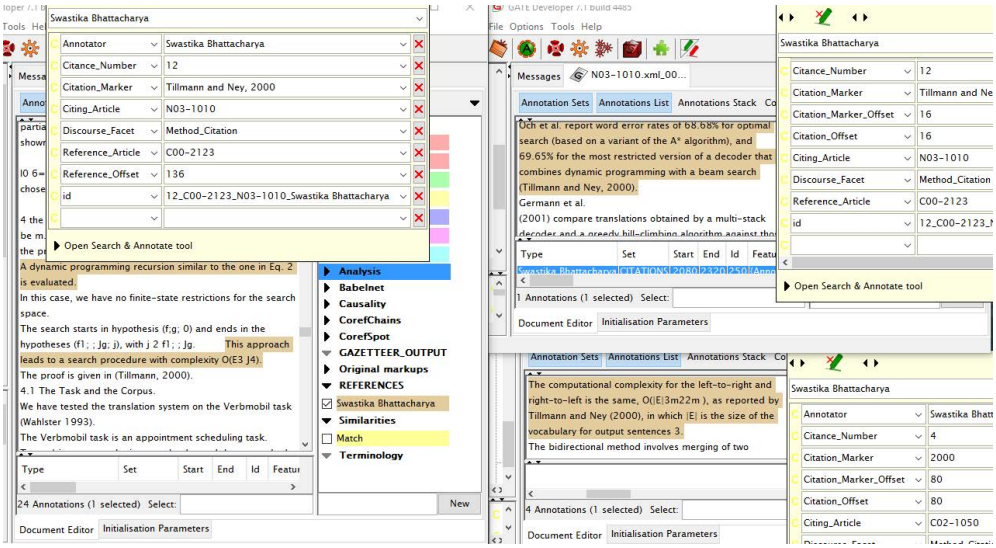


**Figure 5.4:** GATE GUI representation of the annotations of a reference paper (left side) and two citing papers from the same cluster (right side).

Each document was annotated using processing resources from the GATE system (Maynard et al., 2002) and the SUMMA library (Saggion, 2008a). Additionally, in order to further enrich the documents, some components from the freely-available Dr. Inventor (DRI) Framework Library (Ronzano & Saggion, 2015) were used.

The tokenizer, sentence splitter, Part of Speech tagger, and lemmatizer available in GATE's ANNIE[37] component were used to initially process the documents. GATE was also used to annotate each document with gazetteers, Teufel's (Teufel, 2000) action and concept lexicons were used to create gazetteers lists to identify in text scientific concepts (e.g. *research*: 'analyze', 'check' and 'gather'; *problem*: 'violate', 'spoil' and 'mistake', and *solution*: 'fix', 'cure' and 'accomplish').

The SUMMA library (Saggion, 2008a) was used to produce term vectors, normalized term vectors, BabelNet (Navigli & Ponzetto, 2012a) synset ID vec-

[37] https://gate.ac.uk/ie/annie.html

tors, normalized babelnet synset ID vectors, terms *n*-grams (up to three) and Part of Speech *n*-grams (up to three) for each document.

The Dr. Inventor's library (Ronzano & Saggion, 2015) for analyzing scientific documents was additionally applied to each document to generate rich semantic information such as citation marker, BabelNet concepts (Navigli & Ponzetto, 2012b), causality markers, co-reference chains, and rhetorical sentence classification. The library classifies each sentence of a paper based on a rhetorical category of scientific discourse among: *Approach*, *Background*, *Challenge*, *Outcome* and *FutureWork*. In other words, it predicts the probability of the sentence of belonging to one of the five rhetorical categories provided. See (Fisas Elizalde et al., 2016) for more details about the corpus used for training the classifier.

## 5.3 Participation in the $1^{rst}$ CL-SciSumm Shared Task (2016)

The year 2016 was the first year of the CL-SciSumm shared task series and our first participation (Saggion et al., 2016a) in the shared task as well. We already had an annotated corpus from the organizers so we decided to approach Task 1 as a classification problem trying to predict whether a sentence in the RP has been cited or not. We modeled pairs of reference and citance sentences as a feature vector. Then, we used such pair representation to enable the supervised training of distinct binary classification algorithms tailored to determine whether they are a match.

As for Task 2 we decided to use the freely available text summarization library SUMMA (Saggion, 2008d, 2014) to generate a series of sentence relevance features which are used to train a linear regression model following the methodology that was already used in (Brügmann et al., 2015).

### 5.3.1 Task1: Identifying Cited Sentences and Their Facets

In order to identify RP text spans for each citance (Task 1A), we modeled pairs of reference and citance sentences as a feature vector. Then, we used such pair representation to enable the training of distinct binary classification algorithms tailored to determine whether they are a match.

On the other hand we used the same representation of pairs of sentences for identifying to what facet of the reference paper a cited text span belongs to (Task 1B): we classified each pair of sentences in one out of 5 predefined facets Aim, Hypothesis, Implication, Results or Method.

To this end, we relied on the WEKA machine learning framework (Witten et al., 2011). We evaluated the performance of six classification algorithms: SMO, Naive Bayes, J48, Lazy IBK, Decision table and Random Forest for both tasks. We performed 10-fold cross validation experiments with the training data in order to decide which algorithm to use during testing.

In the remainder of this Section, we describe the set of sentence pair features we used, and their relevance with respect to the characterization of sentences similarity. When presenting the features, we group subsets of related features in the same subsection (Position features, Similarity features, etc.).

### Position Features

We exploited the following set of position related features for both Task 1A (text spans for citance sentence) and 1B (the facet such text span belongs to):

- *Sentence position* (**sentence_position**): the position of the sentence in a reference paper. We normalized this feature ($1/position$).
- *Sentence section position* (**sentence_section_position**): the position of the sentence in its section in the reference paper. We normalized this feature ($1/positioninsection$).
- *Facet position* (**facet_aim**, **facet_hypothesis**, **facet_implication**, **facet_method** and **facet_result**): five features were generated to indicate to which facet a cited text span belongs. Binary values were calculated by analyzing the reference paper sentence's section title and looking for any words which could indicate the feature facet: aim, hypothesis, implication, method or result. The value of the feature is 1 for section titles containing a word that indicate such facet and 0 otherwise.

### WordNet Semantic Similarity Measures features

The following set of Semantic Similarity features were exploited for task 1A (text spans for citance sentence) with the exception of the cosine similarity which was used for both task 1A and task 1B. We used WordNet Similarity for Java (WS4J) library which includes several semantic relatedness algorithms that rely on WordNet 3.0. Given a pair of sentences (reference and citance), we retrieve all the synsets associated to nouns and verbs in each one of them. Then, by considering all the pairs of synsets belonging to different sentences,

we compute similarity values between citance sentence and reference sentence as follows[38]:

- *Path similarity* (Hirst & St-Onge, 1998) (**path_similarity**): The shorter the path length between two words/senses in WordNet, the more similar they are.
- *JCN similarity* (Jiang & Conrath, 1997) (**jiangconrath_similarity**): the conditional probability of encountering an instance of a child-synset given an instance of a parent synset.
- *LCH similarity* (Leacock & Chodorow, 1998) (**lch_similarity**): the length of the shortest path between two synsets for their measure of similarity.
- *LESK similarity* (Banerjee & Pedersen, 2002) (**lesk_similarity**): Similarity of two concepts is defined as a function of the overlap between the corresponding definitions (i.e., their WordNet glosses).
- *LIN similarity* (Lin, 1998) (**lin_similarity**): The Similarity between two word senses is measured by the ratio between the amount of information needed to state the commonality of these two senses and the information needed to fully describe them.
- *RESNIK similarity* (Resnik, 1995) (**resnik_similarity**): The probability of encountering an instance of a concept in a large corpus.
- *WUP similarity* (Wu & Palmer, 1994) (**wup_similarity**): The depths of the two synsets in the WordNet taxonomies, along with the depth of the lowest common subsumer.
- *Cosine similarity* (**cosine_similarity**): The cosine similarity between the normalized vectors of the two sentences in the instance pair (this computation is different from the other similarity features).

**Rhetorical Category Probability Features**

We computed a set of features based on the rhetorical category probability for both task 1A (text spans for citance sentence) and 1B (the facet which the text span belongs to):

[38] We calculated similarity values between each token in the citance sentence and each and every token in the reference sentence. Finally averaging all the similaries for the given sentence pair.

| Class | Precision | Recall | F-Measure |
|---|---|---|---|
| Sent. Match | 0.674 | 0.293 | 0.408 |
| Sent. NoMatch | 0.916 | 0.982 | 0.948 |
| Averages | 0.888 | 0.904 | 0.886 |

**Table 5.1:** J48 performance on testing data (10-fold cross validation) for the citance/reference matching problem (Task 1A). Last row of the table contains weighted average values.

- *Rhetorical Category Probability* (**probability_approach**, **probability_background**, **probability_challenge**, **probability_future_work** and **probability_outcome**): five features were exploited to represent the probability of the reference text span to belong to such a facet (from the Dr. Inventor corpus and computed from the Dr. Inventor library).

**Bag of Words (BoW) Features**

We also added both the reference sentence string and the citance sentence string to the set of features and then converted them to word vectors by using WEKA (i.e., bag-of-words).

#### 5.3.1.1 Task 1A: Matching Citations to Reference Papers

The training data was prepared as follows: positive instances of the problem were the pairs of sentences from the citance which when matched with cited text spans from the references (according to information given in the gold annotations). Negative instances, instead, were pairs of sentences from citances to identified cited text spans which were not annotated as matches by the annotators (complementary information). As a consequence we cast the Task 1A as a binary classification problem where we decide for each pair of citance sentence and reference paper sentence whether they match or not, or in other words whether the reference paper sentence reflects the reason of that specific citations. We produced 3,786 instances unevenly distributed (3,356 no matches vs 430 matches). After testing several algorithms from WEKA, we opted for the J48 implementation of decision trees. Ten fold cross-validation results are presented in Table 5.1.

#### 5.3.1.2 Task 1B: Identifying Citation Facets

The training data was prepared similarly to Task 1A (5.3.1.1), pairs of citing sentences and matched cited sentences (according to the gold annotations) were used to create instances. The facet of each instance was also given by the gold standard. This procedure produced just 432 instances with the following distribution: Aim (72), Implication (26), Result (76), Hypothesis (1), Method (257). After testing several algorithms from WEKA, we opted for the Support Vector Machines (SMO) implementation provided by the tool. We used polynomial Kernels and performed no parameter optimization due to time constraints. Ten fold cross-validation results are presented in Table 5.2.

| Class | Precision | Recall | F-Measure |
|---|---|---|---|
| Aim | 0.886 | 0.861 | 0.873 |
| Implication | 0.875 | 0.808 | 0.84 |
| Results | 0.971 | 0.895 | 0.932 |
| Hypothesis | 0.0 | 0.0 | 0.0 |
| Method | 0.929 | 0.969 | 0.949 |
| Averages | 0.924 | 0.926 | 0.924 |

**Table 5.2:** SMO performance on testing data (10-fold cross validation) for the facet identification problem (Task 1B). Last row of the table contains weighted average values.

### 5.3.2 Task 2: Summarizing Scientific Articles

In order to summarize the reference paper by taking into account how it is mentioned in the citing papers, we combined information from the reference and citing papers. We have implemented a series of sentence relevance features, using the resources of the freely available text summarization library SUMMA (Saggion, 2008d, 2014). All these features are numeric and are used to train a linear regression model following the methodology that was already used in (Brügmann et al., 2015).

In addition to a rich set of features provided by the DRI Framework, document processing for summarization is carried out with SUMMA on reference and citing papers. More specifically, the following computations with the library are carried out to enable the summarization of scientific documents:

- Each token (i.e., lemma) is weighted by its term frequency* inverted document frequency, where inverted document values are computed from training data previously analysed (test documents in the CL-SciSumm 2016 dataset);
- For each sentence a vector of terms and normalized weights is created by using the previously computed weights;
- For the title, a single vector of terms and normalized weights is also created (*title vector*);
- Using the normalized sentence term vectors in the whole document, a centroid vector of terms is computed (*document centroid*);
- Using the normalized sentence term vectors of the abstracts, a centroid vector of terms is computed (*abstract centroid*);
- All vectors corresponding to sentences citing the reference paper (from all citing papers) are used to create a centroid (*citances vector*).

The following is the set of sentence relevance features we have used for training a linear regression summarization system. Note that all text-based similarities we mention are the result of comparing two vectors using the cosine similarity function implemented in SUMMA. The reference paper features are as follows:

- *Sentence Abstract Similarity (***abs_sim***)*: the similarity of a sentence to the author abstract;
- *Sentence Centroid Similarity (***centroid_sim***)*: the similarity of a sentence to the document centroid (e.g., the average of all sentence vectors in the document);
- *First Sentence Similarity (***firt_sim***)*: the similarity of a sentence to the title vector;
- *Position Score (***position_score***)*: the SUMMA implementation of the position method where sentences at the beginning of the document have high scores and sentence at the end of the document have low scores;
- *Position in Section Score (***in_sec***)*: a score representing the position of the sentence in the section of the document. Sentences in the first section get higher scores, sentences in the last section get lower scores;

- *Sentence Position in Section Score (***in_sec_sent***)*: a position method applied to sentences in each section of the document (sentences at the beginning of the section get higher scores and sentences at the end of the section get lower scores);
- *Normalised Cue-phrase Score(***norm_cue***)*: we produce a normalized score for each sentence which is the total number of cue-words in the sentence divided by the total number of cue-words in the document. We have relied on Teufel & Moens (2002)'s formulaic expressions to implement our cue-phrase gazetteer lookup procedure;
- *TextRank Normalized Score (***textrank_score***)*: the SUMMA implementation of the TextRank algorithm (Mihalcea & Tarau, 2004a) but with a normalization procedure which yields values for sentences between 0 and 1.

The cluster-based features are as follows:

- *Citing Paper Maximum Similarity (***cps_max***)*: each reference paper sentence vector is compared (using cosine) to each citance vector in each Citing Paper to obtain the maximum possible cosine similarity;
- *Citing Paper Average Similarity (***cps_avg***)*: the average cosine similarity between a reference paper vector and all citance vectors in the cluster is produced;
- *Citing Paper Citances Similarity (***cps_sim***)*: the similary of the sentence vector to the centroid of the citance vectors.

The approach taken to score sentence is to produce a cumulative score of the weighted values of summarization features $f_1, ...f_n$ using the following formula:

$$score(S) = \sum_{i=0}^{n} w_i * f_i \tag{5.1}$$

with $S$ as the sentence to score, $f_i$ as the value of feature $i$ and $w_i$ as the weight assigned to feature $i$. As we stated before, the weights of each feature in the formula are learned from training data. We fit a linear regression model using 10 testing documents from the provided annotated document for a total of 2,585 instances. The target numerical value to learn is computed from two sources (giving rise to two different systems): On the one hand, we compute

the similarity of each reference paper sentence (i.e. vector) to the combined vectors of texts fragments identified as the annotators as cited text spans; on the other hand, we compute the similarity of each reference paper sentence (i.e. vector) to a vector of the community-based summary provided for training by the organizers.

### 5.3.3 The Final System

The final system was assembled as follows. Given a cluster of documents with reference and citing papers, the following procedure was applied for tasks 1A and 1B.

1. The documents were annotated with the citance information (no matched reference sentences were annotated);
2. All the document processing algorithms were applied to reference and citing papers as described in Section 5.2.1 and the features computed;
3. Instances were created using a citance sentence from each Citing Paper and each sentence from the reference paper;
4. The instances were sent to the matching classifier which returned a match/no match class and a confidence value;
5. The matched instances according to the previous steps were sent to the facet classifier to obtain the predicted citation facet.

Two runs were produced for tasks 1A and 1B. In one run, all matched sentences for a given citance were returned. In a second run, only top matches (with higher confidence i.e. 0.80) were returned. In order to produce the summaries for each cluster, summarization features were computed using the procedure described in Section 5.3.2, and SUMMA was exploited to score and extract top scored sentences based on formula (5.1). Two 250-word text extractive summaries were produced per cluster using the models described in Section 5.3.2.

### 5.3.4 Results Comparison Against the Other Participants

Ten teams participated in this shared task with a total of 23 submissions, the organizers compared the performance of all the systems and provided a report with detailed results (Jaidka et al., 2016).

Table 5.3 provides the system ID prefixes mapped to system description papers in the shared task. The results for Task 1 were provided in Table 5.4 while the results of Task 2 can be seen in Table 5.5. We believe that our system did not perform well for Task 1A for many reasons, the dataset provided by the organizers had a lot of noise during the OCR process of converting the PDF into text. The dataset were annotated by one annotator per cluster only, there were many instances in which the annotated Reference Paper sentence that best reflects the citing paper citation is just the title of the Reference Paper. Finally, the way we modeled our approach by doing a binary classification with match or no-match made our labels skewed since most of the sentences were annotated as not-match.

| System | Reference | System | Reference |
|---|---|---|---|
| sys3 | (Conroy & Davis, 2015) | sys10 | Our System |
| sys5 | (Malenfant & Lapalme, 2016) | sys12 | (Lu et al., 2016) |
| sys6 | (Nomoto, 2016) | sys13 | (Aggarwal & Sharma, 2016) |
| sys8 | (Li et al., 2016) | sys15 | (Moraes et al., 2016) |
| sys9 | (Klampfl et al., 2016) | sys16 | (Cao et al., 2016) |

**Table 5.3:** System ID prefixes mapped to system description papers.

| Task 1A | | Task 1B | |
|---|---|---|---|
| **System id** | **F1 score** | **System id** | **F1 score** |
| sys15 | 0.134 | sys8 | 0.317 |
| sys8 | 0.126 | sys16 | 0.153 |
| sys6 | 0.096 | sys10 (Our System) | 0.139 |
| sys16 | 0.094 | sys15 | 0.068 |
| sys9 | 0.051 | sys5 | 0.064 |
| sys13 | 0.047 | sys13 | 0.053 |
| sys5 | 0.039 | sys12 | 0.011 |
| sys10 (Our System) | 0.023 | | |
| sys12 | 0.021 | | |

**Table 5.4:** Task 1 results for the participant's best systems at the CL-SciSumm 2016 shared task.

For task 1A our system did not perform well, as for Task 1B our system ranked as the third best. Regarding Task 2, our system's performance versus abstract summaries was the third best for both metrics ROUGE-2 and ROUGE-SU4. Moreover, against the human summaries we achieved the second best system for ROUGE-2 and the third best for ROUGE-SU4. Finally, against the com-

| System | Vs. Abstract | | Vs. Human | | Vs. Community | |
|---|---|---|---|---|---|---|
| | R–2 | RSU–4 | R–2 | RSU–4 | R–2 | RSU–4 |
| System10 (Our System) | 0.192 | 0.124 | 0.134 | 0.092 | 0.245 | 0.162 |
| System15 | 0.177 | 0.106 | 0.133 | 0.089 | 0.254 | 0.166 |
| System16 | 0.052 | 0.053 | 0.070 | 0.047 | 0.157 | 0.129 |
| System3 | 0.399 | 0.214 | 0.189 | 0.124 | 0.139 | 0.107 |
| System5 | 0.099 | 0.086 | 0.084 | 0.065 | 0.106 | 0.082 |
| System8 | 0.668 | 0.422 | 0.219 | 0.136 | 0.249 | 0.150 |

**Table 5.5:** Task 2 results for the participant's best systems at the CL-SciSumm 2016 shared task, the systems were evaluated against the target paper's abstract, human summaries and community summaries.

munity summaries our system was the third best system for ROUGE-2 and the second best for ROUGE-SU4.

What we learned after our participation in the Computational Linguistics Scientific Document Summarization 2016 is that modeling Task 1A as a binary classification problem was not a good idea, since there are a lot of sentences that are labeled as not-match and only few sentences have cited by each citation. This modeling of the problem made the data skewed, making the classifier leaning mostly towards predicting non-match for the Reference Paper sentences.

## 5.4 Participation in the $2^{nd}$ CL-SciSumm Shared Task (2017)

In this iteration of the shared task the CL-SciSumm organizers have expanded the data by making the total number of clusters to 40: 30 clusters as training data and 10 clusters as testing. For the 2017 challenge of the CL-SciSumm shared task we decided to change our approach for Task 1A since our system for the year 2016 had not performed well. After considering the results from the 2016 challenge and reading the work of other participants, we noticed that a voting theme over multiple systems was a promising idea. Hence we decided to change the way we model the problem for Task 1A from a binary classification into a sentence ranking problem using a different unsupervised approaches and also incorporating a voting theme among those systems. Finally, we used the same methods for Task 1B and Task 2 with a noticeable fine tuning and addition of features (Abura'ed et al., 2017).

### 5.4.1 Task 1A: Matching Citations to Reference Papers

In this section we present the experiments aimed to tackle Task 1A, and we present the experiments we applied to find the sentences that have been cited in the Reference Papers.

#### Word Embeddings Distance

We used the Google News embeddings[39] (three million words in 300 dimensional vectors trained using word2vec (Mikolov et al., 2013a) over a news text corpus of 100 billion words) and the ACL Anthology Reference Corpus embeddings (Liu, 2017) (100 and 300 dimensional vectors trained over a corpus of ACL papers (Bird et al., 2008)). Words with similar meanings generate vectors that are close in the embeddings space. From these vectors it is possible to create embeddings for larger units such as phrases, sentences or paragraphs. A simple technique for creating text embeddings that has achieved good results in tasks like extractive summarization (Kågebäck et al., 2014) and semantic classification (White et al., 2015) is to use the average–or centroid–of the words contained in the texts as their vectorial representations. The embeddings thus created tend to keep the proximity relation if the texts they represent have related–close–words.

We built embeddings for each citance in the citing papers by taking the centroid of the embeddings of all the words contained in it. The same procedure was used to build embeddings for each of the sentences of the reference paper. Afterwards, We calculated the cosine similarity between both embeddings in the vector space. We experimented with different combinations of embeddings: using only Google News vectors, using only ACL vectors (100 or 300 dimensions) and using the concatenation of Google News and ACL vectors (400 or 600 dimensions). We ran several test considering as candidates the top two, five, eight and ten sentences from the reference papers most similar to the given citances. Since the evaluation used for Task 1B uses the F score, we aimed at optimizing this metric. We did this by saving 10 clusters from the training data for validation and using 20 clusters for training. The best performance for the validation set was achieved using the concatenation of Google and ACL-300 vectors and considering the two top candidate sentences from the reference papers.

#### Modified Jaccard

[39] https://code.google.com/archive/p/word2vec/

We used a modified version of the Jaccard similarity index that takes into consideration the inverted frequency of the word in a corpus instead of just the word occurrences. For this experiment we calculated IDF values of word stems using both the training set and an ACL reference corpus of around 4,000 documents. The modified Jaccard similarity between two text spans $s_1$ and $s_2$ is defined in equation 5.2. Our modification assigns greater weight to matching word stems that are infrequent in the corpus, based on the idea that two text spans that share infrequent words are more likely to be semantically related.

$$MJ(s_1, s_2) = \frac{\sum_{t \in s_1 \cap s_2} 2^{idf(t)}}{|s_1 \cup s_2|} \tag{5.2}$$

### BabelNet Embeddings Distance

BabelNet (Navigli & Ponzetto, 2012a) is an ontology of concepts (synsets) that integrates many resources, including Wikipedia and WordNet. We used a set of BabelNet embeddings (Mancini et al., 2016) containing 2.5 million vectors trained over a corpus of 300 million words tagged with BabelNet synsets. Using the Babelfy API,[40] we obtained the list of BabelNet synsets associated to each sentence of the corpus and used them to build sentence embeddings analogously as we did with the word embeddings. The BabelNet embeddings include many vectors for each synset (one for each lexicalization). We therefore calculated the centroid of all the vectors associated to each synset to generate its embedding. We proceeded analogously to the word embeddings experiment described above: we calculated embeddings for the citances and for the sentences in the Reference Papers and then selected as candidates the top $N$ sentences according to their cosine distance to the citances. We did some tests over the validation corpus to determine the value of $N$, the best results were again achieved considering the two sentences from the Reference Papers that are most similar to the citances.

### Voting System

We propose a system that leverages the best results obtained by the word embeddings, Modified Jaccard and BabelNet embeddings systems: the top five candidates obtained for each of the systems are first considered and then a voting process chooses candidates from all the sentences that were selected by at least two systems. If no sentence was chosen by at least two, only the

[40] http://babelfy.org/guide

top sentence selected by the Modified Jaccard system[41] is returned. Unlike the other systems described for this task–where a fixed number of candidate sentences are returned–in this case the number of sentences obtained is variable.

Table 5.6 shows the performance of the results over the validation data. The experiments are word embeddings (WE), Modified Jaccard (MJ), BabelNet embeddings (BN) and the voting scheme (Voting). The best results over the validation corpus are achieved by the voting system.

| Method | Avg. Precision | Avg. Recall | Avg. F-Measure |
|---|---|---|---|
| WE | 0.077 | 0.116 | 0.091 |
| MJ | 0.120 | 0.184 | 0.144 |
| BN | 0.083 | 0.127 | 0.099 |
| Voting | 0.117 | 0.199 | **0.146** |

**Table 5.6:** Performance for Task 1A over the validation corpus.

### 5.4.2 Task 1B: Identifying Citation Facets

In this section we present experiments aimed at identifying the facets to which the cited text spans belong. We modeled pairs of reference and citance sentences as feature vectors, which we then used to train classification algorithms that determine whether a cited text span belongs to one of the predefined facets. We have adopted the same features from our participation at the CL-SciSumm 2016 shared task in addition to a new set of features as described below. For the classification algorithms we relied on implementations included in the WEKA machine learning framework (Witten et al., 2016).

#### Features

In addition to the set of features we used at the CL-SciSumm 2016 shared task (See Section 5.3) we calculated the following features:

**Text Similarity Features**: The more similar a text is to another the more likely it is that they will be part of the same facet. We used TF*IDF vector representations of the sentences produced by the SUMMA library. In addition to the one based on word lemmas from section 5.3, we added one based on BabelNet synsets–and computed their cosine similarity. We also calculated the

[41] Modified Jaccard was chosen as default as it was the system for which the best F-measures were obtained when run independently.

Jaccard and Modified Jaccard coefficients for the lemmas, generating a total of four text similarity features.

**Dr. Inventor Sentence Related Features**: Other features obtained by means of the DRI Framework that we believed could be of use in predicting a sentence belonging to a particular facet include:

- Citation marker: three features to represent the number of citation markers in the reference sentence, citing sentence and the pair of sentences together;
- Cause and effect: two features to represent if the reference or citing sentence participates in one or more causal relations;
- Co-reference chains: three features to represent the number of nominals and pro-nominals chained in the reference sentence, citing sentence and the pair of sentences together

**Scientific Gazetteer Features**: We generated a set of features based on Teufel's action and concept lexicon. The lexicon contains 58 lists. Each one is used to produce a feature which is the ratio of words in the sentence matching the list to the number of words in the sentence. The features are computed for the reference sentence, the citing sentence, and their combination, giving rise to 174 features.

**Bag-of-word Features**: four string features are produced to represent the *bi-gram lemmas*, *POS-tags bi-gram*, *lemmas* and *POS-tags* for the combination of the reference and the citing sentences.

Based on these features we trained classifiers with 1,386 instances distributed as follows: *Aim* (134), *Implication* (150), *Result* (262), *Hypothesis* (32), *Method* (808). Considering the skewed distribution of the *Method* facet, we decided to train two models: one binary classifier to predict whether the instance is a *Method* or not and a multi-class classifier to identify one of the other facets in case it was previously classified as *not-Method*. We evaluated the performance of several classification algorithms including: SMO algorithm for Support Vector Machines (SMO), naive Bayes (NB), K-nearest neighbors (IBk), random committee (RC), logistic regression (LR) and random forest (RF). We performed 10-fold cross validation experiments with the training data in order to decide which algorithm to use. The best results were obtained with the RF algorithm for the binary *Method* classifier and the SMO for the multi-class classifier representing the *non-Method* facet (Table 5.7).

| Classifier | Algorithm | Avg. Precision | Avg. Recall | Avg. F-Measure |
|---|---|---|---|---|
| Method Facet [Binary] | RF | 0.882 | 0.875 | 0.873 |
| Other Facets [Multi-class] | SMO | 0.921 | 0.920 | 0.920 |

**Table 5.7:** Algorithms used for the two classifiers trained over the described set of features, evaluated with 10-fold cross validation, with their Precision, Recall and F-measure scores.

### 5.4.3 Task 2: Summarizing Scientific Articles

The proposed summarizer is a modified version of our 2016 summarization system (See Section 5.3.2) with some additional features.

A set of additional computed information has been added as follows:

- Using the ACL word embeddings, a vector is created for each sentence in the document–average of the word embeddings of the words in the sentence (ACL vectors);
- Using the Google news word embeddings, a vector is created for each sentence in the document–average of the word embeddings of the words in the sentence (Google vectors);
- Using the sentence vectors (ACL, Google), two centroids are created for the document–each an average of the sentence vectors in the whole document;
- Using the sentence vectors (ACL, Google), two centroids are created for the abstract of the document–each an average of the sentence vectors in the abstract;
- In the citing papers, token frequency and ACL, Google vectors are also computed.

The additional features to help train the linear regression algorithm are described below. Text similarity features are the result of comparing two vectors of the same type (e.g. ACL, or Google) using the cosine similarity function implemented in SUMMA. Therefore, two different feature values are always generated. The reference paper features are as follows:

- Sentence Abstract Similarity Scores: for ACL and Google vectors;
- Sentence Centroid Similarity Scores: for ACL and Google vectors;
- First Sentence Similarity Scores: for ACL and Google vectors;

- TextRank Normalized Scores: the SUMMA implementation of the TextRank algorithm (Mihalcea & Tarau, 2004a) but with a normalization procedure which yields values for sentences between 0 and 1. Each score is computed using a different sentence vector (ACL, and Google);
- Term Frequency Score: we sum up the TF*IDF values of all content words in the sentence and the obtained value is normalized to yield a value between 0 and 1 which is computed using the set of scores from the entire document;
- Citation Marker Score: the ratio of the number of citation markers in the sentence to the total number of citation markers in the paper;
- Rhetorical Class Probability Scores: the probability that the sentence belongs to a DRI rhetorical class;

The Citing Paper features are as follows:

- Citing Paper Maximum Similarity Scores: for ACL and Google vectors;
- Citing Paper Minimum Similarity Scores: each reference paper sentence vector is compared to each citance vector in each Citing Paper to get the minimum possible cosine similarity (for ACL, and Google vectors);
- Citing Paper Average Similarity Scores: for ACL and Google vectors;

| Method | ROUGE-2 | | | ROUGE-SU4 | | |
|---|---|---|---|---|---|---|
| | Abstract | Community | Human | Abstract | Community | Human |
| ACL_abs | **0.2985** | **0.2000** | **0.1907** | **0.2066** | **0.1164** | **0.1347** |
| ACL_com | 0.2164 | 0.1889 | 0.1195 | 0.1656 | 0.1129 | 0.1070 |
| ACL_hum | 0.0996 | 0.1163 | 0.1055 | 0.0924 | 0.0681 | 0.0895 |
| Google_abs | 0.2477 | 0.1870 | 0.1365 | 0.1813 | 0.1045 | 0.1003 |
| Google_com | 0.1032 | 0.1600 | 0.0676 | 0.0914 | 0.0832 | 0.0615 |
| Google_hum | 0.1443 | 0.1143 | 0.0531 | 0.1201 | 0.0701 | 0.0675 |
| SUMMA_abs | 0.2402 | 0.1436 | 0.1208 | 0.1526 | 0.0860 | 0.0888 |
| SUMMA_com | 0.1687 | 0.1797 | 0.0975 | 0.1189 | 0.0867 | 0.0765 |
| SUMMA_hum | 0.2181 | 0.1722 | 0.1516 | 0.1611 | 0.1139 | 0.1121 |

**Table 5.8:** ROUGE-2 and ROUGE-SU4 results for all configurations before submitting our Task 2 runs. Twenty document clusters from the training data were used and all models were tested over eight document clusters from the testing data

As stated before, the weights for each feature are obtained from training data and although the ideal score to be learned is in principle unknown, we approximate it with training data. By relying on the gold standard summaries–(a) an author abstract, (b) a human-written abstract, and (c) a community-based abstract–we created different target scores. We compared, using cosine similarity, each sentence vector in the reference paper with each vector in the summary and used the *maximum* similarity values as the target score for the reference paper (e.g., *score(S)*) for learning. This method produced nine different functions to learn: SUMMA, ACL, and Google vectors times *abstract*, *community*, *human* summaries. Note that other target functions are possible but we restricted the number of systems to nine given time constraints. The number of instances used to train the linear regression models was 6,372.

**Evaluating the Summarization Models**

Before submission, we carried out a preliminary evaluation of the nine models using 20 document clusters for training and eight document clusters for testing (we could not use two clusters due to errors generated when processing some of the documents in them). The evaluation framework adopted was to compare each of the summaries generated by the model (9 models times 8 clusters = 72 abstracts) against each of the summary types given by the organizers: *abstract*, *community*, and *human*. The comparison was carried out using ROUGE-2 and ROUGE-SU4 (Lin, 2004) (following the configuration suggested by the task organizers). Average results are presented in Table 5.8 where we highlight the best scores.

### 5.4.4 Submissions to the Challenge and Results

We submitted four runs for tasks 1A, each one applying one of the methods described in Section 5.3.1, with the results obtained by the *Method/No-Method* Facet Classifier for Task 1B.

In addition to the evaluation measures the organizers usually perform, they calculated the resulting ROUGE-2 score for Task 1A at the CL-SciSumm 2017. The results they obtained with the test set are shown in Table 5.9, where we include our best result–obtained with the voting system– as well as the maximum, mean and minimum scores for all the systems submitted (macro averages).

For Task 2 we submitted nine trainable systems corresponding to nine ways of interpreting the gold standard summaries: three vector representations times three gold standard summaries (system names in first column of Table 5.8).

| Score | Task 1A Avg F1 | Task 1A ROUGE2 F1 | Task 1B Avg F1 |
|---|---|---|---|
| LaSTUS/TALN | 0.1070 | 0.0912 | 0.2930 |
| Min. score | 0.0205 | 0.0339 | 0.0000 |
| Mean score | 0.0882 | 0.0714 | 0.2080 |
| Winning score | 0.1463 | 0.1142 | 0.4081 |

**Table 5.9:** LaSTUS/TALN Task 1 best results vs. minimum, mean and maximum scores

In Table 5.10 we show our results compared to the mean, minimum and maximum results obtained in the challenge.

| Score | ROUGE-2 | | | ROUGE-SU4 | | |
|---|---|---|---|---|---|---|
| | Abstract | Community | Human | Abstract | Community | Human |
| LaSTUS/TALN | 0.2974 | 0.2169 | 0.1906 | 0.1635 | 0.1655 | 0.1692 |
| Method | SUMMA_abs | ACL_com | ACL_abs | ACL_abs | ACL_com | ACL_com |
| Min. score | 0.0525 | 0.1203 | 0.0748 | 0.0652 | 0.0918 | 0.0963 |
| Mean score | 0.2374 | 0.1926 | 0.1638 | 0.1500 | 0.1413 | 0.1450 |
| Winning score | 0.3506 | 0.2755 | 0.2038 | 0.1914 | 0.1780 | 0.1740 |

**Table 5.10:** LaSTUS/TALN Task 2 best results vs. minimum, mean and maximum scores

### 5.4.5 The CL-SciSumm 2017 Results Comparison VS the Other Participants

Nine teams participated in Task 1 with a total of 47 systems' submissions and a subset of five teams also participated in Task 2 with a total of 25 systems' submissions. The CL-SciSumm 2017 organizers evaluated all the systems and reported the results (Jaidka et al., 2017b). See table 5.11 which compares the systems for Task 1 of the challenge and table 5.12 that presents the ROUGE scores for Task 2.

Our system for Task 1 performed as the sixth and fifth score for Task 1A and 1B respectively. As for Task 2 over abstract summaries it obtained the second best for ROUGE-2, third best for ROUGE-SU4. As for the human summaries our system ranked fourth for ROUGE-2 but the second best for ROUGE-SU4. Finally, for the communities summary our system was the second best for ROUGE-2 and ROUGE-SU4 as well. We learned that adding more features to Task 1 improved the results over the 2016 participation. However, our Task 1 submission fall out to be within the top 3 systems. Same applies for Task 2 in which by adding ACL and Google embedding to the set of existing features, alongside additional similarity features we managed to achieve a great score over the rest of the participants.

| System | Task 1A: Sentence Overlap (F1) | Task 1B |
|---|---|---|
| NJUST (Ma et al., 2017) | **0.124** | 0.339 |
| TUGRAZ (Felber & Kern, 2017) | 0.110 | 0.337 |
| CIST (Li et al., 2017) | 0.107 | **0.373** |
| PKU (Zhang & Li, 2017) | 0.102 | 0.370 |
| UHouston (Karimi et al., 2017) | 0.091 | 0.271 |
| UPF (our system) | 0.088 | 0.293 |
| NUS (Prasad, 2017) | 0.055 | 0.026 |
| UniMA (Lauscher et al., 2017b) | 0.053 | 0.114 |
| Jadavpur (Pramanick et al., 2017) | 0.042 | 0.100 |

**Table 5.11:** Participants' best performing systems in Task 1, ordered by their F1-scores for sentence overlap on Task 1A.

| System | Vs. Abstract | | Vs. Human | | Vs. Community | |
|---|---|---|---|---|---|---|
| | R–2 | RSU–4 | R–2 | RSU–4 | R–2 | RSU–4 |
| CIST (Li et al., 2017) | **0.351** | 0.185 | **0.275** | **0.178** | **0.204** | 0.168 |
| UPF (our system) | 0.297 | 0.163 | 0.217 | 0.166 | 0.191 | 0.169 |
| UniMA (Lauscher et al., 2017b) | 0.257 | **0.191** | 0.221 | 0.166 | 0.178 | **0.174** |
| NJUST (Ma et al., 2017) | 0.214 | 0.138 | 0.229 | 0.154 | 0.152 | 0.114 |
| Jadavpur (Pramanick et al., 2017) | 0.191 | 0.133 | 0.181 | 0.129 | 0.132 | 0.119 |

**Table 5.12:** Systems' performance at the CL-SciSumm 2017 for Task 2 ordered by their ROUGE–2(R–2) and ROUGE–SU4(R–SU4) F1-scores.

## 5.5 Participation in the $3^{rd}$ CL-SciSumm Shared Task (2018)

The CL-SciSumm shared task organizers has expanded the data for the year 2018 by making the total number of the clusters 60: 40 training clusters and 20 testing clusters. For this year of the challenge we used deep learning techniques alongside supervised and unsupervised approaches to tackle the different tasks. Task 2 was completely based on Convolutional Neural Network (Zhang et al., 1990) while Task 1 adopted a voting scheme of systems from the previous years alongside a new system based on the CNN.

### 5.5.1 Task1: Identifying Cited Sentences and Their Facets

After the organizers reported the key insights for the CL-SciSumm Shared Task iteration on 2017 (Jaidka et al., 2017a), they recommended that future approaches should exploit the structural and semantic characteristics that are

unique to scientific documents and also to go beyond off-the-shelf deep learning methods. For the CL-SciSumm 2018 iteration we decided to submit multiple runs including some of our old successful approaches in addition to the use of deep learning techniques that makes use of structural and semantic characteristics of the scientific papers.

For the new method, we utilized a deep-learning approach (Convolutional Neural Network (CNN) (Zhang et al., 1990)) formulating the problem of finding a set of sentences in a reference paper. The sentences that best reflects a citation in a Citing Paper as a sentence ranking problem which uses a CNN with two inputs and one output. The first input models the reference paper sentences as a Word2Vec representation and the second input calculates a set of features based on the pair of sentences (reference paper sentence and a citation sentence). On the other hand, the network outputs a score (regression score) for each sentence in the reference paper based on the set of citations citing the reference paper. The output score of each reference sentence is based on the position distance length from a sentence that is being cited. A value of 1 is set to cited sentences and the further the sentence is from the nearest cited sentence the less score it has.

The formula we used to score the sentences in the RP can be seen below 5.3 where S: Reference Paper sentence; MD: minimum distance from a cited sentence; T: total number of sentences in the Reference Paper, and Figure 5.5 shows a scoring visual representation of scoring.

$$score(S) = 1 - (MD/T) \tag{5.3}$$

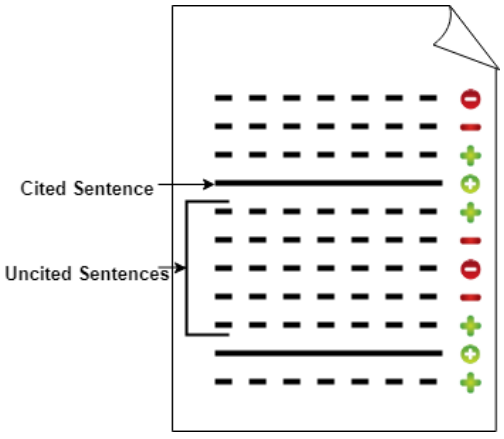


**Figure 5.5:** The scoring theme of a reference paper sentences, the closer a sentence is to a cited sentence the higher score it has.

We also used the same neural network to predict the facet to which a cited sentence belongs. However, for facets we formulated the problem as a classification problem in which the output in that case is one of the five predefined facets classes provided by the organizers.

We modeled each reference sentence as a Word2Vec representation from three different pre-trained Word2Vec models embedded in a 300 dimensional space: (1) ACL [42](Liu, 2017) from the ACL Anthology Reference Corpus (Bird et al., 2008), (2) Google News [43], and (3) Babelnet (Camacho-Collados et al., 2016).

From each reference paper we extracted all the sentences having a number of tokens in a range of 5 to 40 and we used the 300 dimensions of each of the first 15 tokens from each sentence. In order to reduce the number of pairs of sentences to consider, we also excluded sentences which according to the analysis carried out with the Dr. Inventor library belongs to the *Background* or *Future Work* discourse facets since it is assumed those sentences will mainly refer to work carried out by other authors or are still inexistent.

#### 5.5.1.1 Set of Features

As we mentioned in section 5.5.1 in addition to the first input which models sentences as a Word2Vec representation we added a set of features as a second input. The second input is formed by a calculated set of features based on the pair of the cited sentence in the reference paper and the sentence citing it in the Citing Paper. Those features were modeled and motivated to identify the cited sentences and their facets. They were based on the same features from CL-SciSumm 2017 participation based on: Sentence Position Features, WordNet Semantic Similarity Measures Features, Text Similarity Features, Dr. Inventor Sentence Related Features and Scientific Gazetteer Features.

We ran three CNNs over each sentence embeddings in which the width is the 300 dimensions, the height is 2, 3 or 4 respectively to represent: bi, tri and quadri-grams and finally, 3 channels to present the three pre-trained models.

#### 5.5.1.2 Unsupervised Approaches

In addition to the CNN approach we also decided to submit our systems from the 2017 participation (See Section 5.4.1) i.e. Modified Jaccard and Babelnet synset embedding similarities. We also used a voting system over the reported systems.

[42]https://github.com/liuhaixiachina/Sentiment-Analysis-of-Citations-Using-Word2vec/tree/master/trainedmodels

[43]https://code.google.com/archive/p/word2vec/

The only parameter to adjust using these methods is the number of sentences to consider as candidates. In order to optimize this parameter, we tested against CL-SciSumm 2017 test data, which is also a subset of CL-SciSumm 2018 training data. The results of these experiments are shown in table 5.13. The best result is achieved using the BabelNet Embeddings metric, considering only the closest sentence as candidate. The best result for Modified Jaccard is also close in F-measure.

| Method | #Sents | Precision | Recall | F-Measure |
|---|---|---|---|---|
| MJ | 1 | 0.120 | 0.103 | 0.111 |
| MJ | 2 | 0.072 | 0.115 | 0.088 |
| BN | 1 | 0.123 | 0.105 | **0.113** |
| BN | 2 | 0.085 | 0.142 | 0.106 |

**Table 5.13:** Performance for Task 1A unsupervised approaches over the CL-SciSumm 2017 test set.

#### 5.5.1.3 Voting scheme

We designed a voting scheme that intended to leverage the strengths of the different supervised and unsupervised approaches. Since the organizers asked for a maximum of 5 sentences we decided to take a subset of sentences from each run and perform and intersection between them, then choose up to 5 sentences from that set. After trying different scenarios of the number of sentences to choose from each run we considered these four system runs for the voting scheme:

- Top 10 sentences from the Convolutional Neural Network using learning rate 0.0001.
- Top 10 sentences according to Modified Jaccard unsupervised approach.
- Top 10 sentences according to BabelNet Embeddings unsupervised approach.
- Top 40 sentences for each target paper according to the relevance scores.

The voting scheme returns a candidate sentence if at least $N$ (a value between one and four) of the four systems agree on that sentence. The results are ordered according to the maximum relevance score (regardless of the run they were chosen from) and if there are more than five candidates, only the top

five are selected. If there are no candidates in the intersection, the top sentence according to the BabelNet embeddings approach is used as a fallback mechanism. We submitted two runs using $N = 2$ and $N = 3$.

### 5.5.2 Task 2: Summarization of Scientific Articles

In this section, we describe our extractive text summarization approach based on convolutional neural networks which extends on our previous work on trainable summarization (Saggion et al., 2016a; AbuRa'ed et al., 2017). The network generates a summary by selecting the most relevant sentences from the RP using linguistic and semantic features from RP and CPs. The aim of our CNN is to learn the relation between a sentence and a scoring value indicating its relevance.

#### 5.5.2.1 Context Features

In order to extract the linguistic information from both sources (RP and CPs), we reused the features we implemented during our past participation at CL-SciSumm 2017 (See section 5.4.3) for Task 2 and we developed a complex feature extraction method to characterize each sentence in the RP and its relation with the corresponding CPs.

#### 5.5.2.2 Scoring Values

As commented above, our CNN learns the relation between features and a score, that is, a regression task by devising various scoring functions to represent the likelihood of a sentence belonging to a summary (for abstract, community and human). The nomenclature followed to symbolize a scoring function is $SC_{Sum}$, where $SC$ is the specific scoring function (which is indicated bellow) and $Sum$ is any summary type: abstract ($Abs$), community ($Com$) or human ($Hum$). The scoring functions are defined bellow:

- Cosine Distance: we calculated the maximum cosine similarity between each sentence vector in the RP with each vector in the gold standard summaries. This method produced three scoring functions (SUMMA ($SU_{Sum}$), ACL ($ACL_{Sum}$), and Google ($Go_{Sum}$)) for each summary type.
- ROUGE-2 Similarity: we also calculated similarities based on the overlap of bigrams between sentences in the RP and gold standard summaries. In this regard, each sentence in the RP is compared with each gold standard summary using ROUGE-2 (Lin, 2004). The precision value

from this comparison is taken for the scoring function and is symbolized as $R2_{Sum}$.

- Scoring Functions Average: Moreover, we computed the average between all scoring functions (SUMMA, ACL, Google and ROUGE-2) for each summary type. In addition, we also calculated a simplified average with vectors not based on word-frequencies (ACL, Google and ROUGE-2). These scoring functions are indicated as $Av_{Sum}$ and $SAv_{Sum}$, respectively.

Finally, these computations produced eighteen different functions to learn: SUMMA (*SU*), ACL (*ACL*) and Google (*Go*) vectors, ROUGE-2 (*R*2), Average (*Av*) and Simplified Average (*SAv*) times abstract (*Abs*), community (*Com*), human (*Hum*) summaries.

#### 5.5.2.3 Convolution Model

CNN consists of multiple convolutional and pooling layers, with fully-connected layers at the end. The network is fed with two different inputs. The inputs are composed of instances related to sentences. The first one is based on the context features. Specifically, *context features* are introduced in the CNN within a sequential window including the context features of the 3 previous and 3 following sentences. And the second input is related to the word embedding information for each sentence. In particular, we used both word embeddings (Google and ACL) as a dual channel, whose stopwords were removed. The size was fixed at 15 words and they were kept static during the training.

Regarding the neural network hyperparameters, the CNN was defined with the Adadelta updater (Zeiler, 2012) and the gradients were computed using back-propagation like Kim (Kim, 2014) and Nguyen (Nguyen & Grishman, 2015). Also, we used the sigmoid activation function, a dropout rate of 0.5, l2 constraint of 3. For the convolutions, we applied 3 filter window sizes (3, 4 and 5) to context features and 4 filter window sizes (2, 3, 4 and 5) to word embeddings. For each window, 150 filters were applied for convolution. Finally, for learning the regression task we applied a Mean Squared Error (MSE) as a loss function.

The Convolution Model Architecture can be seen at Figure 5.6, Reference Paper's sentences are interpreted into two inputs; word embeddings (Google News and ACL) and Context Features with CNNs identifying features based on the surrounding context. Finally, a score is generated for each sentence.

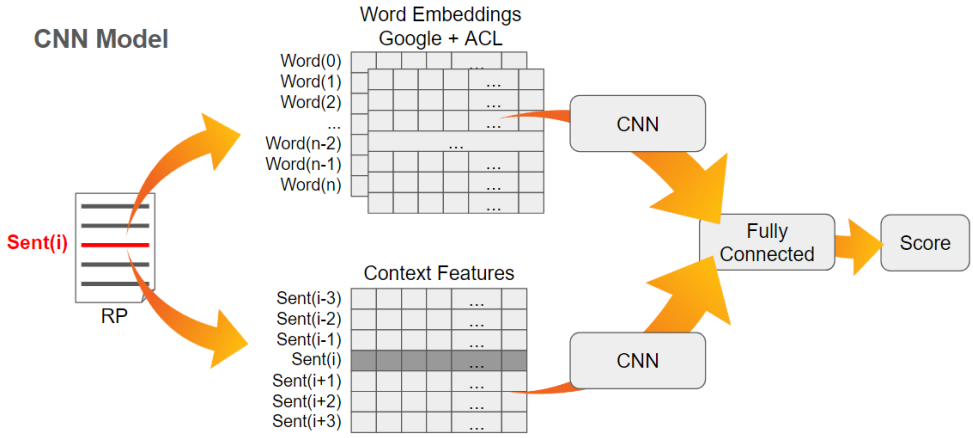


**Figure 5.6:** The Convolution Model Architecture

### 5.5.3 Evaluation

The evaluation consists of generating a 250-word summary according to the task, which is compared against each of the summary types of the gold standard: the reference paper's abstract, a community summary, and a human summary. We trained and evaluated our model using the CL-SciSumm-17 dataset.

| Method | ROUGE-2 | | | ROUGE-SU4 | | |
|---|---|---|---|---|---|---|
| | Abstract | Community | Human | Abstract | Community | Human |
| Winning Score 2017 | 0.351 | 0.217 | 0.275 | 0.191 | 0.174 | 0.178 |
| Our System Scoring Function | 0.555 $SAv_{Abs}$ | 0.274 $Av_{Hum}$ | 0.288 $R2_{Abs}$ | 0.290 $SAv_{Abs}$ | 0.193 $Av_{Hum}$ | 0.240 $Go_{Abs}$ |

**Table 5.14:** ROUGE-2 and ROUGE-SU4 best results for each summary evaluation. In addition, the scoring function employed is specified under each value. The results are based on the F-score value.

Table 5.14 shows the winning scores achieved by the participants in the CL-SciSumm-17 Shared Task 2 (in the first row) and also shows the most representative results achieved in our experiments (in the second). The values shown are also based on the F-score obtained in both ROUGE-2 and ROUGE-SU4 evaluations.

### 5.5.4 Challenge Submissions

We have submitted the following systems for Task 1 (a and b):

- MJ1: top sentence from the unsupervised approach using Modified Jaccard similarity
- BN1: top sentence from the unsupervised approach using BabelNet synset embeddings cosine similarity
- 0.1CNN4: deep learning approach using CNN over the word embedding and a set of features . Learning rate: 0.1 Epoch: 50
- 0.0001CNN4: deep learning approach using CNN over the word embedding and a set of features . Learning rate: 0.0001 Epoch: 50
- Voting2: keep candidates if at least two of the systems agree (MJ, BN, CNN or top 40 sentences from the summarization system 5.5.2)
- Voting3: keep candidates if at least three of the systems agree (MJ, BN, CNN or top 40 sentences from the summarization system 5.5.2)

For the task 2, we have submitted eighteen summaries related to each scoring function and summary. In other words, each resulting summary is defined by $SC_{Sum}$, where $SC$ is the scoring function ($SU$, $ACL$, $Go$, $R2$, $Av$ and $SAv$) and $Sum$ is the summary type ($Abs$, $Com$ and $Hum$). For example, submission *ACL_abs* learns a scoring function which attempts to approximate similarity of a sentence to the abstract of the document using ACL vectors and cosine to compute similarities.

### 5.5.5 The CL-SciSumm 2018 Results Comparison Against the Other Participants

Ten teams participated in Task 1 with a total of 59 systems submissions and a subset of three teams also participated in Task 2 with a total of 52 systems submissions. The results reported by the organizers (Jaidka et al., 2019) of the top 3 systems for Task 1 can be found at table 5.15. Our system's submission for Voting 3 was the third best system for Task 1. Table 5.16 shows the performance of all our systems for task 1.

Our team won the competition for this year's shared task. Table 5.17 shows the results for task 2 across the best three systems highlighting our system on the top.

| System | Task 1A: Sentence Overlap (F1) | Task 1A: ROUGE F1 | Task 1B |
|---|---|---|---|
| System 6 (Wang et al., 2018) | 0.145 | 0.131 | 0.262 |
| System 2 (Ma et al., 2017) | 0.122 | 0.049 | 0.261 |
| System 11 (Our Systems) | 0.117 | 0.084 | 0.108 |

**Table 5.15:** Top 3 Systems' performance in Task 1A and 1B, ordered by their F1-scores for sentence overlap on Task 1A.

| System | Task 1A: Sentence Overlap (F1) | Task 1A: ROUGE F1 | Task 1B |
|---|---|---|---|
| system 11 Voting 3 | **0.117** | 0.084 | 0.108 |
| system 11 MJ1 | 0.099 | **0.114** | 0.070 |
| system 11 BN1 | 0.089 | 0.110 | 0.064 |
| system 11 0.0001CNN4 | 0.083 | 0.041 | **0.150** |
| system 11 Voting2 | 0.070 | 0.025 | 0.122 |
| system 11 0.1CNN4 | 0.025 | 0.023 | 0.083 |

**Table 5.16:** Our systems' performance in Task 1A and 1B, ordered by their F1-scores for sentence overlap on Task 1A.

| System | Vs. Abstract | | Vs. Human | | Vs. Community | |
|---|---|---|---|---|---|---|
| | R–2 | RSU–4 | R–2 | RSU–4 | R–2 | RSU–4 |
| system 11 (Our System) | **0.329** | **0.172** | **0.149** | **0.090** | **0.241** | **0.171** |
| system 7 (Ma et al., 2018) | 0.217 | 0.142 | 0.114 | 0.042 | 0.158 | 0.115 |
| system 2 (Ma et al., 2017) | 0.215 | 0.115 | 0.138 | 0.074 | 0.220 | 0.151 |

**Table 5.17:** Top 3 Systems' performance in Task 2, ordered by their F1-scores for sentence overlap on Task 1A.

## 5.6 Summary and Conclusions

The CL-SciSumm shared task which has been active for 4 years aimed to encourage research towards scientific paper summarization. The organizers have provided a corpus in a form of clusters, that is; a set of reference and citing papers together with manual annotations indicating for each citance to the reference paper, the facet of this citance and the text span(s) in the reference paper that best represent the citance. They also provided 3 different types of summaries; human, abstract and community.

They asked the participants to submit systems that aims to; identify for each citance in the citing papers (i.e., text spans containing a citation) the cited spans of text in the reference paper that most accurately reflect the citance (Task 1A) and why it has been cited (Task 1B). They also asked them to submit

summaries of the reference papers based on those identified sentences (Task 2).

We participated in the shared task for three consecutive years starting from the year 2016 (the first year of the challenge) and over the years we have learned and changed our approaches to improve our results. Ten teams participated in the first iteration of the shared task with a total number of 23 submissions. We presented a supervised model for Task 1 (J48 algorithm for Task 1A and SMO for Task 1B) and Task 2 (Regression model) with a set of features derived from scientific papers. Unfortunately, for task 1A our system did not perform well, as for Task 1B our system ranked as the third best. Regarding Task 2, our system's performance versus abstract summaries was the third best for both metrics ROUGE-2 and ROUGE-SU4. Moreover, against the human summaries we achieved the second best system for ROUGE-2 and the third best for ROUGE-SU4. Finally, against the community summaries our system was the third best system for ROUGE-2 and the second best for ROUGE-SU4. What we learned after our participation in the Computational Linguistics Scientific Document Summarization 2016 is that modeling Task 1A as a binary classification problem was not a good idea, since there are a lot of sentences that are labeled as not-match and only few sentences have cited by each citation. This modeling of the problem made the data skewed, making the classifier leaning mostly towards predicting non-match for the Reference Paper sentences.

In the second participation of the shared task nine teams participated in Task 1 with a total of 47 systems' submissions and a subset of five teams also participated in Task 2 with a total of 25 systems' submissions. We presented unsupervised and supervised methods to address the tasks proposed by the organizers. We used a voting system for Task 1A among two unsupervised methods: Modified Jaccard and Babelnet embedding cosine similarity. The facet classifier was the same from 2016 participation 5.3.1. It uses a set of manually engineered features informed by our previous work. Our citation-based summarization system is a the same from the previous year with additional word embedding features. Our system for Task 1 performed as the sixth and fifth score for Task 1A and 1B respectively. As for Task 2 over abstract summaries it obtained the second best for ROUGE-2, third best for ROUGE-SU4. As for the human summaries our system ranked fourth for ROUGE-2 but the second best for ROUGE-SU4. Finally, for the communities summary our system was the second best for ROUGE-2 and ROUGE-SU4 as well. We learned that adding more features to Task 1 improved the results over the 2016 participation. However, our Task 1 submission fall out to be within the top 3 systems. Same applies for Task 2 in which by adding ACL and Google embedding to the set of existing features, alongside additional similarity features

we managed to achieve a great score among the rest of the participants.

Finally, ten teams participated in the CL-SciSumm 2018 shared task submitting 59 systems for Task 1, and a subset of three teams submitted 52 systems for Task. For Task 1A, we implemented supervised and unsupervised methods. Our supervised systems are based on Convolutional Neural Networks (CNN), while the unsupervised techniques take advantage of word embedding representations and features computed from the linguistic and semantic analysis of the documents. However, as committing to only one system could result in an under-performing approach, we applied many different system configurations combining them through a voting mechanism. For Task 1B we used the same CNN system of Task 1A where the output was a set of facets. Regarding Task 2, we have developed a neural network based on convolutions to learn a specific scoring function. The CNN model was fed by a combination of word embeddings with sentence relevance and citation features extracted from each document cluster (RP and CPs). The approach was developed and evaluated following the CL-SciSumm Shared Task 2 dataset, our approach outperformed results reported in last year CL-SciSumm-17 Shared Task 2. We also won the competition for that iteration of the challenge. We made the software available for the research community [44,45].

[44]https://github.com/AhmedAbuRaed/CLSciSumm2018
[45]https://github.com/AhmedAbuRaed/CL-SciSumm2017

# Chapter 6

# Generating Related Work Reports through Extractive Summarization

In this chapter we describe our methods for the automatic generation of descriptive related work reports through the extractive summarization of a list of scientific papers. To do so, we identify which parts of the scientific papers are worth extracting, then we generate the related work report in an organized way.

## 6.1 Introduction

By utilizing the citation network we aim to summarize each scientific paper separately by means of extractive summarization and then combine the summarized scientific papers to form the final related work report. Our approach has three stages:

- Scoring the sentences of the scientific papers based on their citation network;
- Selecting sentences from each scientific paper to be mentioned in the related work report;
- Generating an organized related work report by grouping the sentences of the scientific papers that belong to the same topic together;

In order to score the sentences we use both supervised and unsupervised approaches that we implemented in Chapter 5. For the unsupervised learning we use two methods that we implemented during our participation at the second CL-SciSumm challenge (see Section 5.4.1): one is based on a modified variant of Jaccard Similarity, and the other is based on the BabelNet (Navigli

& Ponzetto, 2012a) Embeddings Distance. As for the supervised approach we use several variations based on Convolutional Neural Networks (CNNs) (Zhang et al., 1990) that we implemented during our third participation at the CL-SciSumm challenge (see Section 5.5.2). Once we have scored the sentences we use two methods to decide how many sentences to select from each scientific paper in order to be added in the final related work report: one is based on selecting a unified number of sentences from each scientific paper, and the other is based on a weight applied for each scientific paper based on the number of scientific papers citing it. The final stage is grouping sentences of scientific papers that belong to the same topic together, forming an ordered and organized related work report. We use Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to perform topic modeling in order to identify the topic of each scientific paper.

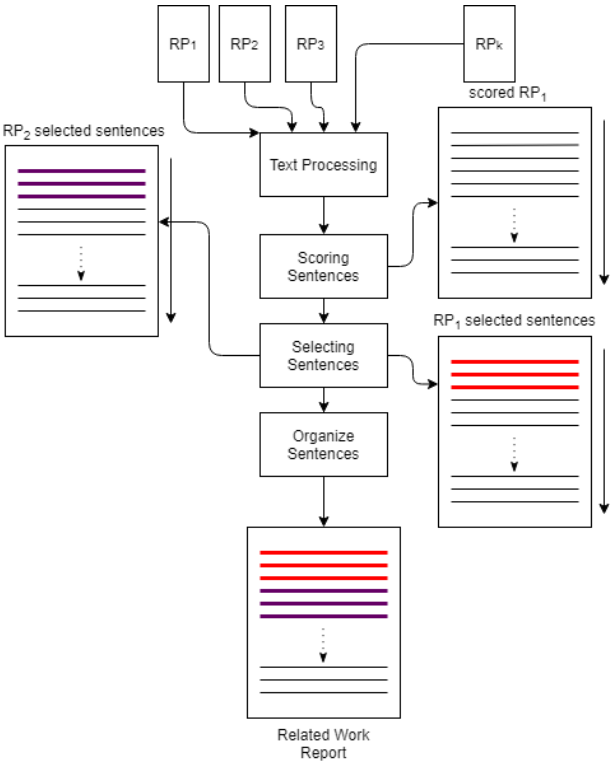


**Figure 6.1:** Method architecture showing a related work report comprises of $k$ number of reference papers.

Figure 6.1 shows our approach's stages and Figure 6.2 shows an example of an already organized related work report formed by a list of scientific papers represented as vector $\vec{RP}$. The related work report has $N$ number of sentences.

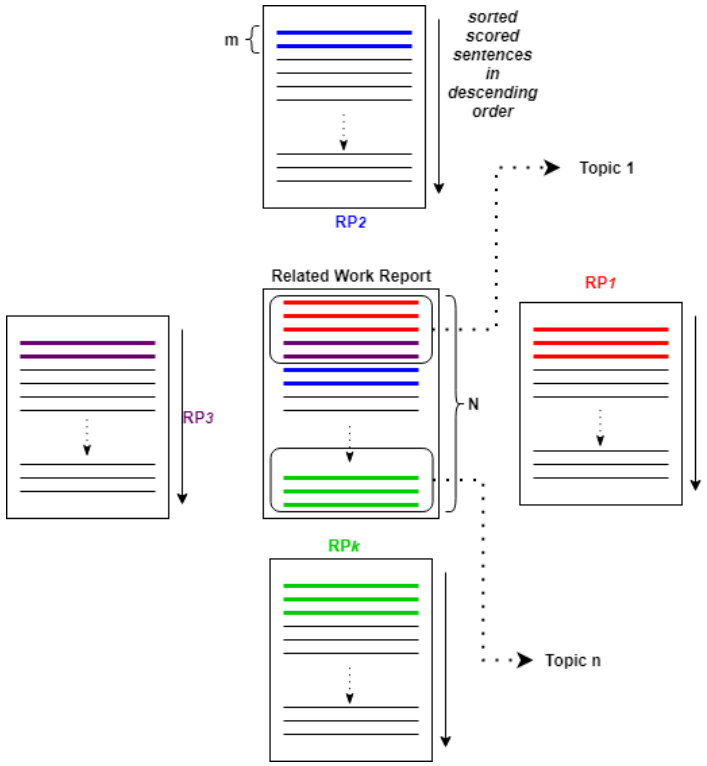


**Figure 6.2:** An example showing how a related work report is formed after applying the three stages.

For each scientific paper to be mentioned in the related work report i.e. reference paper, sentences are sorted in descending order based on their score. The value $m_i$ indicates the number of sentences selected from each reference paper $RP_i$. In this example the system is using a weighted method (based on the number of citing papers) to select sentences from each reference paper, hence the system selected 3 sentences from reference paper 1 ($RP_1$), 2 sentences from reference papers 2 and 3 ($RP_2$ and $RP_3$) and finally 3 sentences from the last reference paper ($RP_k$). The system then groups sentences of reference papers that belong to the same topic together.

In order to generate related work reports through extractive summarization of scientific papers we utilize our multi-level Annotated Corpus of Scientific Papers (AbuRa'ed et al., 2020b) (see Chapter 3). In order to apply the first stage of our approach we use Level 2 of the corpus to score the reference papers' sentences through their connection with the scientific papers citing them in the citation network. Next, to apply stages two and three of our approach to select sentences and generate an organized related work report we use Level

1 of the corpus, in which we try to re-create the related work section of the target paper by summarizing the reference papers mentioned in it. Finally, for evaluation we use the gold related work sections of the target papers provided by the corpus.

The contribution of this chapter is a system for the automatic generation of extractive related work reports together with the automatic and human evaluation of the resulting system. The software is made available for the research community [46].

## 6.2 Scoring sentences of the Reference Papers

Researchers tend to cite the major contributions of a scientific paper. Therefore, by utilizing the citation network of a scientific paper with the papers that are citing it, we could know what those researchers consider an important context in the scientific paper through identifying the sentences that have been cited.

We use both supervised and unsupervised learning approaches to score the sentences of each reference paper taking into consideration its citation network.

### 6.2.1 Unsupervised methods

For our unsupervised learning methods, having a reference paper alongside a set of scientific papers that are citing it, we model a pair of sentences: a citation context (i.e. the sentence that is explicitly mentioning the reference paper) from a citing paper, with each and every sentence in the reference paper, we use the similarity between each pair of sentences to score the reference paper sentence.

As each reference paper is usually cited by multiple citing papers we modeled a final score for each sentence in the reference paper as follows: for each citation context in a citing paper we have calculated the similarities based on one of the two metrics: i.e. modified Jaccard Similarity and BabelNet Embeddings Distance, with each sentence in the reference paper they cite. Afterwards, we have ranked the sentences of the reference paper based on the similarity metric from best to worst. We then compute the final score of the reference paper sentences based on that ranking and a weight we assign for each citing paper. We assign a weight for each citing paper based on the number of scientific papers citing it, the weight applied will provide a

[46] https://github.com/AhmedAbuRaed/RWRG

priority for the citing papers that are considered more valuable by the scientific community.

The equation to score the sentences of the reference paper can be seen below:

$$score(S_j) = \sum_{i=0}^{n} rank_i(S_j) * w_i(CP_i) \tag{6.1}$$

where $n$ is the number of citation context and $rank_i$ is the rank of the reference paper's sentence for the citation context $_i$. $w_i$ is the weight of the citing paper that contains the citation context. In order to normalize all the scores of the reference paper sentences to be between 0 and 1, we have used a normalized value for the ranking of the sentence in which we assign a value of 1 for the sentence with the highest ranking and a value of 0 for the sentences with the lowest ranking. We have also normalized the weight of the citing papers.

Once we have scored the sentences of the reference papers using the unsupervised methods described we sort the sentences in descending order before moving to the next stages. Finally, as fail safe for cases in which there were no citations to a reference paper we set the system to score sentences based on their position in the reference paper, the title and abstract sentences will have the highest scores and the further the sentence is from the title the lower score it gets.

### 6.2.2 Supervised methods

For our supervised learning approach we have access to a training data set that provides three gold summaries for a set of scientific papers. Hence, we use Convolutional Neural Networks (CNNs) to learn the relation between each sentence in the scientific paper and a scoring value indicating its relevance to one of the gold summaries. As described in Section 5.5.2 we train each model using two inputs : a set of context features: six of which are based on the citation network and are dedicated for citation context similarities, and a Word2Vec representation of the reference paper as a second input. The model learns a total of six scores that are derived from the similarity between the reference paper's sentence and each gold summary. Then, we use the trained models to score the sentences of the reference papers in our multi-level corpus.

The organizers of the CL-SciSumm challenge (Chandrasekaran et al., 2019) which addresses the problem of summarizing a scientific paper taking advantage of its citation network, have provided three gold summaries for each reference paper alongside manual annotations stating which sentences in the reference paper have been cited by the citation context of the citing papers, described in Section 5.2. The three types of summaries for each Reference Paper

are: the abstract, the community summary and a human-written summary. We used the same implementation and scoring functions as our third participation in the CL-SciSumm challenge (See Section 5.5.2). After training the CNN models we use the reference papers of the multi-level annotated corpus as a testing data set to score their sentences.

Once we have scored the sentences of the reference papers using the supervised methods described we sort the sentences in descending order before moving to the next stages. Finally, there was no need for a fail safe for cases in which there were no citations to a reference paper since the models are already trained based on the CL-SciSumm challenge data. Hence, during the training process we only rely on the provided summaries not on the citations. However, those citation are presented as context features so cases with no citation would just get zero as a feature value.

## 6.3 Selecting Sentences from the Reference Paper

Scoring the sentences of each reference paper is a good way to measure how important these sentences are in the scope of the scientific paper itself. By selecting the sentences with the highest scores in the scientific paper we could know the major contributions of its author in the related field. However, it would be good to know how a specific reference paper stacks up against the rest of the reference papers, and whether or not we should treat it in the same manner when it comes to the number of sentences to select from it.

We implemented two different systems: one represents equally all the reference papers by selecting a unified number of sentences to represent each reference paper and the other represents each reference paper based on a weight that is specified by the number of citations this reference paper has. We refer to the related work section alongside the list of reference papers to be mentioned in it as a cluster. We create two vectors: $\vec{M}$ a vector mapping each reference paper with the number of sentences to select from it to add in the related work report, and $\vec{C}$ a vector mapping each reference paper with the number of papers citing it. These two vectors are essential to give an insight of the value each reference paper has in the list of reference papers.

We define $N$ as the total number of sentences that the related work report have, $m_i$ is the number of sentences chosen to represent a reference paper $rp_i$ in the related work report in which $m_i \in \vec{M}$ and $rp_i \in \vec{RP}$, $K$ as the total number of reference papers presented in a related work report. In that context $N$ could be referred to as the sentence compression variable that determines how many

sentences will be in the final related work report, This value can be usually defined by the user, once we have identified the importance of each sentence in the reference papers. We can present the related work report by concatenating $m_i$ sentences from each reference paper $rp_i$ (summaries).

Our first system assigns a unified value of $m_i$ for each reference paper, such value is based on $N$ and $K$. For example: if we need to generate a related work report with 20 sentences: $N$ = 20 (the entire report should have 20 sentences from the list of reference papers) and we have 5 reference papers to present ($K$ = 5), then we can assign the value of $m_i$ to be 4 for each reference paper forming $\vec{M} = [4 \quad 4 \quad 4 \quad 4 \quad 4]$, hence $m = N$ / $K$.

On the other hand, not all reference papers have an equal importance, therefore we decided to also run a second system that assigns the value of $m_i$ based on a specific weight. First, in order to guarantee that at least one sentence is selected from each reference paper we first assign a value of one for each $m_i \in \vec{M}$ then distribute the rest of the required sentences based on the weight. The minimum value of $m_i$ is 1 and the maximum value of $m_i$ is identified as: $N-K$ for each reference paper. Example: Assuming we have $K = 5$ and $N = 10$ (the entire report should have 10 sentences from all reference papers). An extreme case would be to take only one sentence from each reference paper. Then the rest of the sentences left will be $10-5 = 5$ sentences to be taken by one of the reference papers. In that context, we assign the weight based on the number of papers citing each reference paper, then we calculate $m_i$ based on that weight. The algorithm works as follow:

Algorithm 1 initially assigns $m_i$ a value of 1 for all reference papers. Afterwards, if there are still more sentences to be presented it starts assigning more sentences for the reference papers based on their weight till all the number of sentences required (value of $N$) is satisfied. Since in the previous stage we have sorted the sentences in descending order based on the score, when selecting sentences from a reference paper we select from the top of the sorted list of sentences.

## 6.4 Generating the Related Work Report

Once we have identified how many sentences to collect from each reference paper to be represented in the related work report, we group the sentences of the reference papers that share the same topic together. An author usually starts with a certain related topic and then moves onward stating each and every reference paper related to that specific topic before moving to the next topic.

**Algorithm 1: Calculating the values of $\vec{M}$**

**Input:** $N$; $\vec{C}$ ;
**Output:** $\vec{M}$
$required = N$ ;
**for** $i \leftarrow 0$ **to** $K-1$ **do**
    $m_i \in \vec{M} = 1$;
    $required = required - 1$;
**if** $required > 0$ **then**
    $leftSentences = required$;
    $totalNumCitingPapers = \sum_{n=0}^{K-1} c_n \in \vec{C}$;
    **while** $required > 0$ **do**
        **for** $i \leftarrow 0$ **to** $K-1$ **do**
            $proportion = \left\lfloor \left( \left\lfloor \frac{100 \times c_i}{totalNumCitingPapers} \right\rfloor \times 0.01 \right) \times leftSentences \right\rfloor$;
            $m_i \in \vec{M} += proportion$;
            $required -= proportion$;
            **if** $required <= 0$ **then**
                break;

The intuition is to group the sentences of reference papers that share the same topic together. Therefore, to find the topics across the reference papers we used Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and modeled each reference paper based on its Title and Abstract. In order to find the optimal number of topics to train the LDA model on, we build many LDA models based on different numbers of topics (*numT*). Then, we select the one that gives the highest coherence value (Mimno et al., 2011) or until the coherence value converges. By choosing a '*numT*' that marks the end of a rapid growth of a topic coherence usually offers meaningful and interpretative topics, while picking an even higher value can sometimes provide more granular sub-topics. Once we identify the LDA model with the ideal number of topics to train on (*numT*), we use it to identify the topics to which each reference paper belongs. We choose the topic with the highest probability as the representative of a reference paper's topic, therefore we assign each reference paper to only one topic. We define the topics vector that maps a reference paper to its topic as : $\vec{T}$ in which each reference paper belongs to one topic $t_i \in \vec{T}$.

We organize the sentences based on the reference papers' topics. For each topic $t_i \in \vec{T}$ we group the sentences of its reference papers together before

moving to the next topic. For each reference paper we select $m_i \in \vec{M}$ sentences and add them in the right order. We start with $rp_i \in \vec{RP}$ such that $i = argmax(\vec{M})$ then we pick the topic $t_i \in \vec{T}$ of this $rp_i \in \vec{RP}$ to start with, and group that topic's reference papers together before moving to the next topic and repeat the same process until there are no more sentences to add to the related work report.

## 6.5 Experiments

For our experiments we aimed to create related work reports from a list of reference papers. We have used the Multi-level Annotated Corpus of Scientific Papers as our main data set (testing data) and our goal was to recreate the related work section (report) of the target paper for each cluster provided. We implemented the there stages sequentially for all our systems. We compared our systems against a set of baselines: SUMMA centroid (Saggion, 2008b), MEAD (Radev et al., 2004), TextRank (Mihalcea & Tarau, 2004b) and LexRank (Erkan & Radev, 2004). We performed both automatic and human evaluation by comparing the systems to the gold related work sections of the target paper in the Multi-level Annotated Corpus of Scientific Papers.

### 6.5.1 Baselines

For our experiments we implemented several extractive summarization baselines alongside a set of simple baselines based on the observations arising from the analysis of citation sentences and scientific abstracts on the use of titles and abstracts (Jaidka et al., 2013; Saggion, 1999). The *title* baseline is to use the title of each cited article as citation sentences. The *abstract first* baseline uses as citation sentences the first sentence of the abstract of the cited articles while the *abstract last* baseline uses the last sentence.

The second set of baselines is composed of available systems that use well-established extractive techniques. We have made sure that all the baselines have the same conditions as our systems. That is we fed each and every scientific paper to the baseline and guaranteed that at least the system will select one sentence from each. We also instructed the system to generate the same number of sentences as the gold related work sections ($N$ system = $N$ gold). We describe the systems as follows:

- *MEAD* (Radev et al., 2004) is a well-known extractive document summarizer which generates summaries using centroids alongside other fea-

tures such as the position of the sentence and the length.

- *TextRank* (Mihalcea & Tarau, 2004b) and *LexRank* (Erkan & Radev, 2004) are both extractive and unsupervised graph-based text summarization systems which create sentence graphs in order to compute centrality values for each sentence. Both algorithms have similar underlying methods to compute centrality which are based on the PageRank ranking algorithm. They differ in how links are weighted in the document graph.

- *SUMMA* (Saggion, 2008b) is a Java implementation of several sentence scoring functions. We use the implementation of the centroid scoring functionality to select the most central sentence in a document.

## 6.6 Results, Evaluation and Discussion

In this section we compare our systems against the baseline systems for the task of automatic generation of descriptive related work reports. We have performed both automatic and human evaluation. For automatic evaluation we have used 4 ROUGE metrics: ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4. We only present here ROUGE-1 and ROUGE-2 metrics for all the systems except for the CNN approach for which we only present the top five systems. For the complete list of ROUGE metrics alongside all the systems of the CNN approach please see Table C.1 and Table C.2 in the appendix. ROUGE measures combine precision and recall in a harmonic F-measure which is generally used to assess the systems' performance. The results of ROUGE-1 and ROUGE-2 metrics can be found at Table 6.1. MBL: refers to the unified number of sentences to be selected from each reference paper ($N$ / $K$).

The non-informed extractive baselines which do not perform any analysis of the input (e.g. use of titles or sentences from abstracts) tend to have a high precision but low recall, especially precise is the title. In addition, the results show that using a specific weight for each reference paper or using the same number of sentences when selecting the sentences from each reference paper leads to similar results. It can be noticed that selecting a unified number of sentences from each reference paper has performed slightly better than the system that uses weights to decide how many sentences to select.

Moreover, we also ran human evaluation on 10 clusters, we manually selected 10 clusters that discusses different varieties of topics, each cluster was evaluated by three evaluators who are experts in Natural Language Processing

| | ROUGE-1 | | | ROUGE-2 | | |
|---|---|---|---|---|---|---|
| **SYSTEM** | **R** | **P** | **F** | **R** | **P** | **F** |
| Titles | 0.074 | **0.375** | 0.119 | 0.013 | 0.072 | 0.022 |
| AbsFS | 0.126 | 0.272 | 0.155 | 0.019 | 0.041 | 0.023 |
| AbsLS | 0.114 | 0.263 | 0.150 | 0.013 | 0.035 | 0.018 |
| SUMMA | 0.293 | 0.103 | 0.154 | 0.102 | 0.027 | 0.047 |
| MEAD | 0.361 | 0.137 | 0.205 | 0.118 | 0.025 | 0.049 |
| LexRank | 0.312 | 0.228 | 0.259 | 0.107 | 0.060 | 0.076 |
| TexRank | 0.367 | 0.116 | 0.186 | 0.117 | 0.017 | 0.040 |
| Babelnet | 0.387 | 0.231 | 0.286 | 0.152 | 0.084 | 0.107 |
| MJ | 0.336 | 0.266 | 0.292 | 0.151 | 0.114 | 0.127 |
| $CNN_{ROUGE-2-abstract}$ | 0.303 | 0.285 | 0.290 | 0.137 | **0.122** | 0.127 |
| $CNN_{AvgGAR-abstract}$ | 0.340 | 0.265 | 0.294 | 0.141 | 0.105 | 0.118 |
| $CNN_{AvgSGAR-abstract}$ | 0.344 | 0.272 | 0.300 | 0.139 | 0.106 | 0.119 |
| $CNN_{ROUGE-2-community}$ | 0.335 | 0.270 | 0.296 | 0.143 | 0.107 | 0.121 |
| $CNN_{ROUGE-2-human}$ | 0.310 | 0.281 | 0.291 | 0.143 | 0.121 | 0.129 |
| Babelnet (MBL) | **0.409** | 0.242 | 0.299 | **0.158** | 0.087 | 0.110 |
| MJ (MBL) | 0.350 | 0.270 | 0.299 | 0.154 | 0.112 | 0.127 |
| $CNN_{ROUGE-2-abstract(MBL)}$ | 0.322 | 0.291 | 0.302 | 0.142 | 0.120 | 0.128 |
| $CNN_{AvgGAR-abstract(MBL)}$ | 0.360 | 0.273 | 0.307 | 0.148 | 0.108 | 0.124 |
| $CNN_{AvgSGAR-abstract(MBL)}$ | 0.359 | 0.278 | **0.310** | 0.149 | 0.110 | 0.125 |
| $CNN_{ROUGE-2-community(MBL)}$ | 0.353 | 0.275 | 0.305 | 0.152 | 0.111 | 0.126 |
| $CNN_{ROUGE-2-human(MBL)}$ | 0.319 | 0.285 | 0.298 | 0.145 | 0.120 | **0.130** |

**Table 6.1:** Automatic evaluation results of our systems against the baselines for ROUGE-1 and ROUGE-2 metrics. Only the top 5 systems of the CNN approach are shown.

(NLP). The evaluators are mostly Europeans with the age group between 25-34 with an expert level of English language, they have a range between good and very good in their expertise in Natural Language Processing. Table 6.2 and Table 6.3 are two examples of our system's generation of a related work report one with and one without topic modeling applied. It can be noticed from the generated summaries that some sentences are using first pronouns e.g. "We" referring to the authors of the reference paper. This limitation of the approach can be solved through using passive sentences which we did not cover in this thesis. Also abstractive summarization is considered a solution for this problem since it learns sequence generation rather than copying already existing sentences from the reference paper. We have implemented a sequence-to-sequence system in the next chapter (See Chapter 7).

The objective of this evaluation is to assess the appropriateness of four different related work sections for a given target paper in the test data set i.e. Multi-level corpus. The four related work sections represent: the best system of the

| |
|---|
| (Blunsom et al. 2007) Statistical machine translation (SMT) has seen a resurgence in popularity in recent years ... (Kumar and Byrne 2004) We also show how MBR decoding can be used to incorporate syntactic structure into a statistical MT system ... template model for statistical machine translation. (Matsusaki et al. 2005) This paper defines a generative probabilistic model of parse trees, which we call PCFG-LA. This paper defines a generative model of parse trees that we call PCFG with latent annotations (PCFG-LA). (May and Knight 2006) We also demonstrate our algorithm's effectiveness ... to deal with grammars that produce trees. (Petrov et al. 2006) In this paper, we investigate the learning of a grammar consistent with a treebank at ... likelihood of the training trees. We present a method that combines the strengths of both manual and automatic approaches while addressing some of their common shortcomings. (Tromble et al. 2008) In this paper we explore a different strategy to perform MBR decoding over Translation Lattices ... that compactly encode a huge number of translation ... We begin with a review of MBR decoding for Statistical Machine Translation (SMT). |

**Table 6.2:** An example of a summary generated by the our system without topic modeling applied.

| |
|---|
| (Blunsom et al. 2007) Statistical machine translation (SMT) has seen a resurgence in popularity in recent years ... (Kumar and Byrne 2004) We also show how MBR decoding can be used to incorporate syntactic structure into a statistical MT system ... template model for statistical machine translation. (Tromble et al. 2008) In this paper we explore a different strategy to perform MBR decoding over Translation Lattices ... that compactly encode a huge number of translation ... We begin with a review of MBR decoding for Statistical Machine Translation (SMT). (Matsusaki et al. 2005) This paper defines a generative probabilistic model of parse trees, which we call PCFG-LA. This paper defines a generative model of parse trees that we call PCFG with latent annotations (PCFG-LA). (May and Knight 2006) We also demonstrate our algorithm's effectiveness ... to deal with grammars that produce trees. (Petrov et al. 2006) In this paper, we investigate the learning of a grammar consistent with a treebank at ... likelihood of the training trees. We present a method that combines the strengths of both manual and automatic approaches while addressing some of their common shortcomings. |

**Table 6.3:** An example of a summary generated by the our system with topic modeling applied.

baselines i.e. LexRank, the gold related work section and the best system we have with and without topic modeling applied i.e. $CNN_{AvgSGAR-abstract(MBL)}$.

To carry out the evaluation we prepared each reference paper's Title, Abstract and Introduction in PDF format. Alongside the scientific paper we provided in a random order the related work sections in text format to be evaluated. We also added a folder with the references that are mentioned in the related work section, we also provided the bibliographic information about each of the references which will be cited in the related work section. Given the target scientific paper Title, Abstract and Introduction section alongside a related work section, we asked them for their opinion on three fronts:

- Responsiveness: How good do you consider the related work section given that it must include information on the list of reference papers and must fit in the target paper.

- Linguistic quality: How do you rate the readability and grammaticality of the related work section? That is: is it understandable? is it grammatically correct (are the sentences correct)? Are there any spelling mistakes? Is punctuation appropriate?

- Text organization: How well organized and coherent the related work section is? That is: does the discourse (topics) flows from sentences to sentence? Are the sentences organized in a coherent way? Is the text not redundant?

We instructed them to read the target scientific paper's Title, Abstract and Introduction (the pdf file), and then to read each related work section (the text file). Once they had finished reading the related work section we informed them to fill the evaluation form indicating the scores of each metric. Finally, we requested that they should not check the web for a related work section or the target paper to avoid influence from external variables and use the references folder if they felt they had to.

Table 6.4 present the average of all the metrics across the 10 clusters for our system with and without topic modeling applied, LexRank: the best baseline in the automatic evaluation and finally the gold related work report. Finally, for the sake of completeness we provide a more detailed list for each cluster with the average across all the three evaluators for each metric. What can be noticed is that our system with topic modeling super-passes the baseline in all metrics and it is considered an improvement over not implementing topic modeling for our system.

| System | Responsiveness | Linguistic quality | Text organization |
|---|---|---|---|
| Gold | 4.5 | 4.4 | 4.3 |
| LexRank | 2.4 | 2.5 | 2.0 |
| WithoutTM | 2.9 | 3.3 | 2.3 |
| TopicModeling | 3.1 | 3.6 | 2.5 |

**Table 6.4:** The results of the Human Evaluation over our system with and without applying topic modeling against the LexRank baseline.

## 6.7 Summary

We presented a state of the art system for automatic generation of descriptive related work reports through extractive summarization of a list of scientific papers to be mentioned in the related work report. The system has three sequential stages: scoring the sentences of the reference papers, selecting the top sentences from those papers and finally, generating an organized related work report.

In order to score the sentences, the system applies both supervised and unsupervised methods that we have implemented during our participation in the CL-SciSumm challenge. As for selecting the number of sentences that represents each scientific paper we applied two methods: selecting a unified number of sentences and selecting a number of sentences based on a weight assigned for the scientific papers derived from the number of scientific papers citing them. Finally, we applied topic modelling to control the flow of sentences in an organised way, in which sentences of the scientific papers that belong to the same topic are presented together. Finally, we have evaluated our systems against a set of baselines using both automatic and human evaluation.

# Chapter 7

# Generating Related Work Reports through Abstractive Summarization

In this chapter we describe our methods in automatic generation of descriptive related work reports by using abstractive summarization of scientific papers. We create a neural sequence learning process which produces citation sentences to be included in a related work section of an article. We train the neural architecture using an available set of scientific data of citation sentences and we test our models over a data set of related work sections; we also compare the performance to a set of baseline extractive summarizers , an abstractive summarizer and a Convolutional Neural Network (CNN) state of the art approach. Our quantitative results based on available evaluation metrics are promising.

## 7.1 Introduction

While in chapter 6 we present a system to generate descriptive related work reports by extractively summarizing a list of scientific papers, recently there has been a growing interest in abstractive text summarization (Chopra et al., 2016; Nallapati et al., 2016b; Rush et al., 2015; Zeng et al., 2016). Abstractive text summarization is the task of generating a short summary consisting of a few sentences that captures the main ideas of an article by mapping an input sequence of words in a source document to a target sequence of words i.e. sequence-to-sequence models. Sequence-to-sequence models have taken advantage of deep learning networks and got more complicated over time to include longer documents as an input. In this context, we decided to compare our extractive method with a non-extractive one (abstractive).

However, since abstractive summarization does not perform well for extra long documents we changed the nature of our experiments, in this chapter we summarize each scientific paper using only its title and abstract rather than the entire text. We are aware that this will make this approach not directly comparable with the extracting one; which uses the full scientific paper to extract the summaries, but we believe that it will provide an insight of the capabilities of the sequence-to-sequence models for this kind of tasks even with its known limitations: handeling short text, repeated words and missing vocabulary.

Taking advantage of an available data set for scientific summarization composed of research articles, citation sentences, and human summaries we train a sequence-to-sequence model to simulate the generation of citation sentences. We concatenate citation sentences automatically generated from each cited paper to produce a novel related work section which we evaluate by comparing the generated texts to the gold related work section using content-based evaluation metrics. The comparison is carried out with our abstractive approach, several baselines, unsupervised summarizers, and an extractive state of the art neural networks approach.

To model our generative approach, we make use of pointer–generator neural networks (Vinyals et al., 2015a) which are sequence-to-sequence models that produce an output sequence consisting of elements from the input sequence. We use the pointer–generator networks with two Neural Networks (NN) architectures which have recently achieved good performance in complicated tasks; Transformer (Vaswani et al., 2017), that uses stacked self-attention and point-wise fully connected layers for both the encoder and decoder, and Bi-Directional RNNs. More specifically, in (Kalchbrenner & Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014) it is introduced as a variation of RNNs called sequence-to-sequence (seq2seq) learning which uses recurrent neural networks to map variable-length input sequences to variable-length output sequences. While relatively new, the sequence-to-sequence approach has achieved state-of-the-art results in not only its original application – machine translation – (Luong & Manning, 2015; Jean et al., 2014; Luong et al., 2015; Jean et al., 2015; Luong et al., 2014), but also in image caption generation (Vinyals et al., 2015b), and text summarization (Nallapati et al., 2016b).

Sequence-to-sequence learning aims to indirectly model the conditional probability $p(y|x)$ of mapping an input sequence, $x = x_1, ..., x_n$, into an output sequence, $y = y_1, ..., y_m$ accomplishing such goal through the encoder-decoder framework proposed by (Sutskever et al., 2014; Cho et al., 2014). We use sequence-to-sequence architecture to generate each citation sentence to be included in the related work section from an input sequence which is composed of a title and an abstract of a scientific paper that is being cited. To directly

tackle the problem of producing a related work section, we will use a gold-standard data set of related work sections and their cited papers to test our approach. We will feed our model with a set of sentences from the cited papers and accumulate the generated citation sentences to produce a related work section.

The contributions of this chapter are the following:

- The design and evaluation of a related work reports generation system using abstractive summarization of scientific papers;
- A new data set of over 15K pairs of articles and citation sentences to train sequence-to-sequence models;
- A comparison with state-of-the-art methods showing the potential of the approach;
- The data, the software and instructions on how to reproduce our work are available for the community [47].

## 7.2 Data

We make use of two different types of data: a data set of scientific papers and their citation sentences that we use to train our citation sentence generation model, and we used our data set from Chapter 3 as a gold-standard data set of related work sections and their cited papers to test the whole process. Additionally, we study the effect of a filter over the data sets in order to select sentences which explicitly indicate the author's work.

### 7.2.1 Training Datasets

We make use of the data available in the ScisummNet Corpus (Yasunaga et al., 2019b,c). This corpus has been released by Yale LILY lab and expanded from the CL-Scisumm project (Mayr et al., 2019; Jaidka et al., 2014b). This dataset provides over 1,000 papers of the Association for Computational Linguistics (ACL) anthology network (Bird et al., 2008) with their citation networks (e.g. citation sentences, citation counts) and their author abstracts. Additionally, we collect data similar to ScisumNet but from Open Academic Graph (OAG) and Microsoft Academic Graph (MAG) (Sinha et al., 2015; Tang et al., 2008). MAG is a diverse graph containing scientific publication records, citation relationships between those publications, as well as authors, institutions, journals,

[47] https://github.com/AhmedAbuRaed/SPSeq2Seq

conferences, and fields of study. OAG is a large knowledge graph unifying two billion records both academic graphs: Microsoft Academic Graph (MAG) and AMiner (Tang, 2016). We used the available OAG dumps to gain access to the list of all paper IDs at MAG. Afterward, we used Microsoft Cognitive Services Academic Knowledge API to access MAG nodes. The obtained papers were kept if and only if: (i) MAG contained an abstract for the paper and (ii) MAG contained at least one of the papers being cited. The references of the stored papers were extracted iteratively to obtain more data. All the collected data, that is available for the community[48], has been indexed for efficient processing. The collected data amounts to: 940 pairs from ScisummNet Corpus and 15,574 pairs from our new dataset. In summary, our data from these two sources consists of pairs of input and output sequences as follows:

$$< T_i \oplus A_i, C_i > \tag{7.1}$$

Where the $i-$th input sequence is a concatenation ($\oplus$) of a scientific paper's title $T_i$ and abstract $A_i$, as for the output sequence we use the citation sentence $C_i$ used by the citing scientific paper.

For further analysis, we also applied a filter on the same data which selects sentences from the abstract that are directly related to the scientific paper author or presentation. The filter is based on Teufel's (Teufel, 2000) first pronoun (e.g. we, our and my) and presentation nouns (e.g. this paper, study and article) gazetteers. The filter is only applied to the abstract sentences (the title is never removed). The resulting sentences from the filter process are the title and abstracts' sentences that contain any of the first pronoun and presentation nouns. This process will exclude any sentences that do not explicitly mention the authors nor the presented work directly. An example of the data used for training the citation sentence generator is shown in Figure 7.1. In the example[49], the citation sentence contains some literal (e.g. "negative evidence from edited textual corpora") and non-literal (e.g. "high precision" instead of "80% precision" or "checkers" instead of "detecting grammatical errors") elements extracted from title and abstract of the cited work. From the set of citation sentences available for each paper we use the one that is most similar

[48]The dataset can be accessed though this link: https://github.com/AhmedAbuRaed/SPSeq2Seq

[49]Cited paper: Martin Chodorow and Claudia Leacock. 2000. An unsupervised method for detecting grammatical errors. In Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference (NAACL 2000). Association for Computational Linguistics, Stroudsburg, PA, USA, 140-147. Citing paper: Chung-Chi Huang, Mei-Hua Chen, Shih-Ting Huang, Jason S. Chang. EdIt: A Broad-Coverage Grammar Checker Using Pattern Grammar. Proceedings of the ACL-HLT 2011 System Demonstrations.

(closest) to title and abstract in terms of the BLEU score measure (Papineni et al., 2002) used to compare a target and source translations.

| | |
|---|---|
| title | *An* **Unsupervised** *Method For* **Detecting Grammatical Errors** |
| abstract | We present an **unsupervised** method for **detecting grammatical errors** by inferring **negative evidence from edited textual corpora**. The system was developed and tested using essay-length responses ... The error-recognition system, ALEK, performs with about **80% precision** and **20% recall**. |
| cit. sent. | Among **unsupervised checkers**, Chodorow and Leacock (2000) exploits **negative evidence from edited textual corpora** achieving **high precision** but **low recall.** |

**Figure 7.1:** Example of a scientific article (title $\oplus$ (non-filtered) abstract) and a citation sentence. Similar phrases have been highlighted.

A similar example showing the filtered version of the previous example can be seen in Figure 7.2. We can notice that after applying the filter process most of the shared phrases are still present.

| | |
|---|---|
| title | *An* **Unsupervised** *Method For* **Detecting Grammatical Errors** |
| abstract | We present an **unsupervised** method for **detecting grammatical errors** by inferring **negative evidence from edited textual corpora**. |
| cit. sent. | Among **unsupervised checkers**, Chodorow and Leacock (2000) exploits **negative evidence from edited textual corpora** achieving high precision but low recall. |

**Figure 7.2:** Example of a Filtered scientific article (title $\oplus$ filtered abstract) and a citation sentence. Similar phrases have been highlighted.

### 7.2.2 Testing Data set

In order to test our approach, we make use of our data set (previously described in Chapter 3) used for related work generation. See Figure 7.3 which represents a segment of a related work section for a scientific paper in the

corpus [50] that is citing three different scientific papers [51] [52] [53].

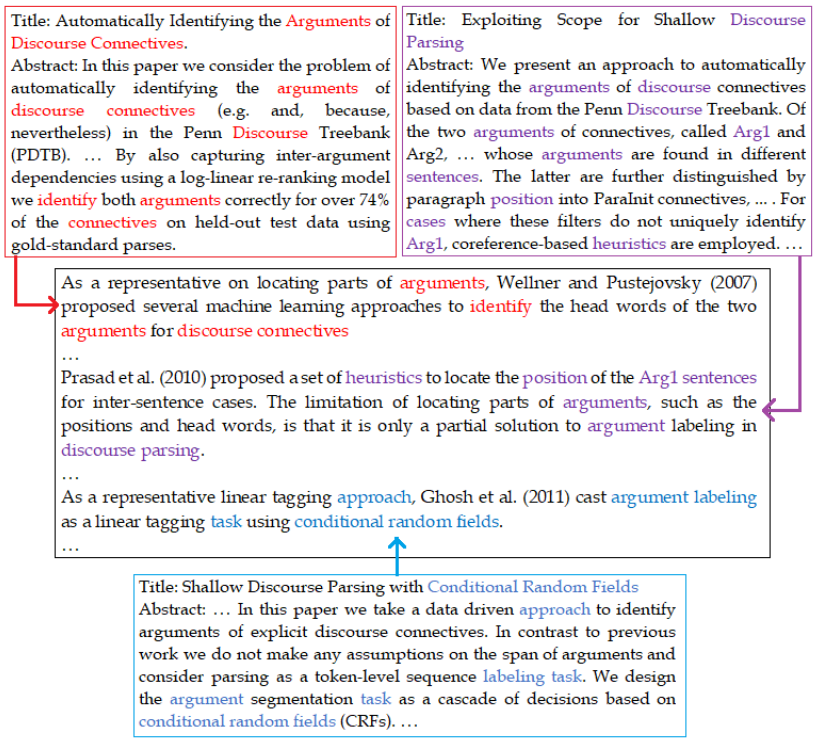


**Figure 7.3:** Example of a scientific paper citing three other scientific papers.

We already have access to each citation sentence manually annotated in the related work section of the target scientific paper linking it with its cited paper. Finally, the same filtering process applied on the training data set was applied for the testing data set.

[50] Kong, Fang, Hwee Tou Ng, and Guodong Zhou. "A constituent-based approach to argument labeling with joint inference in discourse parsing." In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 68-77. 2014.

[51] Ben Wellner and James Pustejovsky. 2007. Automatically identifying the arguments of discourse connectives. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 92–101.

[52] Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2005. Attribution and the (non-)alignment of syntactic and discourse arguments of connectives. In Proceedings of the Workshop on Frontiers in Corpus Annotation II: Pie in the Sky, pages 29–36

[53] Sucheta Ghosh, Richard Johansson, Giuseppe Riccardi, and Sara Tonelli. 2011. Shallow discourse parsing with conditional random fields. In Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 1071–1079.

## 7.3 Methodology

Our approach is based on pointer–generator neural networks with copy-attention technique and coverage mechanism (See et al., 2017; Wu et al., 2016). Copy-based generation can copy words from the source text via pointing, which aids accurate reproduction of information while retaining the ability to produce novel words through the generator. As for coverage (Wu et al., 2016), it is a mechanism to keep track of what has been summarized discouraging repetition by forcing penalties on repeated text therefore controlling redundancy of the generated output.

We utilize pointer–generator neural networks with two different architectures; Bidirectional Recurrent Neural Network (BRNN) (Schuster & Paliwal, 1997) which maps a source sequence to a target sequence, and Transformers (Vaswani et al., 2017); where the closest model to the one we use is so-called Copy-Transformer proposed in (Gehrmann et al., 2018). See Figure 7.4 which shows the pointer–generator neural network used with the BRNN architecture. For each decoder time-step a generation probability $P_{gen} \in [0,1]$ is calculated, which weights the probability of generating words from the vocabulary, versus copying words from the source text. The vocabulary distribution and the attention distribution are weighted and summed to obtain the final distribution, from which we make our prediction. The figure presents an example of a scientific paper at the input text [54] being cited by another scientific paper [55] and the network is trying to generate the next token for the citation context (summary) in which the next token to be generated by the decoder is "speech" which has the highest attention in the attention distribution.

Sequence-to-sequence models are particularly good at translation, where the sequence of words from one language is transformed into a sequence of different words in another language. However, summarization can, in certain cases, be casted as sequence-to-sequence modeling to summarize a long source into a shorter one in the same language to form the final output summary. We use BRNN (Schuster & Paliwal, 1997) which is a natural generalization of feed-forward neural networks where the source sequence tokens are fed one-by-one into a single-layer of a bidirectional LSTM (encoder), producing a sequence of encoder hidden states $h_i$. On each step $t$, a single-layer of a unidirectional

[54] Cited paper: Brill, Eric. "A simple rule-based part of speech tagger." In Proceedings of the third conference on Applied natural language processing, pp. 152-155. Association for Computational Linguistics, 1992

[55] Citing Paper: Modi, Deepa, and Neeta Nain. "Part-of-Speech Tagging of Hindi Corpus Using Rule-Based Method." In Proceedings of the International Conference on Recent Cognizance in Wireless Communication and Image Processing, pp. 241-247. Springer, New Delhi, 2016.

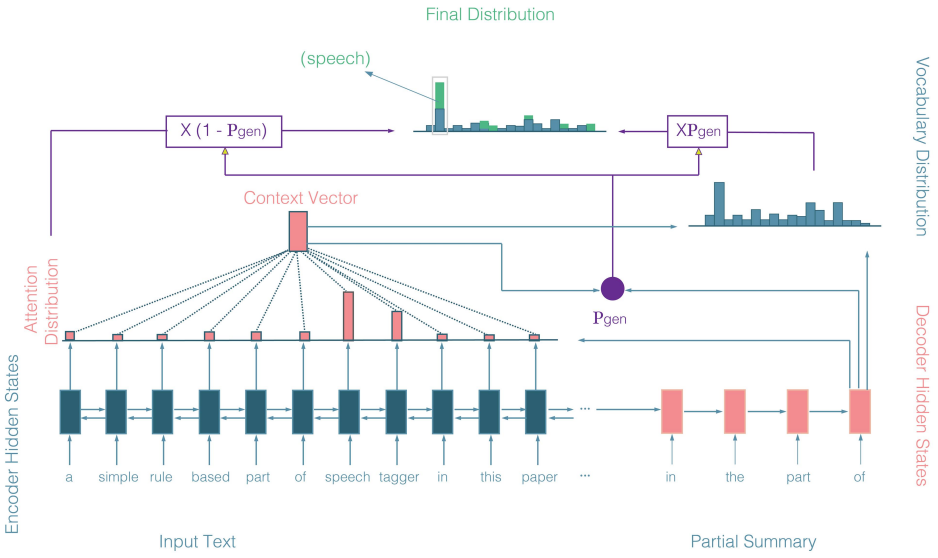


**Figure 7.4:** The pointer–generator architecture.

LSTM receives the word embedding of the previous word (while training, this is the previous word of the reference summary; at test time it is the previous word emitted by the decoder), and has decoder state $st$. We also applied *the transformer* (Vaswani et al., 2017) - encoder–decoder–based architecture - for "translating" one sequence into another one as a basis. This architecture uses stacked self-attention and point-wise fully connected layers for both the encoder and decoder. See the model architecture at Figure 7.5 which we use with the pointer–generator neural network separately replacing the BRNN architecture. The encoder is composed of a stack of N identical layers. Each layer has two sub-layers. The first is a multi-head, self-attention mechanism, and the second is a simple, position wise fully connected feed-forward network. The decoder is also composed of a stack of N identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. The transformer uses a self-attention layer by adding a mechanism called "multi-headed" attention expanding the model's ability to focus on different positions of the input, giving the attention layer multiple "representation subspaces" for the weight matrices, and allowing selection of important parts of the sequence at each step to adjust the distribution over the vocabulary which is essential while summarizing. We rely on the Neural Machine Translation (NMT) tool

OpenNMT-py (Klein et al., 2017) to implement our abstractive models. OpenNMT is an open source initiative for NMT and neural sequence modeling. It is a general-purpose attention-based sequence-to-sequence system that also implements the latest state-of-the-art sequence-to-sequence techniques.

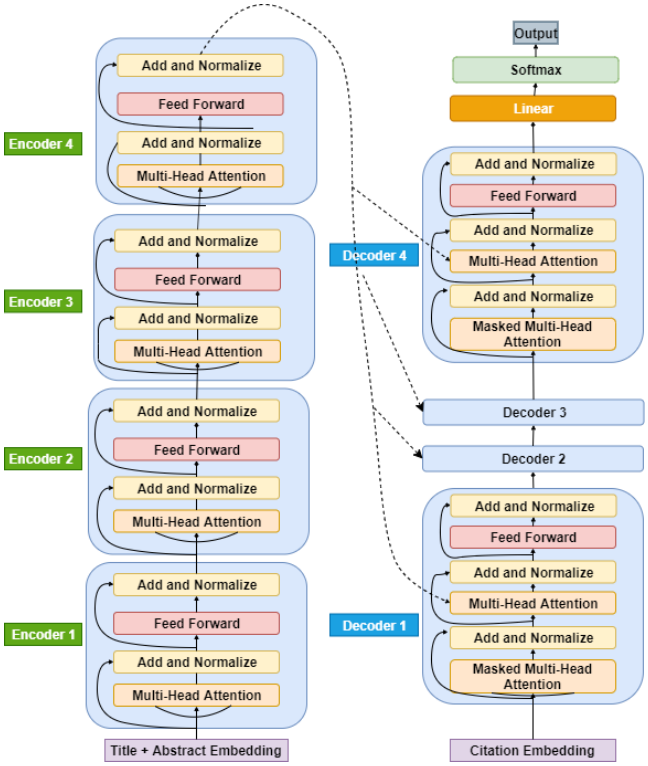


**Figure 7.5:** The Transformer model architecture: encoder to the left and decoder to the right.

## 7.4 Experiments

In order to compare our approach, we implemented several baselines over the RWSData. Alongside, we ran several experiments to generate abstractive summaries for each cluster i.e. a related work section for a target paper.

### 7.4.1 Baselines

For our experiments we implemented the same baselines that we used in Chapter 6 with $SEQ^3$ (Baziotis et al., 2019), an unsupervised abstractive method

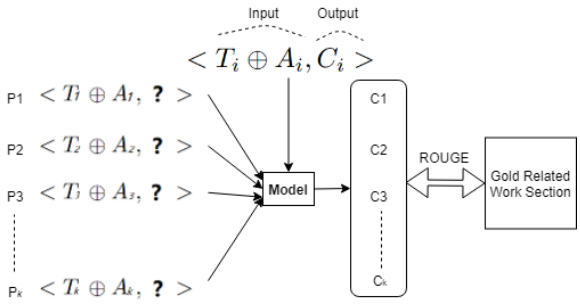


**Figure 7.6:** Generation of related work sections from a set of papers ($P_1 ... P_n$) and evaluation. Model represents any of the sentence extraction/generation systems tested in this work. Output citation sentences ($C_i$) are concatenated and compared to a gold standard related work section

.

which uses a sequence-to-sequence-to-sequence auto-encoder, as an additional baseline.

For the *title*, *abstract first* and *abstract last* baselines there was no need to change anything. As for the rest of the baselines i.e. *MEAD* (Radev et al., 2004), *TextRank* (Mihalcea & Tarau, 2004b), *LexRank* (Erkan & Radev, 2004), *SUMMA* (Saggion, 2008a) and *SEQ*$^3$ (Baziotis et al., 2019). We gave those baselines the title and abstract of each reference paper as an input and generated one sentence as an output.

## 7.4.2 Extracting Sentences with a Convolutional Neural Network

This system, which is based on a neural network architecture which achieved state of the art performance in the Sci-Summ 2018 Challenge (Abura'ed et al., 2018; Mayr et al., 2019), takes advantage of the potential of convolutions to abstract higher level features from sentences in order to learn its relevance in a specific document (Abura'ed et al., 2017, 2018). This relevance is based on the relationship between a set of features extracted and computed for each sentence and the scoring function. The system assigns a score between 0 (not relevant) and 1 (highly relevant).

### 7.4.2.1 Extraction of Sentence Features

The set of sentence features is organized into two inputs to feed the system. First, we transformed each word from a sentence into a vector by looking up word embeddings. In this scenario, we used two pre-trained word embed-

dings, which were concatenated: the Google News embeddings[56] (three million words in 300 dimensional vectors trained using word2vec (Mikolov et al., 2013a) over a news text corpus of 100 billion words) and the Association for Computational Linguistics (ACL) Anthology Reference Corpus embeddings (Liu, 2017) (300 dimensional vector trained over a corpus of ACL papers (Bird et al., 2008)). This embedding matrix representing the words contained in a sentence is introduced in the system as input. In addition to word embeddings, we used SUMMA (Saggion, 2008a; Abura'ed et al., 2018) to extract features for each sentence, in order to provide information about its context in the document:

- Sentence Document Similarity: the cosine similarity of a sentence vector to the article centroid.
- Title Sentence Similarity: the cosine similarity of a sentence vector to the vector of the first sentence, that is, the title of the RP.
- TextRank Normalized: a sentence vector is computed to obtain a normalized score using the TextRank algorithm (Mihalcea & Tarau, 2004b).
- Position: a score representing the position of the sentence in the article.
- Normalized Cue-phrase: the total number of cue-words in the sentence divided by the total number of cue-words in the article based on (Teufel & Moens, 2002) formulaic expressions.
- Term Frequency: we sum up the TF*IDF values of all words in the sentence. Then, the obtained value is normalized using the set of scores from the whole document.
- Rhetorical Class Probability: the probability that the sentence belongs to each of five rhetorical categories – background, outcome, approach, challenge, and future work (five features, one per each rhetorical category) according to the scientific document analyser Dr Inventor (Ronzano & Saggion, 2015).

To calculate the similarities and TextRank Normalized features, we computed three different vectors based on the sentence representations. A vector similarity is the result of comparing two vectors of the same type using the cosine distance function. From the previous input, we also used the Google and ACL pre-trained word embeddings to generate two sentence vectors by calculating

[56] https://code.google.com/archive/p/word2vec/

the centroid (or average) of the words vectors contained in a sentence. The third vector is based on a SUMMA word vector (Saggion, 2008a), which is computed from the TF*IDF of each word. Finally, the context features are also introduced in the system (as a second input) within a sequential window including the context features of the 3 previous and 3 following sentences.

#### 7.4.2.2 Scoring Functions

The aim of the system is to learn a scoring function in order to select the most relevant sentences from a document (title + abstract). In other words, the system learns the relation between both set of features (word embeddings and context features) and a score, learning a regression task. In this work three scoring functions are defined related to the three sentence vectors (SUMMA, Google and ACL), which are basically based on the similarity between sentences in the document (title + abstract) and the gold citation sentence.

#### 7.4.2.3 Convolutional Model

The network independently decodes each input (word embeddings and context features) by convolutions to abstract higher level features. Each convolution applies a filter to produce a new feature, which is included in the resulting feature map. The convolution can be replicated with different windows with multiple filters giving multiple feature maps. Next, a max-pooling layer selects the most relevant feature from each feature map. Relevant features are concatenated together in a single feature vector. In order to prevent over-fitting, after max-pooling layer we applied dropout regularization over the single feature vector (Hinton et al., 2012). At this point, both single feature vectors generate by each input are also concatenated and the resulting vector is passed to two subsequent fully-connected layers. The fully-connected layers scale a large amount of features from the previous vector to a single output value, in order to learn the regression task. We also rescale the weights whose l2-norms exceed a hyperparameter as in (Kim, 2014) and (Nguyen & Grishman, 2015).

### 7.4.3 Sequence to Sequence Approach

We feed our training sequences (see Section 7.2) to the model and use the validation data to tune the hyper parameters and keep the learning rate in check during training. We have used 15,000 pairs for training, 1,514 pairs for development and 219 pairs for testing. The final model is fed with the set of reference papers (titles and abstracts) in the testing dataset generating a citation context for each reference paper (see Figure 7.6). Finally, we group the

generated set of citations context together to form the final related work section. We ran all our experiments on both the Title and Abstract as described at section 7.2 and the filtered version of the data.

#### 7.4.3.1 Training

For our abstractive sequence-to-sequence approach we generated several models while training the data. We ran two separate encoder-decoder architectures i.e. Transformer and BRNN as mentioned at section 7.3 with 4 recurrent layers for the transformer architecture and one layer of Bi-Directional RNN. We set the hidden size of the recurrent unit to 512 and used ADAM (Kingma & Ba, 2014) and AdaGrad (Duchi et al., 2011) optimizers respectively. We set the system to share the same weight vectors for shared vocabulary between the encoder and decoder and we add a sinusoidal position encoding to each vocabulary. This option drastically decreases the number of parameters a model has to learn.

To further represent the sentences we not only rely on the internal representation of words by the OpenNMT-py (Klein et al., 2017) tool, but we also use word-based and character-based word2vec pre-trained models. These models will provide some insight of how changing the representation of the input could affect the results, for that reason we use GoogleNews (Mikolov et al., 2013b) (word-based) and FastText (Mikolov et al., 2018) (character-based) pre-trained models to run additional experiments for both the filtered and unfiltered data.

Regarding batches and normalization, there are two types of batches; sentence based and token based. Sentence based batching sets the batch size based on the number of instances (sentences), while token based batching is also known as dynamic batching due to the fact that a batch is created based on a specific number of tokens. The motivation behind dynamic batching is to avoid any memory problems for sentences that are considered long. It is usually used with greedy algorithms such as the Transformer's multi-head attention technique. For our experiments we batch and normalize based on dynamic batching of size 4,096 tokens for the Transformer architecture. As for the BRNN we batch and normalize based on sentence batching of size 16. We set the network to compute gradients and update the parameters after each set of batches. Moreover, we initialized with Xavier uniform (Glorot & Bengio, 2010) and used 0.2 dropout (Srivastava et al., 2014) mechanism to prevent over-fitting.

We set the network to save models over time - every $K$ steps a model is saved and tested against the validation data generating a total of ten models. See

Figure 7.7 which highlights the accuracy of the network at each check point (Trans is short for Transformer). The figure shows the accuracy over the training and validation steps for the BRNN and Transformer models over the filtered (denoted as F) and unfiltered data. The BRNN models tend to have a slower and more consistent training accuracy improvements over the transformer models, the slowest learning process were recorded over the filtered data. As for validation accuracy the transformer models are more consistent and stable over the validation data. Finally the validation accuracy of both models have a higher accuracy over the filtered data.

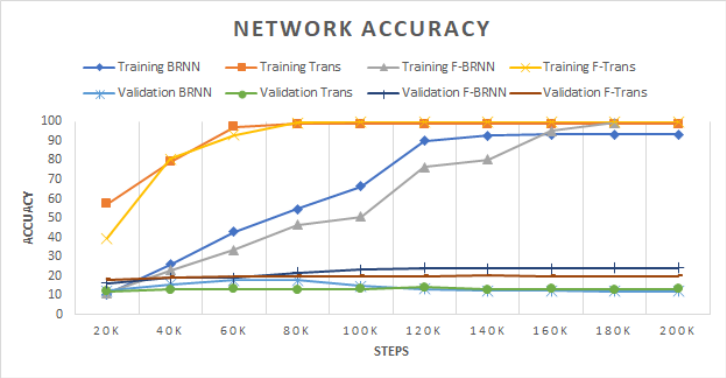


**Figure 7.7:** The neural network accuracy over training and validation data over time

Finally, we used a learning rate decay managed under the "noam" scheme (Goyal et al., 2017) (linear warm-up for a given number of steps followed by exponential decay of the learning rate).

#### 7.4.3.2 Testing

We ran the testing sequences over the models generated at each check point. The network reported the perplexity scores (Jelinek et al., 1977) at each check point (See Figure 7.8) which shows that the Transformer models have less perplexity measures than the BRNN.

The generated sentences from our system varied between readable sentences and sentences that were not acceptable, yet shared common words with the title and abstract of the cited scientific paper. An example of a good generated citation for a paper in the test set [57] is shown in Figure 7.9.

[57] Cited paper: Turney, Peter D. "Measuring semantic similarity by latent relational analysis." arXiv preprint cs/0508053 (2005).

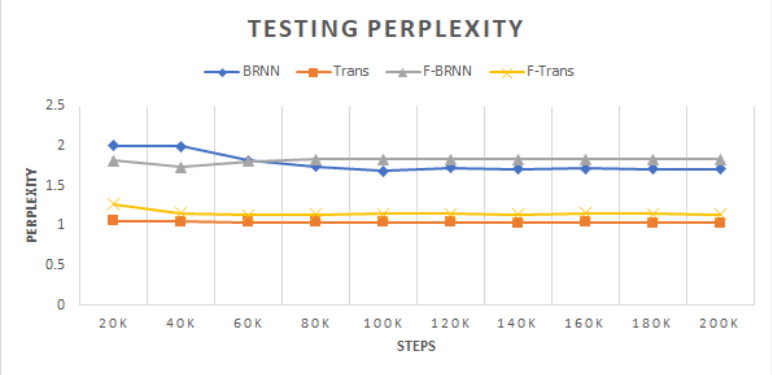


**Figure 7.8:** Perplexity of generated strings at different training points.

| | |
|---|---|
| title<br>abstract | Measuring Semantic Similarity by **Latent Relational Analysis**<br>this paper introduces **latent relational analysis** (lra), a **method** for measuring semantic similarity. this paper **describes** ... classifying semantic relations in **noun** modifier expressions. this paper has introduced a new **method** for calculating **relational** similarity, latent relational analysis. just as attributional similarity measures have proven to have many practical uses, ... . |
| gen. cit. | <CITE>describes a **method** (**latent relational analysis**) that extracts subsequence patterns for **noun** pairs from a large corpus, using query expansion to increase the recall of the search and feature selection and dimensionality reduction to reduce the complexity of the features. |

**Figure 7.9:** Example of a scientific article (title $\oplus$ abstract) and a grammatically correct generated citation sentence with considerable "matching" content.

An example of a poorly generated citation for a testing paper [58] is shown in Figure 7.10. Even though the generated citation is not very readable due to the inclusion of several "main" verbs without proper syntactic structure, some relevant keywords have been selected.

Figure 7.11 shows the entire pipeline of our experiments. We experimented on *title and abstract* of scientific papers and we also applied a filter based on Teufel's (Teufel, 2000) gazetteers producing a *title + filtered abstract*. As for the representation of the sentences, we used the internal representation by OpenNMT-py, word-based Word2Vec pre-trained model (i.e. GoogleNews)

[58]Cited paper: Ibrahim, Ali, Boris Katz, and Jimmy Lin. "Extracting structural paraphrases from aligned monolingual corpora." Proceedings of the second international workshop on Paraphrasing-Volume 16. Association for Computational Linguistics, 2003.

| | |
|---|---|
| title | Extracting Structural **Paraphrases** From Aligned **Monolingual** Corpora. |
| abstract | we present an **approach** for automatically learning **paraphrases** from aligned **monolingual** corpora. we present an **approach** for automatically learning **paraphrases** ... our algorithm works by generalizing the syntactic **paths** between corresponding anchors in aligned **sentence** pairs... we also describe a novel **information** retrieval system under development that is designed to take advantage of structural **paraphrases**. |
| gen. cit. | <CITE >proposed a **information** based **approach** to select **monolingual paraphrases** of a **paraphrases** in the **sentence paths** of a **sentence** paths to reduce the **monolingual** rules of **paraphrases** and penn variations to be identified. |

**Figure 7.10:** Example of a scientific article (title $\oplus$ abstract) and an incoherent generated citation sentence.

| | ROUGE-1 | | | ROUGE-2 | | |
|---|---|---|---|---|---|---|
| **SYSTEM** | **R** | **P** | **F** | **R** | **P** | **F** |
| Titles | 0.074 | **0.375** | 0.119 | 0.013 | **0.072** | 0.022 |
| AbsFS | 0.126 | 0.272 | 0.155 | 0.019 | 0.041 | 0.023 |
| AbsLS | 0.114 | 0.263 | 0.150 | 0.013 | 0.035 | 0.018 |
| SUMMA | 0.130 | 0.236 | 0.158 | 0.019 | 0.026 | 0.020 |
| MEAD | **0.247** | 0.215 | 0.219 | 0.067 | 0.042 | 0.048 |
| LexRank | 0.162 | 0.306 | 0.194 | 0.029 | 0.044 | 0.032 |
| TexRank | 0.211 | 0.232 | 0.207 | 0.043 | 0.038 | 0.038 |
| $SEQ^3$ | 0.045 | 0.140 | 0.066 | 0.0004 | 0.002 | 0.0007 |
| $CNN_{SUMMA}$ | 0.163 | 0.262 | 0.187 | 0.030 | 0.047 | 0.034 |
| $CNN_{Google}$ | 0.191 | 0.261 | 0.207 | 0.034 | 0.0413 | 0.034 |
| $CNN_{ACL}$ | 0.176 | 0.246 | 0.195 | 0.035 | 0.041 | 0.035 |
| $Transformer_{CB}$ | 0.216 | 0.237 | 0.215 | **0.072** | 0.063 | 0.063 |
| $BRNN_{CB}$ | 0.189 | 0.293 | 0.219 | 0.054 | 0.070 | 0.058 |
| $Transformer_{WB}$ | 0.221 | 0.248 | 0.222 | 0.070 | 0.062 | 0.062 |
| $BRNN_{WB}$ | 0.179 | 0.266 | 0.204 | 0.044 | 0.055 | 0.046 |
| Transformer | 0.192 | 0.255 | 0.219 | 0.066 | 0.071 | 0.069 |
| BRNN | 0.223 | 0.238 | **0.230** | 0.069 | **0.072** | **0.070** |

**Table 7.1:** Extractive Baseline Systems VS Abstractive Sequence-to-Sequence System. ROUGE-1 and ROUGE-2 Metrics.

| | ROUGE-1 | | | ROUGE-2 | | |
|---|---|---|---|---|---|---|
| **SYSTEM** | **R** | **P** | **F** | **R** | **P** | **F** |
| Titles | 0.074 | **0.375** | 0.119 | 0.013 | 0.072 | 0.022 |
| AbsFS | 0.118 | 0.271 | 0.150 | 0.019 | 0.043 | 0.024 |
| AbsLS | 0.115 | 0.265 | 0.146 | 0.014 | 0.036 | 0.019 |
| SUMMA | 0.142 | 0.288 | 0.180 | 0.022 | 0.040 | 0.027 |
| MEAD | 0.216 | 0.239 | 0.203 | 0.038 | 0.037 | 0.034 |
| LexRank | 0.138 | 0.292 | 0.172 | 0.024 | 0.038 | 0.028 |
| TexRank | 0.222 | 0.236 | 0.210 | 0.041 | 0.036 | 0.035 |
| $SEQ^3$ | 0.068 | 0.158 | 0.091 | 0.003 | 0.006 | 0.004 |
| $CNN_{SUMMA}$ | 0.118 | 0.250 | 0.146 | 0.018 | 0.038 | 0.023 |
| $CNN_{Google}$ | 0.182 | 0.234 | 0.187 | 0.035 | 0.038 | 0.033 |
| $CNN_{ACL}$ | 0.187 | 0.239 | 0.193 | 0.037 | 0.042 | 0.037 |
| $Transformer_{CB}$ | 0.276 | 0.267 | 0.271 | 0.120 | 0.092 | 0.104 |
| $BRNN_{CB}$ | **0.286** | 0.314 | 0.299 | **0.122** | **0.108** | **0.115** |
| $Transformer_{WB}$ | 0.274 | 0.276 | 0.275 | 0.118 | 0.092 | 0.103 |
| $BRNN_{WB}$ | 0.284 | 0.317 | **0.300** | 0.120 | 0.107 | 0.113 |
| Transformer | 0.261 | 0.251 | 0.256 | 0.116 | 0.088 | 0.100 |
| BRNN | 0.281 | 0.298 | 0.289 | 0.117 | 0.100 | 0.108 |

**Table 7.2:** Extractive Baseline Systems VS Abstractive Sequence-to-Sequence System Over the Filtered Data. ROUGE-1 and ROUGE-2 metrics

and character-based word2vec pre-trained model (i.e. FastText). The input source is fed to the pointer generator architecture (BRNN or Transformer) which generates a summary (i.e. citation context) based on the presentation of sentences.

## 7.5 Results and Discussion

In this section we compare our abstractive sequence-to-sequence approaches with the baselines. We used several ROUGE metrics (Lin, 2004) to automatically evaluate all the systems. The metrics used from ROUGE are: ROUGE-L: which uses the *Longest Common Subsequence (LCS)* evaluating the structural similarity between two summaries therefore paying attention to syntax; ROUGE-1: which checks the overlap of each word between the automated summary and the gold standard paying attention to word content; ROUGE-2: similar to ROUGE-1 but at the level of bi-gram overlap; and finally ROUGE-SU4: which considers Skip-bigram plus unigram-based co-occurrence statistics therefore considering long sequences as the basis for evaluation. ROUGE

| System | Filtered | | Non-filtered | | |
|---|---|---|---|---|---|
| | mean | sd | mean | sd | sig. |
| BRNN | 0.28 | 0.002 | 0.23 | 0.0008 | $1 * 10^{-4\dagger}$ |
| $BRNN_{CB}$ | 0.29 | 0.002 | 0.21 | 0.0009 | $9.86 * 10^{-7\dagger}$ |
| $BRNN_{WB}$ | 0.30 | 0.002 | 0.20 | 0.001 | $2 * 10^{-8\dagger}$ |
| Transf | 0.25 | 0.003 | 0.21 | 0.0005 | $0.01^{\dagger}$ |
| $Transf_{CB}$ | 0.27 | 0.001 | 0.21 | 0.001 | $2.39 * 10^{-6\dagger}$ |
| $Transf_{WB}$ | 0.27 | 0.001 | 0.22 | 0.001 | $2 * 10^{-6\dagger}$ |
| $CNN_{SUMMA}$ | 0.14 | 0.001 | 0.18 | 0.003 | $8 * 10^{-4\dagger}$ |
| $CNN_{Google}$ | 0.18 | 0.001 | 0.20 | 0.003 | 0.14 |
| $CNN_{ACL}$ | 0.19 | 0.001 | 0.19 | 0.002 | 0.86 |
| SUMMA | 0.18 | 0.001 | 0.15 | 0.001 | $0.01^{\dagger}$ |
| MEAD | 0.20 | 0.004 | 0.21 | 0.003 | 0.2 |
| LexRank | 0.17 | 0.001 | 0.19 | 0.003 | 0.09 |
| TextRank | 0.21 | 0.002 | 0.20 | 0.002 | 0.78 |

**Table 7.3:** Comparison of *filtered* vs. *non-filtered* ROUGE-1 results with two-tailed $t$-test$(20)$. Mean, standard deviation (sd), and p-values (sig.) are reported.

| System | Filtered | | Non-filtered | | |
|---|---|---|---|---|---|
| | mean | sd | mean | sd | sig. |
| BRNN | 0.10 | 0.002 | 0.07 | 0.0003 | $8 * 10^{-2\dagger}$ |
| $BRNN_{CB}$ | 0.11 | 0.002 | 0.058 | 0.0003 | $1.6 * 10^{-4\dagger}$ |
| $BRNN_{WB}$ | 0.11 | 0.002 | 0.047 | 0.002 | $2.45 * 10^{-6\dagger}$ |
| Transf | 0.10 | 0.002 | 0.069 | 0.0002 | $1.6 * 10^{-2}\dagger$ |
| $Transf_{CB}$ | 0.10 | 0.0018 | 0.063 | 0.0002 | $1 * 10^{-3\dagger}$ |
| $Transf_{WB}$ | 0.10 | 0.002 | 0.62 | 0.0001 | $7 * 10^{-4\dagger}$ |
| $CNN_{SUMMA}$ | 0.023 | 0.0001 | 0.034 | 0.0005 | $0.02^{\dagger}$ |
| $CNN_{Google}$ | 0.035 | 0.0003 | 0.034 | 0.0005 | 0.83 |
| $CNN_{ACL}$ | 0.037 | 0.0005 | 0.035 | 0.0007 | 0.77 |
| SUMMA | 0.027 | 0.00027 | 0.019 | 0.0001 | 0.09 |
| MEAD | 0.034 | 0.00098 | 0.049 | 0.00076 | $2 * 10^{-3\dagger}$ |
| LexRank | 0.028 | 0.0002 | 0.032 | 0.0002 | 0.39 |
| TextRank | 0.035 | 0.0004 | 0.038 | 0.00035 | 0.47 |

**Table 7.4:** Comparison of *filtered* vs. *non-filtered* ROUGE-2 results with two-tailed $t$-test$(20)$. Mean, standard deviation (sd), and p-values (sig.) are reported.

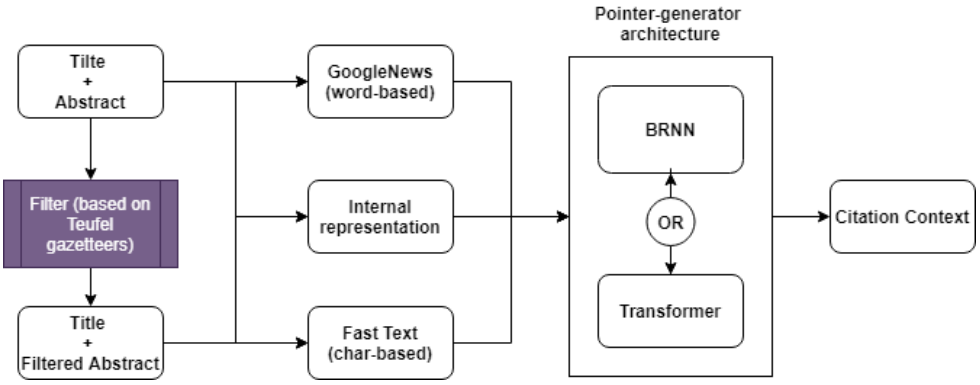


**Figure 7.11:** An outline of the performed experiments showing the different scenarios we used over our approach.

measures combine precision and recall in a harmonic F-measure which is generally used to assess the systems' performance. The results of ROUGE-1 and ROUGE-2 metrics before filtering the data can be found at Table 7.1 and over the filtered data at Table 7.2. ROUGE-L and ROUGE-SU4 results are computed for the sake of completeness and provided in the Appendix in tables C.5 and C.4 for unfiltered and filtered data respectively. As can be appreciated from the numbers in Tables 7.1, C.4, 7.2 and C.5 the non-informed extractive baselines which do not perform any analysis of the input (e.g. use of titles or sentences from abstracts) tend to have a high precision but low recall, specially the precision of the title. For all ROUGE measures, and disregarding the status of the input data (filtered/non-filtered), the sequence-to-sequence models obtain the higher scores in terms of ROUGE-F. For precision and recall variants of ROUGE in the case of non-filtered data, we can observe that MEAD is better at Recall and LexRank at precision, but not achieving the best F-score. This trend is not observed in the filtered data where the sequence-to-sequence models obtain higher results for precision, recall, and F-score (for all ROUGE measures).

We have analysed the ROUGE results by running a $t-$test[59] (using the R software and selecting 95% confidence level). We report our analysis on Tables 7.3 and 7.4 for the differences when the same approach is trained with different data types (filtered vs. non-filtered). Moreover, for each sequence-to-sequence model we analyze the effect of the embedding condition used (none, word embedding, character embedding), see Table 7.5. More specifically, Table 7.3 compares with ROUGE-1 means of the different systems under the filtered and non-filtered conditions. We can observe that differences are statistically

[59]Normality of the data was verified with a Kolmogorov-Smirnov test of normality.

significant for all sequence-to-sequence models († in the sig. column indicates if a difference was found). Table 7.4 compares ROUGE-2 results showing similar findings. Where the effect of the word embedding condition is of concerned, Table 7.5 compares with ROUGE (1 and 2) means for the BRNN and Transformer approaches. Differences are statistically significant for 9 out of 12 conditions († in the sig. column indicates if a difference was found).

| Embedding | **BRNN** | | **Transf** | | **sig.** |
|---|---|---|---|---|---|
| ROUGE-1 Filtered | | | | | |
| | mean | sd | mean | sd | |
| None | 0.28 | 0.002 | 0.25 | 0.003 | $1.09 * 10^{-6}$† |
| Word | 0.30 | 0.002 | 0.27 | 0.001 | 0.003† |
| Character | 0.29 | 0.002 | 0.27 | 0.001 | 0.0002† |
| ROUGE-1 Non-filtered | | | | | |
| | mean | sd | mean | sd | |
| None | 0.23 | 0.0008 | 0.21 | 0.0005 | 0.015† |
| Word | 0.22 | 0.001 | 0.20 | 0.001 | 0.008† |
| Character | 0.21 | 0.0009 | 0.21 | 0.0009 | 0.58 |
| ROUGE-2 Filtered | | | | | |
| | mean | sd | mean | sd | |
| None | 0.108 | 0.002 | 0.100 | 0.002 | 0.001† |
| Word | 0.11 | 0.002 | 0.10 | 0.001 | $8.5 * 10^{-5}$† |
| Character | 0.11 | 0.002 | 0.10 | 0.001 | 0.0029† |
| ROUGE-2 Non-filtered | | | | | |
| | mean | sd | mean | sd | |
| None | 0.07 | 0.0003 | 0.069 | 0.0001 | 0.50 |
| Word | 0.06 | 0.0001 | 0.046 | 0.0002 | 0.0001† |
| Character | 0.06 | 0.0002 | 0.058 | 0.0003 | 0.11 |

**Table 7.5:** Effect of pre-trained embedding in ROUGE scores using two-tailed $t$-test(20). Mean, standard deviation (sd), and p-values (sig.) are reported.

Certain limitations apply to abstractive summarization methods in which the generated text could be repetitive for certain phrases that appear often in the training data (i.e. stop words). Such repetitive could affect the comprehensibility of the text. Using a huge dataset as training could reduce the repetition also some post-processing steps could be applied. We have utilized OpenNMT-py to prevent the model from repeating trigrams in the same sentence, which could help addressing this problem. An example of an incoherent sentence in Figure 7.10 shows that the syntactic structure of the outcome text should be improved. Denoising is one of the promising techniques to tackle

this issue (Artetxe et al., 2018). It involves in reordering the input sequence and reconstructing the original word order that makes the model learn how to compose words to result in correct syntactic transformation. It is relevant to our task since there are cases when the change of positions of words in a citation sentence and a corresponding change in the syntactic structure are required to compose a meaningful summary (cf., the upper-right text in Figure 7.3). We are going to try this technique in the future taking into account that according to the recent works in denoising (Surya et al., 2019) for complex syntactic operations such as sentence splitting, rephrasing, and paraphrasing, some explicit mechanisms should also be employed.

Although our work is related to a number of scientific summarization approaches, the work most similar to ours is (Hu & Wan, 2014) who made available the dataset of related work sections used in our evaluation. Their approach however can not be compared directly with ours due to several factors but most importantly: (i) their software is not available to run and (ii) their paper does not indicate which part of the corpus was used for evaluation, leaving reproducible research of their approach difficult to achieve. We argue that the complete comparison of approaches we have carried out here provides a solid picture into the use of sequence-to-sequence approaches for this specific summarization task.

## 7.6 Conclusion

Being an essential part of every scientific article, related work sections or literature reviews pose important challenges for natural language processing in the context of the scientific text. Here, we have been concerned with the generation of "descriptive" related work given a set of scientific papers to summarize. Based on previous research, which indicate that related work sections usually include elements from titles and abstracts of the cited papers, we have reduced the complexity of the task by considering as input to our generation process only those parts of the scientific articles. Since it has also been shown that related work sections exhibit cut-and-paste summarization strategies we have investigated a sequence-to-sequence approach in order to automatically generate citation sentences for each paper to cite. Our sequence-to-sequence approach makes use of a novel dataset which we make available to the research community for further research. We additionally have presented a comparison between our abstractive approach against a set of extractive methods and evaluated them based on a gold standard dataset using content-based metrics. Our results indicate that our approach outperforms the simple as well as the informed baselines and competitive neural network approaches. Finally, We

make the data, the software and instructions on how to reproduce our work available for the community on github [60].

[60] https://github.com/AhmedAbuRaed/SPSeq2Seq

# Chapter 8

# Summary and Future Perspectives

## 8.1 Introduction

In this thesis, we have presented a number of computational approaches for the automatic generation of descriptive related work reports. These approaches utilize extractive and abstractive summarization of scientific papers to describe each scientific paper to be mentioned in the related work report.

Throughout our research we have faced a number of research questions that we presented at the beginning of the thesis and managed to answer. Moreover, based on the results presented in this thesis, we can now say that the goal that we set at the beginning of this thesis has been successfully attained. That is: Given a list of scientific papers, our main objective was to automatically summarize those scientific papers and generate a descriptive related work report. Such a report can later be used as a related work section in a scientific paper or a review of a related work in a study field which serves as a review for scholars.

In Chapter 1 we started this thesis by presenting our motivation behind analyzing and summarizing scientific papers in order to automatically generate descriptive related work reports. Then in Chapter 2 we present related works starting with automatic text summarization (ATS) in the domain of scientific texts followed by automated related work summarization and sequence-to-sequence methods. In Chapter 3 we presented a corpus in the field of scientific text mining and summarization to allow the study of automatic related work text generation. The corpus provides related work sections of scientific papers, a manually annotated layer of referenced cited papers, a level of citing papers referring to the cited papers in the related work section, and a layer of rich linguistic, rhetorical, and semantic annotations computed automatically. After that, in Chapter 4 we presented the results of our experiments on the

detection of implicit citations/references to a reseach paper, with the aim of using this method for improving the performance of a reference scope detection system . In Chapter 5 We described the systems developed to participate in the CL-SciSumm summarization challenge, where we reflect on our first research question in which we identify which sentences in a scientific paper are worth extracting in order to generate a related work report. We participated for three years in a row in this challenge and we won the competition on our last participation. Afterwards, in Chapter 6 we show how to automatically generate related work reports through extractive summarization of the list of scientific papers to be mentioned in the related work report. In this chapter we address the second and third research questions by generating organized related work reports and performing automatic and human evaluation. Finally, in Chapter 7 we describe our sequence-to-sequence approach that makes use of a novel dataset which we make available to the research community for further research. The approach generates related work reports through abtractive summarization of the list of scientific papers to be mentioned in the related work report. This is based on previous research (Jaidka et al., 2013; Saggion, 1999), which indicate that related work sections usually include elements from titles and abstracts of the cited papers.

## 8.2 Summary of Contributions

We now present a summary of the main contributions of this thesis. In Chapter 3 we created a manually annotated corpus (3 annotators) and automatically processed it. We also presented experiments to assess several text representation mechanisms (e.g. lemmas, embeddings, synsets) for the retrieval of sentences likely to be cited by scientific papers comparing system results to the gold standard annotations. The corpus is available for research and development purposes in two versions[61]; one version contains the manual annotations (agreed cited sentences) and the other contains the full machine readable corpus with the automatic analysis just described;

In Chapter 4 we designed a system with a novel set of features for implicit citation identification. We also ran a set of experiments demonstrating the improved performance of the taken approach. A novel data-set was created for the implicit citation identification task. The software and data developed are being made available to the research community[62].

In Chapter 5 we developed systems to identify which sentences in a Reference

[61] http://taln.upf.edu/sciencecorpus

[62] https://github.com/AhmedAbuRaed/CitationContextExtension

Paper have been cited by a citation context. Multiple supervised and unsupervised methods have been implemented to participate in the CL-SciSumm shared task. The software is made available for the research community [63,64].

In Chapter 6 we presented a state of the art system for automatic generation of related work reports using extractive summarization of scientific papers. performed automatic evaluation using ROUGE and a human evaluation. The software is made available for the research community [65];

Finally, in Chapter 7 we presented the design and evaluation of an abstractive related work section generation system. We created a new data set of over 15K pairs of articles and citation sentences to train sequence-to-sequence models. A comparison with state-of-the-art methods was drawn up showing the potential of the approach. The data, the software and instructions on how to reproduce our work are available for the community [66].

## 8.3 Future Work

We believe that there is always room to extend our work. For example Our multi-level corpus that we described in Chapter 3 can be enlarged to include more clusters. This is motivated by the need to have more accurate deep learning models. One of the downsides of deep learning is the need for a huge amount of data to get a better performance. We believe that the amount of already annotated and processed data sets and the availability of resources to investigate scientific papers are not enough for the ultimate utilization of deep learning methods.

In Chapter 4 there are several avenues of possible research to improve over our work in detecting implicit citations in a scientific paper. More data for training and evaluation might be necessary to create better classifiers. Also, it would be very interesting to try more advanced techniques, for example using deep learning methods over the classical machine learning methods we tried. That does not mean that we don't also recommend using unsupervised methods in case there is no access to more data for training supervised models. While working on Chapter 5 we realized that unsupervised methods can perform well with scientific text. Therefore, we recommend trying to find a similarity between each sentence in the scientific paper and an explicit citation sentence, given that the methods to identify explicit citations have a very high accuracy.

[63] https://github.com/AhmedAbuRaed/CLSciSumm2018
[64] https://github.com/AhmedAbuRaed/CL-SciSumm2017
[65] https://github.com/AhmedAbuRaed/RWRG
[66] https://github.com/AhmedAbuRaed/SPSeq2Seq

So, one could first identify explicit citations, then, try to find the similarity between a pair of sentences: explicit citation sentence with the other sentences could give a good insight on identifying implicit citations.

As for our participation at the CL-SciSumm challenge, we have already tried many methods and we have improved over time. Still there is more that is worth trying. For example, we could carry out an exhaustive performance and feature analysis on test/development data. We could also extend our corpus described in Chapter 3 as we said before.

There are things that we did not try or can be improved in Chapter 6, such as: sentence ordering, using passive sentences, redundancy removal and finally, generating citation sentences for multiple scientific papers (citation list). We have tried topic modelling (Blei et al., 2003) to order the sentences but we believe that there is still room for improvement. Probabilistic ordering (Lapata, 2003) is another way to investigate sentence ordering. Also, a limitation that we reported is that our method extract sentences from the reference papers directly. Therefore it can be noticed that the generated related work reports has sentences that use first pronouns, since the authors of the reference paper describe their work. In order to avoid this we can investigate a post-processing step to replace such sentences with passive sentences.

For the seq2seq methods which we described in Chapter 7 we noticed two things: when we increased the number of instances to consider in the training and validation data we got better results. Also, when we applied some filtration on the data we got better results as reported in the Chapter. Therefore using the same resources i.e. Microsoft Academic Graph and Semantic Scholar to extend the size of the data, we found that this improved the results. Moreover, since filtering the data by including only sentences that refer directly to the authors of the scientific papers will improve the model, then we suggest finding ways to have a more active filtering of data. The shorter length of a sequence the better learning process you get.

There are more ways to build upon our methods, Devlin et al. (2018) presented a novel state of the art language representation model called BERT. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in many layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. So it is possible to try to represent the scientific paper using BERT and score sentences based on that representation. Of course citations towards those scientific pa-

pers can also be taken into consideration.

Finally, we shared all of our resources with the community which makes it easier to either reproduce our work or build on it.

# Appendix A

# Publications by the Author

# Appendix B

# Resources

## B.1 Data Released

## B.2 Software Released

# Appendix C

# Additional Figures and Tables

| | ROUGE-1 | | | ROUGE-2 | | |
|---|---|---|---|---|---|---|
| **SYSTEM** | **R** | **P** | **F** | **R** | **P** | **F** |
| Titles | 0.074 | **0.375** | 0.119 | 0.013 | 0.072 | 0.022 |
| AbsFS | 0.126 | 0.272 | 0.155 | 0.019 | 0.041 | 0.023 |
| AbsLS | 0.114 | 0.263 | 0.150 | 0.013 | 0.035 | 0.018 |
| SUMMA | 0.293 | 0.103 | 0.154 | 0.102 | 0.027 | 0.047 |
| MEAD | 0.361 | 0.137 | 0.205 | 0.118 | 0.025 | 0.049 |
| LexRank | 0.312 | 0.228 | 0.259 | 0.107 | 0.060 | 0.076 |
| TexRank | 0.367 | 0.116 | 0.186 | 0.117 | 0.017 | 0.040 |
| Babelnet | 0.387 | 0.231 | 0.286 | 0.152 | 0.084 | 0.107 |
| MJ | 0.336 | 0.266 | 0.292 | 0.151 | 0.114 | 0.127 |
| $CNN_{SUMMA-abstract}$ | 0.317 | 0.264 | 0.285 | 0.137 | 0.105 | 0.118 |
| $CNN_{Google-abstract}$ | 0.349 | 0.240 | 0.281 | 0.137 | 0.089 | 0.107 |
| $CNN_{ACL-abstract}$ | 0.352 | 0.245 | 0.285 | 0.138 | 0.090 | 0.107 |
| $CNN_{ROUGE-2-abstract}$ | 0.303 | 0.285 | 0.290 | 0.137 | **0.122** | 0.127 |
| $CNN_{AvgGAR-abstract}$ | 0.340 | 0.265 | 0.294 | 0.141 | 0.105 | 0.118 |
| $CNN_{AvgSGAR-abstract}$ | 0.344 | 0.272 | 0.300 | 0.139 | 0.106 | 0.119 |
| $CNN_{SUMMA-community}$ | 0.318 | 0.257 | 0.282 | 0.132 | 0.099 | 0.112 |
| $CNN_{Google-community}$ | 0.358 | 0.222 | 0.271 | 0.144 | 0.084 | 0.104 |
| $CNN_{ACL-community}$ | 0.366 | 0.221 | 0.273 | 0.142 | 0.080 | 0.101 |
| $CNN_{ROUGE-2-community}$ | 0.335 | 0.270 | 0.296 | 0.143 | 0.107 | 0.121 |
| $CNN_{AvgGAR-community}$ | 0.350 | 0.240 | 0.281 | 0.145 | 0.092 | 0.111 |
| $CNN_{AvgSGAR-community}$ | 0.354 | 0.248 | 0.287 | 0.145 | 0.094 | 0.112 |
| $CNN_{SUMMA-human}$ | 0.280 | 0.271 | 0.272 | 0.127 | 0.115 | 0.119 |
| $CNN_{Google-human}$ | 0.304 | 0.251 | 0.270 | 0.128 | 0.097 | 0.109 |
| $CNN_{ACL-human}$ | 0.318 | 0.243 | 0.272 | 0.133 | 0.094 | 0.109 |
| $CNN_{ROUGE-2-human}$ | 0.310 | 0.281 | 0.291 | 0.143 | 0.121 | 0.129 |
| $CNN_{AvgGAR-human}$ | 0.326 | 0.269 | 0.291 | 0.134 | 0.104 | 0.116 |
| $CNN_{AvgSGAR-human}$ | 0.328 | 0.261 | 0.286 | 0.135 | 0.100 | 0.113 |
| Babelnet (MBL) | **0.409** | 0.242 | 0.299 | **0.158** | 0.087 | 0.110 |
| MJ (MBL) | 0.350 | 0.270 | 0.299 | 0.154 | 0.112 | 0.127 |
| $CNN_{SUMMA-abstract(MBL)}$ | 0.335 | 0.267 | 0.294 | 0.143 | 0.106 | 0.120 |
| $CNN_{Google-abstract(MBL)}$ | 0.362 | 0.244 | 0.287 | 0.137 | 0.088 | 0.105 |
| $CNN_{ACL-abstract(MBL)}$ | 0.370 | 0.250 | 0.293 | 0.143 | 0.092 | 0.110 |
| $CNN_{ROUGE-2-abstract(MBL)}$ | 0.322 | 0.291 | 0.302 | 0.142 | 0.120 | 0.128 |
| $CNN_{AvgGAR-abstract(MBL)}$ | 0.360 | 0.273 | 0.307 | 0.148 | 0.108 | 0.124 |
| $CNN_{AvgSGAR-abstract(MBL)}$ | 0.359 | 0.278 | **0.310** | 0.149 | 0.110 | 0.125 |
| $CNN_{SUMMA-community(MBL)}$ | 0.330 | 0.266 | 0.290 | 0.134 | 0.102 | 0.114 |
| $CNN_{Google-community(MBL)}$ | 0.377 | 0.226 | 0.278 | 0.148 | 0.084 | 0.105 |
| $CNN_{ACL-community(MBL)}$ | 0.382 | 0.220 | 0.276 | 0.145 | 0.078 | 0.100 |
| $CNN_{ROUGE-2-community(MBL)}$ | 0.353 | 0.275 | 0.305 | 0.152 | 0.111 | 0.126 |
| $CNN_{AvgGAR-community(MBL)}$ | 0.372 | 0.245 | 0.290 | 0.154 | 0.094 | 0.114 |
| $CNN_{AvgSGAR-community(MBL)}$ | 0.369 | 0.247 | 0.290 | 0.151 | 0.095 | 0.114 |
| $CNN_{SUMMA-human(MBL)}$ | 0.287 | 0.273 | 0.276 | 0.130 | 0.115 | 0.120 |
| $CNN_{Google-human(MBL)}$ | 0.318 | 0.243 | 0.271 | 0.133 | 0.095 | 0.109 |
| $CNN_{ACL-human(MBL)}$ | 0.333 | 0.251 | 0.283 | 0.137 | 0.095 | 0.111 |
| $CNN_{ROUGE-2-human(MBL)}$ | 0.319 | 0.285 | 0.298 | 0.145 | 0.120 | **0.130** |
| $CNN_{AvgGAR-human(MBL)}$ | 0.341 | 0.279 | 0.302 | 0.139 | 0.107 | 0.119 |
| $CNN_{AvgSGAR-human(MBL)}$ | 0.346 | 0.268 | 0.297 | 0.142 | 0.103 | 0.117 |

**Table C.1:** Automatic evaluation results of our systems against the baselines for ROUGE-1 and ROUGE-2 metrics.

| | ROUGE-L | | | ROUGE-SU4 | | |
|---|---|---|---|---|---|---|
| **SYSTEM** | **R** | **P** | **F** | **R** | **P** | **F** |
| Titles | 0.087 | **0.363** | 0.134 | 0.029 | **0.147** | 0.046 |
| AbsFS | 0.149 | 0.260 | 0.174 | 0.051 | 0.082 | 0.056 |
| AbsLS | 0.127 | 0.221 | 0.151 | 0.045 | 0.079 | 0.054 |
| SUMMA | 0.250 | 0.091 | 0.135 | 0.156 | 0.046 | 0.075 |
| MEAD | 0.269 | 0.117 | 0.168 | 0.198 | 0.034 | 0.071 |
| LexRank | 0.243 | 0.197 | 0.215 | 0.169 | 0.074 | 0.105 |
| TexRank | 0.282 | 0.117 | 0.172 | 0.196 | 0.025 | 0.060 |
| Babelnet | 0.327 | 0.199 | 0.245 | 0.218 | 0.108 | 0.142 |
| MJ | 0.266 | 0.222 | 0.238 | 0.207 | 0.134 | 0.159 |
| $CNN_{SUMMA-abstract}$ | 0.260 | 0.210 | 0.230 | 0.185 | 0.128 | 0.149 |
| $CNN_{Google-abstract}$ | 0.295 | 0.195 | 0.233 | 0.193 | 0.115 | 0.141 |
| $CNN_{ACL-abstract}$ | 0.305 | 0.208 | 0.245 | 0.196 | 0.117 | 0.144 |
| $CNN_{ROUGE-2-abstract}$ | 0.257 | 0.251 | 0.251 | 0.174 | 0.137 | 0.150 |
| $CNN_{AvgGAR-abstract}$ | 0.290 | 0.222 | 0.249 | 0.188 | 0.125 | 0.148 |
| $CNN_{AvgSGAR-abstract}$ | 0.291 | 0.227 | 0.253 | 0.190 | 0.128 | 0.150 |
| $CNN_{SUMMA-community}$ | 0.259 | 0.206 | 0.228 | 0.182 | 0.121 | 0.143 |
| $CNN_{Google-community}$ | 0.306 | 0.183 | 0.227 | 0.199 | 0.105 | 0.135 |
| $CNN_{ACL-community}$ | 0.307 | 0.185 | 0.229 | 0.204 | 0.104 | 0.136 |
| $CNN_{ROUGE-2-community}$ | 0.279 | 0.229 | 0.249 | 0.192 | 0.127 | 0.151 |
| $CNN_{AvgGAR-community}$ | 0.292 | 0.199 | 0.235 | 0.199 | 0.114 | 0.142 |
| $CNN_{AvgSGAR-community}$ | 0.296 | 0.207 | 0.241 | 0.201 | 0.117 | 0.145 |
| $CNN_{SUMMA-human}$ | 0.231 | 0.217 | 0.222 | 0.167 | 0.134 | 0.147 |
| $CNN_{Google-human}$ | 0.252 | 0.192 | 0.215 | 0.174 | 0.119 | 0.138 |
| $CNN_{ACL-human}$ | 0.265 | 0.195 | 0.223 | 0.182 | 0.116 | 0.139 |
| $CNN_{ROUGE-2-human}$ | 0.263 | 0.240 | 0.248 | 0.182 | 0.137 | 0.153 |
| $CNN_{AvgGAR-human}$ | 0.279 | 0.214 | 0.240 | 0.184 | 0.128 | 0.149 |
| $CNN_{AvgSGAR-human}$ | 0.279 | 0.211 | 0.237 | 0.186 | 0.124 | 0.146 |
| Babelnet (MBL) | **0.345** | 0.203 | 0.252 | **0.228** | 0.112 | 0.147 |
| MJ (MBL) | 0.273 | 0.219 | 0.240 | 0.215 | 0.135 | **0.162** |
| $CNN_{SUMMA-abstract(MBL)}$ | 0.277 | 0.211 | 0.237 | 0.194 | 0.128 | 0.152 |
| $CNN_{Google-abstract(MBL)}$ | 0.311 | 0.197 | 0.239 | 0.197 | 0.114 | 0.142 |
| $CNN_{ACL-abstract(MBL)}$ | 0.319 | 0.209 | 0.249 | 0.206 | 0.119 | 0.147 |
| $CNN_{ROUGE-2-abstract(MBL)}$ | 0.273 | 0.255 | 0.262 | 0.184 | 0.137 | 0.154 |
| $CNN_{AvgGAR-abstract(MBL)}$ | 0.309 | 0.230 | 0.262 | 0.198 | 0.129 | 0.154 |
| $CNN_{AvgSGAR-abstract(MBL)}$ | 0.302 | 0.229 | 0.258 | 0.199 | 0.132 | 0.156 |
| $CNN_{SUMMA-community(MBL)}$ | 0.265 | 0.210 | 0.230 | 0.187 | 0.125 | 0.147 |
| $CNN_{Google-community(MBL)}$ | 0.325 | 0.186 | 0.235 | 0.211 | 0.108 | 0.140 |
| $CNN_{ACL-community(MBL)}$ | 0.321 | 0.180 | 0.229 | 0.210 | 0.102 | 0.136 |
| $CNN_{ROUGE-2-community(MBL)}$ | 0.288 | 0.228 | 0.251 | 0.204 | 0.130 | 0.156 |
| $CNN_{AvgGAR-community(MBL)}$ | 0.316 | 0.203 | 0.244 | 0.210 | 0.115 | 0.146 |
| $CNN_{AvgSGAR-community(MBL)}$ | 0.311 | 0.207 | 0.245 | 0.209 | 0.116 | 0.146 |
| $CNN_{SUMMA-human(MBL)}$ | 0.233 | 0.214 | 0.221 | 0.170 | 0.133 | 0.147 |
| $CNN_{Google-human(MBL)}$ | 0.263 | 0.190 | 0.217 | 0.182 | 0.114 | 0.137 |
| $CNN_{ACL-human(MBL)}$ | 0.278 | 0.202 | 0.232 | 0.189 | 0.117 | 0.143 |
| $CNN_{ROUGE-2-human(MBL)}$ | 0.273 | 0.246 | **0.256** | 0.185 | 0.136 | 0.154 |
| $CNN_{AvgGAR-human(MBL)}$ | 0.293 | 0.222 | 0.250 | 0.191 | 0.132 | 0.153 |
| $CNN_{AvgSGAR-human(MBL)}$ | 0.292 | 0.218 | 0.246 | 0.195 | 0.127 | 0.150 |

**Table C.2:** Automatic evaluation results of our systems against the baselines for ROUGE-1 and ROUGE-2 metrics.

| Cluster | System | Responsiveness | Linguistic quality | Text organization |
|---|---|---|---|---|
| C08-1013 | Gold | 4.3 | 4 | 4 |
| | LexRank | 2.3 | 1.6 | 2.3 |
| | NoTM | 2.6 | 2.6 | 2 |
| | TM | 3.6 | 3 | 2 |
| C08-1031 | Gold | 4.3 | 4.6 | 4.6 |
| | LexRank | 2.6 | 1.6 | 1.3 |
| | NoTM | 3 | 3.6 | 2 |
| | TM | 3.3 | 3.6 | 2 |
| C08-1064 | Gold | 4.6 | 3.6 | 4 |
| | LexRank | 2.3 | 1.6 | 1.6 |
| | NoTM | 2.6 | 3 | 2 |
| | TM | 3.3 | 4 | 1.6 |
| C08-1066 | Gold | 4.3 | 4.3 | 4.6 |
| | LexRank | 2.3 | 2.3 | 2 |
| | NoTM | 3.3 | 3.6 | 1.6 |
| | TM | 3 | 4.3 | 2.3 |
| N09-1027 | Gold | 5 | 4.3 | 4.6 |
| | LexRank | 1.6 | 3 | 1.3 |
| | NoTM | 2.6 | 3 | 1.6 |
| | TM | 3 | 3.3 | 1.6 |
| N09-1034 | Gold | 5 | 5 | 5 |
| | LexRank | 3 | 2.6 | 2 |
| | NoTM | 2.6 | 3 | 2.3 |
| | TM | 2.6 | 3 | 2.3 |
| P07-1034 | Gold | 4 | 4.6 | 4.6 |
| | LexRank | 2.6 | 3.3 | 2.6 |
| | NoTM | 3 | 3.6 | 2.3 |
| | TM | 3 | 3.6 | 3 |
| P08-1032 | Gold | 4.3 | 4.6 | 4.3 |
| | LexRank | 2.3 | 3 | 2.6 |
| | NoTM | 2.6 | 3.6 | 3 |
| | TM | 3 | 4 | 3.6 |
| P08-1052 | Gold | 5 | 4.6 | 4.3 |
| | LexRank | 2.3 | 3 | 2 |
| | NoTM | 2.6 | 3 | 3 |
| | TM | 3 | 3 | 3 |
| p79-raghavan | Gold | 4 | 4 | 3.3 |
| | LexRank | 3 | 2.6 | 2.6 |
| | NoTM | 4 | 4 | 3.3 |
| | TM | 3.6 | 4 | 3.6 |

The results of the Human Evaluation over our system with and without applying topic modeling against the LexRank baseline - Average across clusters.

| | ROUGE-L | | | ROUGE-SU4 | | |
|---|---|---|---|---|---|---|
| **SYSTEM** | **R** | **P** | **F** | **R** | **P** | **F** |
| Titles | 0.087 | 0.363 | 0.134 | 0.029 | **0.147** | 0.046 |
| AbsFS | 0.149 | 0.260 | 0.174 | 0.051 | 0.082 | 0.056 |
| AbsLS | 0.127 | 0.221 | 0.151 | 0.045 | 0.079 | 0.054 |
| SUMMA | 0.129 | 0.186 | 0.146 | 0.052 | 0.059 | 0.052 |
| MEAD | **0.209** | 0.178 | 0.179 | **0.130** | 0.067 | 0.082 |
| LexRank | 0.161 | 0.259 | 0.183 | 0.067 | 0.092 | 0.070 |
| TexRank | 0.186 | 0.194 | 0.178 | 0.092 | 0.067 | 0.073 |
| $SEQ^3$ | 0.043 | 0.281 | 0.074 | 0.016 | 0.038 | 0.021 |
| $CNN_{SUMMA}$ | 0.170 | 0.227 | 0.181 | 0.070 | 0.081 | 0.070 |
| $CNN_{Google}$ | 0.201 | 0.225 | 0.199 | 0.081 | 0.077 | 0.073 |
| $CNN_{ACL}$ | 0.191 | 0.206 | 0.189 | 0.077 | 0.075 | 0.071 |
| $Transformer_{CB}$ | 0.190 | 0.189 | 0.179 | 0.103 | 0.077 | 0.084 |
| $BRNN_{CB}$ | 0.070 | **0.365** | 0.103 | 0.090 | 0.096 | 0.088 |
| $Transformer_{WB}$ | 0.198 | 0.198 | 0.189 | 0.105 | 0.078 | 0.085 |
| $BRNN_{WB}$ | 0.077 | 0.358 | 0.118 | 0.080 | 0.083 | 0.077 |
| Transformer | 0.166 | 0.228 | 0.192 | 0.098 | 0.089 | 0.093 |
| BRNN | 0.192 | 0.213 | **0.202** | 0.110 | 0.091 | **0.099** |

**Table C.4:** Extractive Baseline Systems VS Abstractive Sequence-to-Sequence System.ROUGE-L and ROUGE-SU4 Metrics

| | ROUGE-L | | | ROUGE-SU4 | | |
|---|---|---|---|---|---|---|
| **SYSTEM** | **R** | **P** | **F** | **R** | **P** | **F** |
| Titles | 0.087 | 0.363 | 0.134 | 0.029 | 0.147 | 0.046 |
| AbsFS | 0.143 | 0.261 | 0.171 | 0.048 | 0.082 | 0.055 |
| AbsLS | 0.131 | 0.236 | 0.157 | 0.045 | 0.078 | 0.052 |
| SUMMA | 0.154 | 0.243 | 0.178 | 0.058 | 0.085 | 0.066 |
| MEAD | 0.179 | 0.190 | 0.166 | 0.093 | 0.070 | 0.072 |
| LexRank | 0.157 | 0.264 | 0.179 | 0.056 | 0.092 | 0.062 |
| TexRank | 0.204 | 0.196 | 0.187 | 0.093 | 0.068 | 0.073 |
| $SEQ^3$ | 0.078 | 0.205 | 0.109 | 0.024 | 0.042 | 0.029 |
| $CNN_{SUMMA}$ | 0.141 | 0.230 | 0.162 | 0.047 | 0.075 | 0.052 |
| $CNN_{Google}$ | 0.189 | 0.197 | 0.179 | 0.077 | 0.069 | 0.066 |
| $CNN_{ACL}$ | 0.191 | 0.203 | 0.185 | 0.082 | 0.072 | 0.071 |
| $Transformer_{CB}$ | 0.231 | 0.231 | 0.231 | 0.155 | 0.097 | 0.119 |
| $BRNN_{CB}$ | 0.117 | 0.437 | 0.184 | 0.163 | 0.117 | 0.136 |
| $Transformer_{WB}$ | **0.238** | 0.235 | **0.237** | 0.153 | 0.099 | 0.120 |
| $BRNN_{WB}$ | 0.137 | 0.415 | 0.206 | **0.165** | **0.119** | **0.138** |
| Transformer | 0.225 | 0.215 | 0.220 | 0.145 | 0.092 | 0.112 |
| BRNN | 0.124 | **0.444** | 0.193 | **0.165** | 0.113 | 0.134 |

**Table C.5:** Extractive Baseline Systems VS Abstractive Sequence-to-Sequence System Over the Filtered Data. ROUGE-L and ROUGE-SU4 metrics

# Appendix D

# Glossary

## D.1 Acronyms

| | |
|---|---|
| *n*-gram | *n*-gram model |
| AAN | ACL Anthology Network |
| ACL | Association for Computational Linguistics |
| ANNIE | a Nearly-New Information Extraction System |
| API | Application Programming Interface |
| ATS | Automatic Text Summarization |
| BE | Basic Elements |
| BioSumm | Biomedical Summarization |
| BoW | Bag of Words |
| BRNN | Bidirectional Recurrent Neural Network |
| CL | Computational Linguistics |
| CNN | Convolutional Neural Network |
| COLING | International Conference on Computational Linguistics |
| CP | Citing Paper |
| CRFs | Conditional random fields |
| CS | Computer Science |
| DP | Dependency Parsing |
| DRI | Dr. Inventor |
| EMNLP | Empirical Methods for Natural Language Processing |
| GATE | General Architecture for Text Engineering |
| GCSum | General Content Summarization |
| HTML | Hypertext Markup Language |
| IDF | Inverse Document Frequency |
| IS | Information Science |
| ISA | Information Science Abstracts |

| | |
|---|---|
| LDA | Latent Dirichlet Allocation |
| LISA | Library & Information Science Abstracts |
| LSTM | Long short-term memory |
| MAG | Microsoft Academic Graph |
| MRF | Markov Random Field |
| MSE | Mean Squared Error |
| NAACL | North American Chapter of the Association for Computational Linguistics |
| NLP | Natural Language Processing |
| NMT | Neural Machine Translation |
| OAG | Open Academic Graph |
| OCR | Optical character recognition |
| PBMT | Phrased Based Machine Translation |
| PDF | Portable Document Format |
| PLSA | Probabilistic Latent Semantic Analysis |
| POS | Part of Speech |
| QA | Question Answering |
| RNN | Recurrent Neural Network |
| ROUGE | Recall-Oriented Understudy for Gisting Evaluation |
| RP | Reference Paper |
| SciSumm | Scientific Document Summarization |
| SCSum | Specific Content Summarization |
| SCUs | Summary Content Units |
| SIGIR | Special Interest Group on Information Retrieval |
| SMO | Sequential Minimal Optimization |
| SUC | Summary Content Unit |
| SVM | Support Vector Machines |
| TE | Textual Entailment |
| TF | Term Frequency |
| TP | Target Paper |
| WEKA | Waikato Environment for Knowledge Analysis |
| WMD | Word Movers Distance |
| WS4J | WordNet Similarity for Java |

# Bibliography

Abu-Jbara, A., Ezra, J., & Radev, D. (2013). Purpose and polarity of citation: Towards nlp-based bibliometrics. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 596–606. [Cited on pages 5 and 52.]

Abu-Jbara, A. & Radev, D. (2011). Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 500–509. Association for Computational Linguistics. [Cited on page 19.]

AbuRa'ed, A., Chiruzzo, L., Saggion, H., Accuosto, P., & Bravo, À. (2017). Lastus/taln @ clscisumm-17: Cross-document sentence matching and scientific text summarization systems. In *Proceedings of the Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017) organized as a part of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2017) and co-located with the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017), Tokyo, Japan, August 11, 2017.*, pp. 55–66. [Cited on page 91.]

AbuRa'ed, A., Saggion, H., & Chiruzzo, L. (2020a). A multi-level annotated corpus of scientific papers for scientific document summarization and cross-document relation discovery. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, & S. Piperidis (Eds.) *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pp. 6672–6679. European Language Resources Association. [Cited on page 12.]

AbuRa'ed, A., Saggion, H., & Chiruzzo, L. (2020b). A multi-level annotated corpus of scientific papers for scientific document summarization and cross-document relation discovery. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*. [Cited on page 101.]

Abura'ed, A., Bravo, A., Chiruzzo, L., & Saggion, H. (2018). Lastus/taln+ inco@ cl-scisumm 2018-using regression and convolutions for

cross-document semantic linking and summarization of scholarly literature. In *Proceedings of the 3nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2018). Ann Arbor, Michigan (July 2018)*. [Cited on pages 12, 41, 122, and 123.]

AbuRa'ed, A., Chiruzzo, L., & Saggion, H. (2017). What sentence are you referring to and why? identifying cited sentences in scientific literature. In *RANLP 2017. International Conference Recent Advances in Natural Language Processing; 2017 Sep 2-8; Varna, Bulgaria.[Stroudsburg (PA)]: ACL; 2017. p. 9-17.* ACL (Association for Computational Linguistics). [Cited on pages 12 and 46.]

AbuRa'ed, A., Chiruzzo, L., & Saggion, H. (2018). Experiments in detection of implicit citations. In *WOSP 2018. 7th International Workshop on Mining Scientific Publications; 2018 May 7; Miyazaki, Japan.[Paris (Francce)]: European Language Resources Association; 2018. 7 p.* ELRA (European Language Resources Association). [Cited on pages 12 and 40.]

Abura'ed, A., Chiruzzo, L., Saggion, H., Accuosto, P., & Bravo Serrano, À. (2017). Lastus/taln@ clscisumm-17: Cross-document sentence matching and scientific text summarization systems. [Cited on pages 12, 78, and 122.]

Agarwal, N., Gvr, K., Reddy, R. S., & Rosé, C. P. (2011). Towards multi-document summarization of scientific articles: making interesting comparisons with scisumm. In *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, pp. 8–15. Association for Computational Linguistics. [Cited on pages 15 and 19.]

Aggarwal, P. & Sharma, R. (2016). Lexical and syntactic cues to identify reference scope of citance. In *Proceedings of the joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL)*, pp. 103–112. [Cited on page 77.]

Altmami, N. I. & Menai, M. E. B. (2018). Semantic graph based automatic summarization of multiple related work sections of scientific articles. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pp. 255–259. Springer. [Cited on page 31.]

Artetxe, M., Labaka, G., Agirre, E., & Cho, K. (2018). Unsupervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018*. [Cited on page 133.]

Athar, A. (2011). Sentiment analysis of citations using sentence structure-based features. In *Proceedings of the ACL 2011 student session*, pp. 81–87. Association for Computational Linguistics. [Cited on page 52.]

Athar, A. (2014). Sentiment analysis of scientific citations. Tech. rep., University of Cambridge, Computer Laboratory. [Cited on page 56.]

Athar, A. & Teufel, S. (2012a). Context-enhanced citation sentiment detection. In *Proceedings of the 2012 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies*, pp. 597–601. Association for Computational Linguistics. [Cited on page 49.]

Athar, A. & Teufel, S. (2012b). Detection of implicit citations for sentiment detection. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, pp. 18–26. Jeju Island, Korea: Association for Computational Linguistics. [Cited on pages 34, 51, 52, 53, 54, 55, 58, 59, and 60.]

Athar, A. & Teufel, S. (2012c). Detection of implicit citations for sentiment detection. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, pp. 18–26. Association for Computational Linguistics. [Cited on pages 51 and 52.]

Banerjee, S. & Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72. [Cited on page 52.]

Banerjee, S. & Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. In *Computational linguistics and intelligent text processing*, pp. 136–145. Springer. [Cited on page 71.]

Baziotis, C., Androutsopoulos, I., Konstas, I., & Potamianos, A. (2019). Seqˆ 3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. *arXiv preprint arXiv:1904.03651*. [Cited on pages 31, 121, and 122.]

Beil, F., Ester, M., & Xu, X. (2002). Frequent term-based text clustering. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 436–442. ACM. [Cited on page 19.]

Bird, S., Dale, R., Dorr, B. J., Gibson, B. R., Joseph, M. T., Kan, M.-Y., Lee, D., Powley, B., Radev, D. R., & Tan, Y. F. (2008). The acl anthology

reference corpus: A reference dataset for bibliographic research in computational linguistics. In *LREC*. [Cited on pages 38, 45, 53, 56, 79, 89, 115, and 123.]

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022. [Cited on pages 100, 106, and 138.]

Bornmann, L. & Mutz, R. (2015). Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *JASIST*, *66*(11), 2215–2222. [Cited on page 1.]

Bražinskas, A., Lapata, M., & Titov, I. (2019). Unsupervised multi-document opinion summarization as copycat-review generation. *arXiv preprint arXiv:1911.02247*. [Cited on page 30.]

Brügmann, S., Bouayad-Agha, N., Burga, A., Carrascosa, S., Ciaramella, A., Ciaramella, M., Codina-Filba, J., Escorsa, E., Judea, A., Mille, S. et al. (2015). Towards content-oriented patent document processing: intelligent patent analysis and summarization. *World Patent Information*, *40*, 30–42. [Cited on pages 69 and 73.]

Camacho-Collados, J., Pilehvar, M. T., & Navigli, R. (2016). Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, *240*, 36–64. [Cited on page 89.]

Cao, Z., Li, W., & Wu, D. (2016). Polyu at cl-scisumm 2016. In *Proceedings of the joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL)*, pp. 132–138. [Cited on pages 25 and 77.]

Chandrasekaran, M. K., Yasunaga, M., Radev, D., Freitag, D., & Kan, M.-Y. (2019). Overview and results: Cl-scisumm shared task 2019. *arXiv preprint arXiv:1907.09854*. [Cited on pages 63 and 103.]

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*. [Cited on page 114.]

Chopra, S., Auli, M., & Rush, A. M. (2016). Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 93–98. [Cited on page 113.]

Chu, E. & Liu, P. J. (2018). Meansum: a neural model for unsupervised multi-document abstractive summarization. *arXiv preprint arXiv:1810.05739*. [Cited on page 31.]

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., & Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*. [Cited on page 30.]

Cohan, A. & Goharian, N. (2017). Scientific article summarization using citation-context and article's discourse structure. *arXiv preprint arXiv:1704.06619*. [Cited on page 22.]

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, *20*(1), 37–46. [Cited on page 41.]

Conroy, J. & Davis, S. T. (2015). Vector space models for scientific document summarization. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 186–191. [Cited on pages 24 and 77.]

Constantin, A., Pettifer, S., & Voronkov, A. (2013). Pdfx: fully-automated pdf-to-xml conversion of scientific literature. In *Proceedings of the 2013 ACM symposium on Document engineering*, pp. 177–180. ACM. [Cited on page 39.]

Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*. [Cited on page 42.]

da Cunha, I. & Wanner, L. (2005). Towards the automatic summarization of medical articles in spanish: Integration of textual, lexical, discursive and syntactic criteria. *Crossing Barriers in Text Summarization Research. RANLP, Borovets*. [Cited on page 17.]

Da Cunha, I., Wanner, L., & Cabré, T. (2007). Summarization of specialized discourse: The case of medical articles in spanish. *Terminology*, *13*(2), 249–286. [Cited on page 17.]

de Solla Price, D. J. & Page, T. (1961). Science since babylon. *American Journal of Physics*, *29*(12), 863–864. [Cited on page 1.]

Debnath, D., Achom, A., & Pakray, P. (2018). Nlp-nitmz@ clscisumm-18. In *BIRNDL@ SIGIR*, pp. 164–171. [Cited on page 27.]

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. [Cited on page 138.]

Dipankar Das, S. & Pramanick, A. (2017). Employing word vectors for identifying, classifying and summarizing scientific documents. In *Proc. of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2017). Tokyo, Japan (August 2017)*. [Cited on page 26.]

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*(Jul), 2121–2159. [Cited on page 125.]

Edmundson, H. P. (1969). New methods in automatic extracting. *J. ACM*, *16*(2), 264–285. [Cited on page 15.]

Endres-Niggemeyer, B., Elisabeth Maier, E., & Alexander Sigel, A. (1995). How to implement a naturalistic model of abstracting: Four core working steps of an expert abstractor. *Information Processing and Management*, *31*(5), 631 – 674. [Cited on page 7.]

Erera, S., Shmueli-Scheuer, M., Feigenblat, G., Nakash, O. P., Boni, O., Roitman, H., Cohen, D., Weiner, B., Mass, Y., Rivlin, O. et al. (2019). A summarization system for scientific documents. *arXiv preprint arXiv:1908.11152*. [Cited on page 23.]

Erkan, G. & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, *22*, 457–479. [Cited on pages 20, 21, 30, 107, 108, and 122.]

Felber, T. & Kern, R. (2017). Graz university of technology at cl-scisumm 2017: Query generation strategies. In *BIRNDL@ SIGIR (2)*, pp. 67–72. [Cited on page 87.]

Ferrés, D., Saggion, H., Ronzano, F., & Bravo, À. (2018). Pdfdigest: an adaptable layout-aware pdf-to-xml textual content extractor for scientific articles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. [Cited on page 39.]

Fisas Elizalde, B., Ronzano, F., & Saggion, H. (2016). A multi-layered annotated corpus of scientific papers. In *Calzolari N, Choukri K, Declerck T, Goggi S, Grobelnik M, Maegaard B, Mariani J, Mazo H, Moreno A, Odijk J, Piperidis S, editors. LREC 2016. Tenth International Conference*

*on Language Resources and Evaluation; 2016 May 23-28; Portorož, Slovenia.[Paris]: ELRA; 2016. p. 3081-8.* ELRA (European Language Resources Association). [Cited on pages 34 and 69.]

Gehrmann, S., Deng, Y., & Rush, A. M. (2018). Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*. [Cited on page 119.]

Glorot, X. & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. [Cited on page 125.]

Gormley, C. & Tong, Z. (2015). *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.". [Cited on page 38.]

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., & He, K. (2017). Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*. [Cited on page 126.]

Hashimoto, H., Shinoda, K., Yokono, H., & Aizawa, A. (2017). Automatic generation of review matrices as multi-document summarization of scientific papers. In *BIRNDL@ SIGIR (1)*, pp. 69–82. [Cited on page 22.]

He, R., Liu, Y., Qin, B., Liu, T., & Li, S. (2008). Hitir's update summary at tac 2008: Extractive content selection for language independence. In *TAC*. [Cited on pages XXI and 50.]

Hearst, M. A. (1997). Text tiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, *23*(1), 33–64. [Cited on page 19.]

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pp. 1693–1701. [Cited on page 31.]

Herrmannova, D. & Knoth, P. (2016). An analysis of the microsoft academic graph. *D-Lib Magazine*, *22*(9/10). [Cited on page 37.]

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*. [Cited on page 124.]

Hirst, G. & St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database*, *305*, 305–332. [Cited on page 71.]

Hoang, C. D. V. & Kan, M.-Y. (2010). Towards automated related work summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 427–435. Association for Computational Linguistics. [Cited on pages 15, 27, 28, 30, 35, 36, and 38.]

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50–57. ACM. [Cited on page 29.]

Hu, Y. & Wan, X. (2014). Automatic generation of related work sections in scientific papers: An optimization approach. In *EMNLP*, pp. 1624–1633. [Cited on pages 15, 29, and 133.]

Jaidka, K., Chandrasekaran, M. K., Elizalde, B. F., Jha, R., Jones, C., Kan, M.-Y., Khanna, A., Molla-Aliod, D., Radev, D. R., Ronzano, F., & Saggion, H. (2014a). The computational linguistics summarization pilot task. In *Proceedings of TAC 2014*. [Cited on pages 34 and 63.]

Jaidka, K., Chandrasekaran, M. K., Jain, D., & Kan, M. Y. (2017a). The cl-scisumm shared task 2017: results and key insights. In *Proceedings of the Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017), organized as a part of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2017)*. [Cited on pages 63 and 87.]

Jaidka, K., Chandrasekaran, M. K., Jain, D., & Kan, M.-Y. (2017b). The cl-scisumm shared task 2017: Results and key insights. *ArXiv*, *abs/1909.00764*. [Cited on page 86.]

Jaidka, K., Chandrasekaran, M. K., Jain, D., & Kan, M.-Y. (2017c). Overview of the CL-SciSumm 2017 shared task. *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries*. [Cited on page 34.]

Jaidka, K., Chandrasekaran, M. K., Jha, R., Jones, C., Kan, M.-Y., Khanna, A., Mollá-Aliod, D., Radev, D. R., Ronzano, F., Saggion, H., & Wee, W. K. (2014b). The computational linguistics summarization pilot task. In *Proceedings of TAC 2014*. [Cited on pages 35 and 115.]

Jaidka, K., Chandrasekaran, M. K., Rustagi, S., & Kan, M.-Y. (2016). Overview of the cl-scisumm 2016 shared task. In *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*, pp. 93–102. [Cited on pages 63, 64, and 76.]

Jaidka, K., Chandrasekaran, M. K., Rustagi, S., & Kan, M.-Y. (2017d). Insights from CL-SciSumm 2016: the faceted scientific document summarization shared task. *International Journal on Digital Libraries*. [Cited on page 34.]

Jaidka, K., Chandrasekaran, M. K., Rustagi, S., & Kan, M. Y. (2017e). Insights from cl-scisumm 2016: the faceted scientific document summarization shared task. *International Journal on Digital Libraries*, pp. 1–9. [Cited on page 63.]

Jaidka, K., Khoo, C., & Na, J.-C. (2013). Deconstructing human literature reviews–a framework for multi-document summarization. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pp. 125–135. [Cited on pages 7, 22, 25, 33, 107, and 136.]

Jaidka, K., Yasunaga, M., Chandrasekaran, M. K., Radev, D., & Kan, M.-Y. (2019). The cl-scisumm shared task 2018: Results and key insights. *arXiv preprint arXiv:1909.00764*. [Cited on pages 63 and 94.]

Jean, S., Cho, K., Memisevic, R., & Bengio, Y. (2014). On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*. [Cited on page 114.]

Jean, S., Firat, O., Cho, K., Memisevic, R., & Bengio, Y. (2015). Montreal neural machine translation systems for wmt'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pp. 134–140. [Cited on page 114.]

Jelinek, F., Mercer, R. L., Bahl, L. R., & Baker, J. K. (1977). Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, *62*(S1), S63–S63. [Cited on page 126.]

Jha, R., Abu-Jbara, A., & Radev, D. R. (2013). A system for summarizing scientific topics starting from keywords. In *ACL (2)*, pp. 572–577. [Cited on page 21.]

Jiang, J. J. & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*. [Cited on page 71.]

Jinha, A. E. (2010). Article 50 million: an estimate of the number of scholarly articles in existence. *Learned Publishing*, *23*(3), 258–263. [Cited on page 1.]

Joseph, M. T. & Radev, D. R. (2007). Citation analysis, centrality, and the acl anthology. *Ann Arbor*, *1001*, 48109–1092. [Cited on page 17.]

Kågebäck, M., Mogren, O., Tahmasebi, N., & Dubhashi, D. (2014). Extractive summarization using continuous vector space models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)@ EACL*, pp. 31–39. Citeseer. [Cited on page 79.]

Kalchbrenner, N. & Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1700–1709. [Cited on page 114.]

Kaplan, D., Iida, R., & Tokunaga, T. (2009). Automatic extraction of citation contexts for research paper summarization: A coreference-chain based approach. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, pp. 88–95. Association for Computational Linguistics. [Cited on page 51.]

Kaplan, D., Tokunaga, T., & Teufel, S. (2016). Citation block determination using textual coherence. *Journal of Information Processing*, *24*(3), 540–553. [Cited on page 51.]

Karimi, S., Moraes, L. F., Das, A., & Verma, R. M. (2017). University of houston@ cl-scisumm 2017: Positional language models, structural correspondence learning and textual entailment. In *BIRNDL@ SIGIR (2)*, pp. 73–85. [Cited on page 87.]

Khoo, C. S., Na, J.-C., & Jaidka, K. (2011). Analysis of the macro-level discourse structure of literature reviews. *Online Information Review*. [Cited on pages 7 and 33.]

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*. [Cited on pages 92 and 124.]

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. [Cited on page 125.]

Klampfl, S., Rexha, A., & Kern, R. (2016). Identifying referenced text in scientific publications by summarisation and classification techniques. In *Proceedings of the joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL)*, pp. 122–131. [Cited on page 77.]

Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. M. (2017). Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*. [Cited on pages 121 and 125.]

Kong, F., Ng, H. T., & Zhou, G. (2014). A constituent-based approach to argument labeling with joint inference in discourse parsing. In *EMNLP*, pp. 68–77. [Cited on page 2.]

Kovatchev, V., Martí, M. A., & Salamó, M. (2018). WARP-text: a web-based tool for annotating relationships between pairs of texts. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pp. 132–136. Santa Fe, New Mexico: Association for Computational Linguistics. [Cited on page 39.]

Kuhn, T. S. & Hawkins, D. (1963). The structure of scientific revolutions. *American Journal of Physics*, *31*(7), 554–555. [Cited on page 2.]

Lapata, M. (2003). Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 545–552. [Cited on page 138.]

Larsen, P. O. & von Ins, M. (2010). The rate of growth in scientific publication and the decline in coverage provided by science citation index. *Scientometrics*, *84*(3), 575–603. [Cited on page 1.]

Lauscher, A., Glavas, G., & Eckert, K. (2017a). Citation-based summarization of scientific articles using semantic textual similarity. In *Proc. of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2017). Tokyo, Japan (August 2017)*. [Cited on page 25.]

Lauscher, A., Glavaš, G., & Eckert, K. (2017b). University of mannheim@ clscisumm-17: Citation-based summarization of scientific articles using semantic textual similarity. In *CEUR workshop proceedings*, vol. 2002, pp. 33–42. RWTH. [Cited on page 87.]

Leacock, C. & Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, *49*(2), 265–283. [Cited on page 71.]

Li, L., Mao, L., Zhang, Y., Chi, J., Huang, T., Cong, X., & Peng, H. (2016). Cist system for cl-scisumm 2016 shared task. In *Proceedings of the joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL)*, pp. 156–167. [Cited on pages 24 and 77.]

Li, L., Zhang, Y., Mao, L., Chi, J., Chen, M., & Huang, Z. (2017). Cist@ clscisumm-17: Multiple features based citation linkage, classification and summarization. [Cited on pages 25 and 87.]

Li, X., He, Y., Meyers, A., & Grishman, R. (2013). Towards fine-grained citation function classification. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pp. 402–407. [Cited on page 52.]

Liakata, M. & Soldatova, L. (2008). Guidelines for the annotation of general scientific concepts. *Aberystwyth University, JISC Project Report http://ie-repository. jisc. ac. uk/88*. [Cited on page 34.]

Liakata, M., Soldatova, L. N. et al. (2009). Semantic annotation of papers: Interface & enrichment tool (sapient). In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pp. 193–200. Association for Computational Linguistics. [Cited on page 34.]

Liakata, M., Teufel, S., Siddharthan, A., Batchelor, C. R. et al. (2010). Corpora for the conceptualisation and zoning of scientific papers. In *LREC*. [Cited on page 5.]

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, vol. 8. Barcelona, Spain. [Cited on pages 20, 29, 65, 85, 91, and 129.]

Lin, D. (1998). An information-theoretic definition of similarity. In *ICML*, vol. 98, pp. 296–304. [Cited on page 71.]

Liu, D. & Gildea, D. (2006). Stochastic iterative alignment for machine translation evaluation. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pp. 539–546. Association for Computational Linguistics. [Cited on page 52.]

Liu, H. (2017). Sentiment analysis of citations using word2vec. *arXiv preprint arXiv:1704.00177*. [Cited on pages 45, 56, 79, 89, and 123.]

Lloret, E. & Palomar, M. (2012). Text summarisation in progress: a literature review. *Artificial Intelligence Review*, *37*(1), 1–41. [Cited on page 16.]

Lopez, P. (2009). Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *International conference on theory and practice of digital libraries*, pp. 473–474. Springer. [Cited on page 39.]

Lu, K., Mao, J., Li, G., & Xu, J. (2016). Recognizing reference spans and classifying their discourse facets. In *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*, pp. 139–145. [Cited on page 77.]

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM J. Res. Dev.*, *2*(2), 159–165. [Cited on page 15.]

Luong, M.-T. & Manning, C. D. (2015). Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pp. 76–79. [Cited on page 114.]

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*. [Cited on page 114.]

Luong, M.-T., Sutskever, I., Le, Q. V., Vinyals, O., & Zaremba, W. (2014). Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*. [Cited on page 114.]

Ma, S., Xu, J., Wang, J., & Zhang, C. (2017). Njust @ clscisumm-17. In *BIRNDL@SIGIR*. [Cited on pages 25, 87, and 95.]

Ma, S., Zhang, H., Xu, J., & Zhang, C. (2018). Njust@ clscisumm-18. In *BIRNDL@ SIGIR*. [Cited on pages 26, 27, and 95.]

Maggio, L., Sewell, J., & Artino, A. (2016). The literature review: A foundation for high-quality medical education research. *Journal of Graduate Medical Education*, *8*, 297–303. [Cited on page 7.]

Malenfant, B. & Lapalme, G. (2016). Rali system description for cl-scisumm 2016 shared task. In *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*, pp. 146–155. [Cited on page 77.]

Mancini, M., Camacho-Collados, J., Iacobacci, I., & Navigli, R. (2016). Embedding words and senses together via joint knowledge-enhanced training. *arXiv preprint arXiv:1612.02703*. [Cited on pages 46, 56, and 80.]

Maynard, D., Tablan, V., Cunningham, H., Ursu, C., Saggion, H., Bontcheva, K., & Wilks, Y. (2002). Architectural elements of language engineering robustness. *Natural Language Engineering*, *8*(2-3), 257–274. [Cited on pages 39, 42, 67, and 68.]

Mayr, P., Chandrasekaran, M. K., & Jaidka, K. (2019). Report on the 3rd joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (birndl 2018). In *ACM SIGIR Forum*, vol. 52, pp. 105–110. ACM. [Cited on pages 115 and 122.]

McNee, S. M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S. K., Rashid, A. M., Konstan, J. A., & Riedl, J. (2002). On the recommending of citations for research papers. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pp. 116–125. ACM. [Cited on page 8.]

Mei, Q., Guo, J., & Radev, D. (2010). Divrank: the interplay of prestige and diversity in information networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1009–1018. Acm. [Cited on page 21.]

Mei, Q. & Zhai, C. (2008). Generating impact-based summaries for scientific literature. *Proceedings of ACL-08: HLT*, pp. 816–824. [Cited on page 18.]

Metzler, D. & Croft, W. B. (2005). A markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 472–479. ACM. [Cited on page 18.]

Mihalcea, R. & Tarau, P. (2004a). TextRank: Bringing order into texts. In *Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing*. [Cited on pages 75 and 84.]

Mihalcea, R. & Tarau, P. (2004b). Textrank: Bringing order into texts. Association for Computational Linguistics. [Cited on pages 107, 108, 122, and 123.]

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. *ICLR Workshop*. [Cited on pages 79 and 123.]

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2018). Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*. [Cited on page 125.]

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119. [Cited on page 125.]

Mimno, D., Wallach, H., Talley, E., Leenders, M., & McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 262–272. [Cited on page 106.]

Mohammad, S., Dorr, B., Egan, M., Hassan, A., Muthukrishan, P., Qazvinian, V., Radev, D., & Zajic, D. (2009). Using citations to generate surveys of scientific paradigms. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 584–592. Association for Computational Linguistics. [Cited on pages 18 and 28.]

Moraes, L., Baki, S., Verma, R., & Lee, D. (2016). University of houston at cl-scisumm 2016: Svms with tree kernels and sentence similarity. In *proceedings of the joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL)*, pp. 113–121. [Cited on pages 25 and 77.]

Moro, A., Raganato, A., & Navigli, R. (2014). Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)*, *2*, 231–244. [Cited on page 57.]

Munroe, R. (2013). The rise of open access. *Science*, *342*(6154), 58–59. [Cited on page 1.]

Nakov, P., Schwartz, A., & Hearst, M. (2004a). Citances: Citation sentences for semantic analysis of bioscience text. [Cited on page 64.]

Nakov, P. I., Schwartz, A. S., & Hearst, M. (2004b). Citances: Citation sentences for semantic analysis of bioscience text. In *Proceedings of the SIGIR'04 workshop on Search and Discovery in Bioinformatics*, pp. 81–88. [Cited on page 16.]

Nallapati, R., Xiang, B., & Zhou, B. (2016a). Sequence-to-sequence rnns for text summarization. [Cited on page 6.]

Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B. et al. (2016b). Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*. [Cited on pages 113 and 114.]

Nallapati, R. M., Ahmed, A., Xing, E. P., & Cohen, W. W. (2008). Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 542–550. ACM. [Cited on page 28.]

Nanba, H., Kando, N., & Okumura, M. (2011). Classification of research papers using citation links and citation types: Towards automatic review article generation. *Advances in Classification Research Online*, *11*(1), 117–134. [Cited on page 20.]

Nanba, H. & Okumura, M. (1999). Towards multi-paper summarization using reference information. In *IJCAI*, vol. 99, pp. 926–931. [Cited on page 51.]

Navigli, R. & Ponzetto, S. P. (2010). Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pp. 216–225. Association for Computational Linguistics. [Cited on page 43.]

Navigli, R. & Ponzetto, S. P. (2012a). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, *193*, 217–250. [Cited on pages 56, 68, 80, and 99.]

Navigli, R. & Ponzetto, S. P. (2012b). Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.*, *193*, 217–250. [Cited on page 69.]

Nenkova, A., Passonneau, R., & McKeown, K. (2007). The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing (TSLP)*, *4*(2), 4. [Cited on page 50.]

Nenkova, A. & Passonneau, R. J. (2004). Evaluating content selection in summarization: The pyramid method. In *HLT-NAACL*, vol. 4, pp. 145–152. Citeseer. [Cited on page 21.]

Nguyen, T. H. & Grishman, R. (2015). Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 39–48. [Cited on pages 92 and 124.]

Nomoto, T. (2016). Neal: A neurally enhanced approach to linking citation and reference. In *Proceedings of the joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL)*, pp. 168–174. [Cited on page 77.]

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics. [Cited on page 117.]

Pautasso, M. (2013). Ten simple rules for writing a literature review. *PLoS computational biology*, *9*, e1003149. [Cited on page 33.]

Pramanick, A., Mandi, S., Dey, M., & Das, D. (2017). Scisumm 2017: Employing word vectors for identifying, classifying and summarizing scientific documents. [Cited on page 87.]

Prasad, A. (2017). Wing-nus at cl-scisumm 2017: Learning from syntactic and semantic similarity for citation contextualization. In *BIRNDL@ SIGIR (2)*, pp. 26–32. [Cited on page 87.]

Qazvinian, V. & Radev, D. R. (2008a). Scientific paper summarization using citation summary networks. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pp. 689–696. Stroudsburg, PA, USA: Association for Computational Linguistics. [Cited on page 5.]

Qazvinian, V. & Radev, D. R. (2008b). Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pp. 689–696. Association for Computational Linguistics. [Cited on pages 17, 20, 21, 35, and 64.]

Qazvinian, V. & Radev, D. R. (2010a). Identifying non-explicit citing sentences for citation-based summarization. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pp. 555–564. [Cited on page 5.]

Qazvinian, V. & Radev, D. R. (2010b). Identifying non-explicit citing sentences for citation-based summarization. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pp. 555–564. Association for Computational Linguistics. [Cited on pages 18, 19, and 51.]

Qazvinian, V., Radev, D. R., Mohammad, S., Dorr, B. J., Zajic, D. M., Whidby, M., & Moon, T. (2013). Generating extractive summaries of scientific paradigms. *J. Artif. Intell. Res.(JAIR)*, *46*, 165–201. [Cited on page 21.]

Qazvinian, V., Radev, D. R., & Özgür, A. (2010). Citation summarization through keyphrase extraction. In *Proceedings of the 23rd international conference on computational linguistics*, pp. 895–903. Association for Computational Linguistics. [Cited on page 51.]

Radev, D. R., Allison, T., Blair-Goldensohn, S., Blitzer, J., Celebi, A., Dimitrov, S., Drabek, E., Hakim, A., Lam, W., Liu, D. et al. (2004). Mead-a platform for multidocument multilingual text summarization. In *LREC*. [Cited on pages 19, 20, 29, 30, 107, and 122.]

Radev, D. R., Muthukrishnan, P., Qazvinian, V., & Abu-Jbara, A. (2013). The acl anthology network corpus. *Language Resources and Evaluation*, *47*(4), 919–944. [Cited on pages 37 and 38.]

Radev, D. R. & Tam, D. (2003). Summarization evaluation using relative utility. In *Proceedings of the twelfth international conference on Information and knowledge management*, pp. 508–511. ACM. [Cited on page 22.]

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. arxiv preprint cmplg/9511007. [Cited on page 71.]

Ronzano, F. & Saggion, H. (2015). Dr. Inventor Framework: Extracting structured information from scientific publications. In *International Conference on Discovery Science*, pp. 209–220. Springer. [Cited on pages 43, 57, 68, 69, and 123.]

Ronzano, F. & Saggion, H. (2016). An empirical assessment of citation information in scientific summarization. In *International Conference on Applications of Natural Language to Information Systems*, pp. 318–325. Springer. [Cited on page 18.]

Rowley, J. & Slack, F. (2004). Conducting a literature review. *Management Research News*, *27*. [Cited on page 7.]

Rush, A. M., Chopra, S., & Weston, J. (2015). A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*. [Cited on page 113.]

Rush, A. M., Harvard, S., Chopra, S., & Weston, J. (2017). A neural attention model for sentence summarization. In *ACLWeb. Proceedings of the 2015 conference on empirical methods in natural language processing*. [Cited on page 6.]

Saggion, H. (1999). Using linguistic knowledge in automatic abstracting. In *27th Annual Meeting of the Association for Computational Linguistics, University of Maryland, College Park, Maryland, USA, 20-26 June 1999*. [Cited on pages 107 and 136.]

Saggion, H. (2008a). A robust and adaptable summarization tool. *Traitement Automatique des Langues*, *49*(2). [Cited on pages 68, 122, 123, and 124.]

Saggion, H. (2008b). SUMMA. A Robust and Adaptable Summarization Tool. *TAL*, *49*(2), 103–125. [Cited on pages 43, 107, and 108.]

Saggion, H. (2008c). SUMMA: A robust and adaptable summarization tool. *Traitement Automatique des Langues*, *49*(2). [Cited on page 56.]

Saggion, H. (2008d). SUMMA: A Robust and Adaptable Summarization Tool. *Traitement Automatique des Langues*, *49*(2), 103–125. [Cited on pages 69 and 73.]

Saggion, H. (2011). Learning predicate insertion rules for document abstracting. In *Computational Linguistics and Intelligent Text Processing - 12th International Conference, CICLing 2011, Tokyo, Japan, February 20-26, 2011. Proceedings, Part II*, pp. 301–312. [Cited on page 7.]

Saggion, H. (2014). Creating summarization systems with SUMMA. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014.*, pp. 4157–4163. [Cited on pages 69 and 73.]

Saggion, H., AbuRa'ed, A., & Ronzano, F. (2016a). Trainable citation-enhanced summarization of scientific articles. In *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL) co-located with the Joint Conference on Digital Libraries 2016 (JCDL 2016), Newark, NJ, USA, June 23, 2016.*, pp. 175–186. [Cited on pages 69 and 91.]

Saggion, H., AbuRa'ed, A., & Ronzano, F. (2016b). Trainable citation-enhanced summarization of scientific articles. In *Cabanac G, Chandrasekaran MK, Frommholz I, Jaidka K, Kan M, Mayr P, Wolfram D, editors. Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL); 2016 June 23; Newark, United States.[place unknown]: CEUR Workshop Proceedings; 2016. p. 175-86.* CEUR Workshop Proceedings. [Cited on page 12.]

Saggion, H. & Lapalme, G. (2002a). Generating indicative-informative summaries with sumum. *Computational linguistics*, *28*(4), 497–526. [Cited on pages 5 and 16.]

Saggion, H. & Lapalme, G. (2002b). Generating indicative-informative summaries with sumum. *Computational Linguistics*, *28*(4), 497–526. [Cited on page 34.]

Saggion, H. & Poibeau, T. (2013). Automatic text summarization: Past, present and future. In *Multi-source, Multilingual Information Extraction and Summarization*, pp. 3–21. [Cited on page 16.]

Schuster, M. & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681. [Cited on page 119.]

See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*. [Cited on pages 5, 13, and 119.]

Siddharthan, A. & Teufel, S. (2007). Whose idea was this, and why does it matter? attributing scientific work to citations. In *Human language technologies 2007: The conference of the North American chapter of the Association for Computational Linguistics; proceedings of the main conference*, pp. 316–323. [Cited on page 55.]

Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B.-j. P., & Wang, K. (2015). An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*, pp. 243–246. ACM. [Cited on pages 37 and 115.]

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958. [Cited on page 125.]

Sujatha, P. & Dhavachelvan, P. (2011). Precision at k in multilingual information retrieval. [Cited on page 46.]

Sun, X. & Zhuge, H. (2018). Summarization of scientific paper through reinforcement ranking on semantic link network. *IEEE Access*, *6*, 40611–40625. [Cited on page 23.]

Surya, S., Mishra, A., Laha, A., Jain, P., & Sankaranarayanan, K. (2019). Unsupervised neural text simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2058–2068. [Cited on page 133.]

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112. [Cited on page 114.]

Tang, J. (2016). Aminer: Toward understanding big scholar data. In *WSDM*. [Cited on page 116.]

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., & Su, Z. (2008). Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 990–998. ACM. [Cited on pages 37 and 115.]

Teufel, S. (2000). *Argumentative zoning: Information extraction from scientific text*. Ph.D. thesis, Citeseer. [Cited on pages 57, 68, 116, and 127.]

Teufel, S. (2006). Argumentative zoning for improved citation indexing. In *Computing attitude and affect in text: Theory and Applications*, pp. 159–169. Springer. [Cited on page 34.]

Teufel, S. & Moens, M. (2002). Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, *28*(4), 409–445. [Cited on pages 5, 16, 43, 75, and 123.]

Teufel, S., Siddharthan, A., & Batchelor, C. (2009). Towards discipline-independent argumentative zoning: evidence from chemistry and computational linguistics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pp. 1493–1502. Association for Computational Linguistics. [Cited on page 5.]

Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013). Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *Proc. of KDD-2013*, pp. 847–855. [Cited on page 58.]

Turney, P. D. (2002). Learning to extract keyphrases from text. *arXiv preprint cs/0212013*. [Cited on page 16.]

Valenzuela, M., Ha, V. A., & Etzioni, O. (2015). Identifying meaningful citations. In *AAAI Workshop: Scholarly Big Data*. [Cited on page 37.]

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008. [Cited on pages 114, 119, and 120.]

Venugopal, A., Zollmann, A., Smith, N. A., & Vogel, S. (2009). Preference grammars: Softening syntactic constraints to improve statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 236–244. [Cited on page 36.]

Vinyals, O., Fortunato, M., & Jaitly, N. (2015a). Pointer networks. In *Advances in Neural Information Processing Systems*, pp. 2692–2700. [Cited on page 114.]

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015b). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164. [Cited on page 114.]

Vu, H. C. D. (2010). *Towards automated related work summarization*. Ph.D. thesis. [Cited on page 27.]

Wade, A. D. (2015). Overview of microsoft academic graph. *Alonso et al.[2]*, p. 8. [Cited on page 37.]

Wang, P., Li, S., Wang, T., Zhou, H., & Tang, J. (2018). Nudt@ clscisumm-18. [Cited on page 95.]

White, L., Togneri, R., Liu, W., & Bennamoun, M. (2015). How well sentence embeddings capture meaning. In *Proceedings of the 20th Australasian Document Computing Symposium*, p. 9. ACM. [Cited on page 79.]

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 3rd edn. [Cited on page 70.]

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann. [Cited on page 81.]

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K. et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*. [Cited on pages 13 and 119.]

Wu, Z. & Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pp. 133–138. Association for Computational Linguistics. [Cited on page 71.]

Xiong, C., Power, R., & Callan, J. (2017). Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web*, pp. 1271–1279. International World Wide Web Conferences Steering Committee. [Cited on page 37.]

Xu, H., Wang, Z., & Weng, X. (2019). Scientific literature summarization using document structure and hierarchical attention model. *IEEE Access*, *7*, 185290–185300. [Cited on page 23.]

Yasunaga, M., Kasai, J., Zhang, R., Fabbri, A., Li, I., Friedman, D., & Radev, D. (2019a). ScisummNet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. Proceedings of AAAI 2019. [Cited on page 33.]

Yasunaga, M., Kasai, J., Zhang, R., Fabbri, A., Li, I., Friedman, D., & Radev, D. (2019b). ScisummNet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of AAAI 2019*. [Cited on page 115.]

Yasunaga, M., Kasai, J., Zhang, R., Fabbri, A. R., Li, I., Friedman, D., & Radev, D. R. (2019c). Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 7386–7393. [Cited on page 115.]

Yih, W.-t. & Qazvinian, V. (2012). Measuring word relatedness using heterogeneous vector space models. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 616–620. Association for Computational Linguistics. [Cited on page 22.]

Zajic, D., Dorr, B. J., Lin, J., & Schwartz, R. (2007). Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing & Management*, *43*(6), 1549–1570. [Cited on page 21.]

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*. [Cited on page 92.]

Zeng, W., Luo, W., Fidler, S., & Urtasun, R. (2016). Efficient summarization with read-again and copy mechanism. *arXiv preprint arXiv:1611.03382*. [Cited on page 113.]

Zhang, D. & Li, S. (2017). Pku@ clscisumm-17: Citation contextualization. In *BIRNDL@ SIGIR (2)*, pp. 86–93. [Cited on page 87.]

Zhang, J., Li, K., Yao, C., & Sun, Y. (2019). Event-based summarization method for scientific literature. *Personal and Ubiquitous Computing*, pp. 1–10. [Cited on page 23.]

Zhang, W., Itoh, K., Tanida, J., & Ichioka, Y. (1990). Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Applied optics*, *29*(32), 4790–4797. [Cited on pages 87, 88, and 100.]

Zhang, X., Lapata, M., Wei, F., & Zhou, M. (2018). Neural latent extractive document summarization. *arXiv preprint arXiv:1808.07187*. [Cited on page 31.]