4-13-2017 12:00 AM

Investigating Citation Linkage Between Research Articles

Kokou Hospice Houngbo, *The University of Western Ontario*

Supervisor: Robert E. Mercer, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Computer Science

Abstract

In recent years, there has been a dramatic increase in scientific publications across the globe. To help navigate this overabundance of information, methods have been devised to find papers with related content, but they are lacking in the ability to provide specific information that a researcher may need without having to read hundreds of linked papers. The search and browsing capabilities of online domain specific scientific repositories are limited to finding a paper citing other papers, but do not point to the specific text that is being cited. Providing this capability to the research community will be beneficial in terms of the time required to acquire the amount of background information they need to undertake their research. In this thesis, we present our effort to develop a citation linkage framework for finding those sentences in a cited article that are the focus of a citation in a citing paper. This undertaking has involved the construction of datasets and corpora that are required to build models for focused information extraction, text classification and information retrieval. As the first part of this thesis, two preprocessing steps that are deemed to assist with the citation linkage task are explored: method mention extraction and rhetorical categorization of scientific discourse. In the second part of this thesis, two methodologies for achieving the citation linkage goal are investigated. Firstly, regression techniques have been used to predict the degree of similarity between citation sentences and their equivalent target sentences with medium Pearson correlation score between predicted and expected values. The resulting learning models are then used to rank sentences in the cited paper based on their predicted scores. Secondly, search engine-like retrieval techniques have been used to rank sentences in the cited paper based on the words contained in the citation sentence. Our experiments show that it is possible to find the set of sentences that a citation refers to in a cited paper with reasonable performance. Possible applications of this work include: creation of better science paper repository navigation tools, development of scientific argumentation across research articles, and multi-document summarization of science articles.

Keywords: Citation linkage, machine learning, information extraction, data mining, text classification, text matching, text similarity detection, corpus building techniques, information retrieval

Acknowledgements

This work has been a challenging, yet an enjoyable journey that has only been possible with the help and support of remarkable people in the academic setting as well as in the familial circle.

My deepest thanks go to my supervisor who has guided me throughout this work and has provided useful insights and constructive criticism when needed.

I thank the University of Western Ontario for the financial support and valuable assistance provided through the various graduate teachers who have been involved in my study.

My family members have been there for me each step of the way, and I am grateful for that. I would like to dedicate this work to a special person who is no longer in this world to rejoice with me at the end of this road. All glory be to God!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Writing across different genres has varying purposes and methods to accomplish various objectives. An academic research paper is distinguished by its being part of a research mosaic. In research literature, the writer is obliged to place the research contribution of the current article in its research context. The method to achieve this placement is to refer explicitly to other research works. References to other works are manifested as *citations*. Citations are thus an important instrument for connecting the ideas in the research literature. This importance has led to a variety of tools to assist researchers. For instance, citation indexes, an idea conceived in 1964 [53], contain a subset of all of the citations in research articles. More recently, methods have been proposed to classify the purpose of a citation [56, 174]. Citation analysis-based bibliometrics are used to assess research and researcher importance [54, 55]. Potential uses of citations include multiple article summarization [33, 34, 148] and the tracking of scientific argumentation [120, 139] across multiple papers.

When considering academic science research articles, some of the more sophisticated uses of citations are not immediately realizable because of the coarse granularity of the citation itself. Citations, when used in science research papers, cite papers. Having a span of text in the referenced article which is significantly smaller than a complete paper, a paragraph or a sentence or a set of sentences, say, would be beneficial for some of the more complex applications mentioned above. Other reasons for having small spans of text will be mentioned below.

This inducement for reducing citation targets from papers to briefer spans of text provides the motivation for the topic of this thesis. This thesis aims at investigating how to determine the target sentences that citation sentences in science research papers refer to in a given cited paper. We call this operation *citation linkage* and believe that this is a very practical problem that can be solved using machine learning and information retrieval techniques.

The central results provided by this thesis are the outcomes of testing the machine learning

and information retrieval techniques on an experimental corpus. Intermediate steps to achieving this goal involve an overview of the experimental corpus, machine learning and information retrieval methodologies, feature engineering techniques, as well as the choice of adequate evaluation measures. In this introductory overview, we present the foundation for this study and the potential challenges that are brought about. Included as part of this thesis are two preprocessing steps that are deemed to assist with the citation linkage task: method mention extraction and rhetorical categorization of scientific discourse. This thesis then presents the use of the two methodologies on the experimental corpus and an evaluation of the results.

1.1 Scientific Publication

A research paper is often supported by many references to previous research works. In scientific publications, researchers want to inform the audience comprising other scientists regarding important issues about their experiments by documenting the different techniques they use during scientific investigations. In reading a scientific paper, most researchers want to get concise information about what has been discovered and reported in the paper. For instance, they might be concerned about getting a much focused aspect of the supporting background information derived from related cited papers. Presenting readers with this focused information will enable them to easily and effectively assimilate more research material. In our present research work, we thus attempt to link the citing sentences to the exact set of sentences they refer to in the cited paper. What is often cited from previous works may be related to the methodology, tools, techniques and set of protocols that the current paper may have adapted or modified.

When a reference is used in a scientific paper, it is accompanied by a span of text that highlights a given aspect of the work described in the cited paper. This span of text—often called the citation context—generally points to a specific topic or idea mentioned in the cited paper. The intention of the writer is to show that there is some relevant background information that the reader needs to get accustomed to in order to understand the content expanded in the ongoing paper. Reading this background information might be sometimes necessary to grasp the meaning of the idea that is being expressed in the paper. Most of the time, the content of citations is expressed in full sentences containing specific terminology that is present in the cited paper. For instance, such terminology can represent a set of methods or tools used during scientific experiments. An author might cite a paper because she adapts or modifies the methods or techniques used in that paper. Linking citation sentences of this kind to their exact places in the cited papers could help researchers focus on relevant information cited in the paper, and thus reduce the amount of time needed to understand a given research material.

The text in which a citation occurs can span one or more sentences in the paper. In this

study, this span is limited to one sentence, and the linkage task is assumed to be a sentence-level matching operation. We believe that our proposed techniques could be applied to most types of citation sentences, but due to time constraints, we want to look at method sentences. Another motivation is that such sentences have the potential of containing methodology terms that can be extracted to build domain specific linguistic resources. To achieve our goal, we need to define an adequate framework that comprises the building of diverse corpora and the choice of adequate evaluation techniques.

1.2 Significance and Potential Benefits

We have hypothesized that linking citation sentences to the body of text they refer to in the cited papers will have the following benefits to the research community:

- It can assist the navigation among research papers
- It can improve information retrieval
- It can be used in the task of citation categorization
- It can be useful in the task of paraphrase detection
- It can be used in the task of focused multi-document summarization

The results of this study can be integrated into an information system for argumentation in a network of research papers. If we find the real span of text that is being cited, the argumentation network will become more precise. In fact, rather than saying that a paper *X* supports a paper *Y*, we can be certain about what kind of methods or techniques are common to both papers or whether the citing paper's technique is a modification of the one used in the cited paper.

1.3 Thesis Feasibility Study

In scientific articles, citations serve to establish a semantic link with a body of knowledge, generally presented in previously published papers. This can be done explicitly using words and expressions from the original paper, or implicitly by referring to common background information inherent to the domain of study. Either way, when paper *A* cites paper *B*, we can expect that both papers will share an idea that can be traced back in the cited paper.

The following examples show diverse forms that a citation and a target sentence can take:

Example 1

Citation Sentence: *"Formalin fixation, the most widely used fixative in histopathology, has many advantages such as the ease of tissue handling, the possibility of long-term storage, an optimal histological quality and its availability in large quantities at low price [15,16]."*

Target Sentence: *"The advantages of formalin fixation are the ease of tissue handling, the possibility of long-term storage of wet material, and its low price."*

In Example 1, we can notice that the citation sentence is a paraphrase of the original text and we can also notice that most of the content words are shared between the citation sentence and the target sentence. A linkage between these sentences could then be done at the word surface level.

Example 2

Citation Sentence: *"During the processes of paraffin embedding, sectioning and further analysis by (real time) PCR, small traces of foreign DNA, e.g. introduced by floater tissue or a contaminated microtome blade, may contaminate the material under investigation thereby possibly influencing interpretation of results [41,42]."*

Target Sentence: *"Because PCR is so sensitive, it is subject to false positives resulting from the introduction of DNA from a contaminating source."*

In Example 2, we can notice that a deeper analysis is required to be able to link both sentences. As humans, we may understand that the expression "possibly influencing interpretation of results" is a synonym for the expression "is subject to false positives". For the matching to be successful we need to introduce some world knowledge information for the computer to be able to interpret the meaning of the sentences.

Example 3

Citation Sentence: *"In formalin fixed tissue several factors affect the degradation of DNA, including the duration of fixation, pH, salt concentration, and temperature.*[1–8]*"*

Target Sentence: *"The overall rate of formalin-induced modification of DNA is dependent on the concentration, temperature, and pH of the fixative."*

In **Example 3**, different keywords are used to point to many *concepts* used in the cited paper. When each of the "*factors*" affecting the degradation of DNA have been referred to in the citing paper, we notice that the citing sentence is just a summary of what has actually been developed in the cited paper. We can notice the use of different concepts such as *pH*, *concentration*, *temperature* and different values affected to them. These concepts are independently developed in some cited papers and summarised in others. A deeper analysis of the sentences is thus necessary for a better linkage.

Example 4

Citation Sentence: *"Sample DNA is often damaged by exposure to formaldehyde and a potentially extremely acidic environment[11,12]."*

Target Sentence: *"However, DNA is relatively stable in mildly acidic solutions, but at around pH*4 *the β glycosidic bonds in the purine bases are hydrolysed."*

In Example 4, the term "extremely acidic environment" has been used to show how a certain level of *pH* values can influence the state of the extracted DNA. A good link between the information in the citation sentences and what they refer to in the cited paper could only be established if we know how the "scaling" information is presented in the source paper and how humans paraphrase such information. There is thus the need to build a paraphrase ontology that may take into account the different forms of rephrasing.

Example 5

Citation Sentence: *"Different PCR buffer systems and/or different Taq polymerases may yield different real time PCR results [26,27]."*

Target Sentence: *"A significant difference can be seen between the results from the different DNA polymerase-buffer systems."*

From the above sentences in Example 5, we might have many interpretations of what the message being conveyed by the author is. The "different results" he/she is referring to can be inferred as: "different" techniques may have been used resulting in the same conclusions; or different techniques resulting in different conclusions.

In conclusion, to be able to achieve a correct linkage we should take into account various difficulty levels, ranging from word level matching to a broader inferential deduction. However, encoding all these requirements into a linkage system is a challenging task that may become intractable if the problem is not narrowed than to a more manageable size.

First of all, a citation linkage operation will primarily aim at determining the similarity between two units of text. Therefore, finding the appropriate level of granularity at which the linkage should be performed will amount to deciding the size of the texts that needs to be compared. For this purpose, we need to decide the boundary of the citation text as well as that of the target cited text.

In this study, we assume that the units of text that need to be matched should primarily be sentences, because most current text applications operate at the sentence level. Thus, the linkage task is a matching operation between a given citation sentence, or part of it, and individual sentences in a cited paper.

Also, we hypothesize that citation sentences that comprise method terminologies used during scientific experiments might likely be linked to sentences of the same category in the cited paper. Therefore, for the problem to be tractable, we intend to limit our current work to the citation sentences belonging to this rhetorical category.

Throughout this work, we have tried to answer the following questions in the subsequent parts of the thesis:

- What is the adequate framework for linking method context citation sentences with sentences they refer to in a cited research article?
- What are the computational linguistic and machine learning techniques that can be used to successfully link citation sentences to the text they refer to in the cited papers?
- Can the linkage task be viewed as a machine learning problem?
- Can the linkage task be viewed as a text retrieval problem?

1.4 Thesis Goal and Methodology

The ultimate goal of this study is to develop techniques for linking citation sentences and their matching cited sentences. This goal can be achieved in several steps, each dealing individually with one objective of the study.

1.4.1 Objective 1 - Identify and Build Citation Linkage Datasets

Most text-based applications require the use of appropriate datasets (corpora) that need to be annotated by domain experts. As there are no suitable datasets for this study, our initial objective is to define guidelines for building such resources. We propose three datasets to be used for the following purposes:

- The first dataset is used to evaluate the task of information extraction presented in Chapter 4. This task aims at providing a framework for recognizing method mentions in biomedical texts. To build this dataset, we use a linguistic-based heuristic to select sentences that have a high probability to belong to the method sentence category and then manually curated these sentences for method mention detection. One characteristic of the dataset is that it requires only a partial intervention by human annotators.

- The second dataset is required to build a model for rhetorical classification in the biomedical domain. This dataset is used in Chapter 6. We hypothesize that the target sentence will likely belong to the same rhetorical category as the citation sentence. We also hypothesize that method sentences may be less linguistically variable than the other scientific rhetorical categories. So, recognizing sentences of the method category might help reduce the target sentence pool to this class.

  The corpus is built automatically by extracting sentences from a large repository of full-text scientific research papers—the PubMed repository [23]. We assume that sentences that appear in the co-referential context of a co-referencing sentence beginning with "This method", will likely talk about a methodology used in a research experiment and reported in the paper. Similarly, a sentence that starts with the expression "This result" is likely to refer to an experimental result context, etc. The co-referential context is limited to the sentence that appears before the anaphoric reference "This...".

- The third dataset is constructed to evaluate the task of citation linkage in scientific research papers. It consists of 22 scientific articles annotated by domain experts. Sentences in cited articles are given a ranking from 0 to 5, to show the degree of similarity with the citation sentence. These papers are chosen from a directed citation network such that a paper A containing the focus citation sentence will point to a target paper B containing the candidate cited sentences.

1.4.2 Objective 2 - Identify Method Mentions from Scientific Sentences

In this study, the linkage task is domain oriented as we target citations that make mention of method keywords and terminologies. We use rule-based and machine learning techniques to automatically extract method terminologies from method sentences. One motivation is that scientific publications contain many references to method terminologies used during scientific experiments and citation sentences of the method category will likely contain relevant methodology terms presented in the cited paper. A linkage experiment can then target the method mention keywords instead of the whole citation sentence. We focus our study on the extraction

of method phrases that contain an explicit mention of method keywords such as *algorithm*, *technique*, *analysis*, *approach* and *method* and other less explicit method terms such as ''*Multiplex Ligation dependent Probe Amplification*".

Our results show that we can extract many of these terms using grammatical patterns. A few other terms can be extracted with machine learning techniques. A brief study of the corpus shows that the context of the method mentions can help in the extraction of important information about the method term.

1.4.3 Objective 3 - Classify Sentences in Scientific Rhetorical Categories

We hypothesize that citation linkage should primarily be viewed as the matching of units of text that convey meaningful information when taken in isolation. An underlying assumption that will reduce the scope of the citation linkage task, and investigated in later chapters, is that citation sentences that make (contain a) reference to method mentions will likely be linked to sentences with similar method terms in the cited paper. So classifying sentences into a set of rhetorical categories that include the method class could reduce the pool of possible target sentences and thereby may be of great value for the citation linkage task. To allow sentence level rhetorical classification of scientific discourse to be a step in the citation linkage task, we need to build a rhetorical classifier because none are publicly available. Feature selection methodologies such as Chi-Squared and Mutual Information were used to automatically select relevant features.

1.4.4 Objective 4 - Investigate Citation Linkage as a Machine Learning Task

The linkage between the citation text and the target text it refers to in the cited paper requires the choice of appropriate text comparison techniques that evaluate the degree of similarity between two spans of textual content. From this perspective and with our choice of the span of text to be a sentence, finding the best matching cited sentences for a given citation sentence will involve the use of one or more text similarity measures to discriminate among the candidate target sentences. A dataset is built by computing the feature values for the pair (citation sentence, candidate target sentence) for every candidate target sentence. We have chosen two sets of candidate target sentences, all sentences of the referenced paper and all sentences categorized as method sentences in the referenced paper, and built the two respective datasets. Each feature denotes a notion of the similarity that may exist between units of text at a surface level or at a more complex semantic level. Many similarity/distance measures are investigated individually

and then combined into a feature pool to build a machine learning model for text similarity detection.

1.4.5 Objective 5 - Investigate Citation Linkage as an Information Retrieval Task

We hypothesize that citation sentences, when used as queries in a given retrieval model, should likely point to relevant sentences in the cited paper. For this purpose, we used many established retrieval algorithms to check whether the task of citation linkage targeting sentences in a cited paper could be performed by ranking the sentences as would do a search engine. We define the domain of a given search result to be all sentences in the cited paper or limited to the method sentences in the cited paper. We also propose two sources for the retrieval query: the full citation sentence and the noun phrases that are extracted from the citation sentence. Therefore, the task of citation linkage as a text retrieval operation is confined to the appropriate set of target sentences in a single paper and a query is a citation sentence or part of it targeting a cited paper.

1.5 Contributions of this Thesis

- We have provided the Computational Linguistics research community with an initial corpus of biochemical research articles annotated with citation target sentences by a graduate student with biochemistry knowledge. See Appendix A and Appendix B for details.

- We have investigated a methodology for building self-annotated and machine validated corpora [73]. Using this methodology, we have provided the Computational Linguistics research community with an initial corpus annotated with three of the four scientific rhetorical categories. See Appendix A for details.

- We have developed a rule-based and a machine learning model to extract method mentions from science articles. Our initial results show an average F-score of 91.89 for the rule-based system and 78.26 for the Conditional Random Field-based machine learning system [72].

- We have developed a learning model to classify sentences from scientific research articles in the scientific rhetorical categories. In a 10-fold cross-validation experiment, we obtain an overall F-score of 0.97 with Naïve Bayes and 0.987 with SVM.

- We have investigated machine learning techniques for text matching applied to the citation linkage context. We test with three regression models using a feature pool comprising many similarity measures, to predict the degree of similarity between citation sentences and their possible target sentences. These experiments prove to suffer from a certain number of issues such as the unbalanced nature of the dataset and the presence of many outliers. However, the dataset reduced to the "Method" rhetorical category has yielded a medium Pearson Correlation value of 0.4217. When we use the learned models to rank the predicted scores for sentences, the Average *NDCG@k* is 30% and the Average *Precision@k* is 29% over all the papers for experiments using full paper sentences as candidate referenced sentences. The Average *NDCG@k* is 41% and the Average *Precision@k* is 42% over all the papers for experiments using Method sentences as candidate referenced sentences. For all ranked sentences, 51% and 46% of the sentences human-rated as 5 and 4 are ranked in the top position, respectively. We can notice that fewer sentences human-rated by the annotators as 5 and 4 have been ranked as a 0 compared to those human-rated as 3, 2 and 1. This may mean that the sentences human-rated as 1, 2, and 3 may also require biochemical knowledge and more background information in the ranking model for the linkage to be effective.

- We have investigated information retrieval techniques for citation linkage. This has proved to be the best way to approach the linkage task. We use search engine-like retrieval methods to rank sentences in the cited paper based on the information contained in the citation sentence. We evaluate the results on a binary-based metric using *Precision@k* as well as on the multi-level utility score using Normalized Discounted Cumulative Gain at rank *k* (*NDCG@k*). The best *Precision@k* is 1.00 (100%), and the best *NDCG@k* score is 1.00 (100%), both occurring when Method sentences are used as candidate referenced sentences. For all ranked sentences, 60% and 44% of the sentences human-rated as 5 and 4 are ranked in the top positions, respectively. The ranking for the sentences human-rated as 5 has improved over the results presented in Chapter 8.

1.5.1 Outline of the thesis

The remaining chapters of the thesis are structured as follow:
Chapter 2 reviews related works and presents an overview of the different areas that the task of citation linkage involves. The related areas are classified into five main categories: citation analysis in research papers, annotation schemes for scientific articles, sentence level information extraction from science articles, text classification techniques, and text similarity techniques.

Chapter 3 presents the citation linkage datasets and different annotation processes.
Chapter 4 shows how domain specific information can be extracted from a citation linkage corpus in order to provide focus information to users.
Chapter 5 presents the different feature selection techniques in the task of the classification of scientific sentences into different rhetorical categories
Chapter 6 shows a method to classify sentences from scientific papers into rhetorical categories.
Chapter 7 discusses different text similarity techniques that are important factors for the task of citation linkage
Chapter 8 considers the task of citation linkage as a machine learning technique.
Chapter 9 considers the task of citation linkage as an information retrieval task.
Chapter 10 summarizes the thesis and presents future directions.

Chapter 2

Background and Related Work

2.1 Introduction

The key objective of this study is to investigate the requirements for citation linkage between scientific research articles. In this chapter we will review related studies in this direction. These studies can be classified into five key areas - a) Citation analysis in research papers, b) Annotation schemes for scientific articles, c) Sentence level information extraction from scientific articles, d) Text classification techniques, e) Text similarity detection techniques.

- **Citation analysis in research papers** studies the relationship between cited works as well as the context in which the citation occurs in a scientific article. It is also concerned with the examination of the text surrounding the citation in order to identify key common patterns in citation reference across the scientific literature.

- **Annotation schemes for scientific articles** are required to identify different types of sentences and classify them into appropriate categories suitable for information extraction. Many text applications rely on sentence categorization and efforts have been undertaken by researchers in their attempt to come up with comprehensive schemes that could enable users to easily navigate the increasingly large scientific literature.

- **Sentence level information extraction from scientific articles** identifies relevant patterns of language expressions as well as specific entity types from unstructured or semi-structured texts. Studies in this direction have focused on extracting specific information types from scientific papers, such as entity names and relations they take part in.

- **Text classification techniques** are needed for the tasks of information extraction and sentence categorization. Many comprehensive studies have reported some of the best techniques used in text classification and information extraction experiments.

- **Text similarity techniques** are required to perform the linkage between a citation sentence and candidate text units in the cited paper. Research in text similarity approaches have been the focus of many recent workshops and challenges.

In the following sections we present different research works tackling each of these requirements.

2.2 Citations in Scientific Publications

Scientific research tends to build upon previous works, and the author is required to cite relevant work that is used as support to the current research. It is thus important for researchers to properly and appropriately cite references in scientific research papers in order to acknowledge their sources and give credit where credit is due. A cited work tends to boost its author's credential in the research community. To find out the extend of the influence of authors and their works in a research field, researchers have not only focused on the number of time these works are cited, but also on the reasons of the citation as well as the analysis of the content, or the part of the work that is being referred to. In 1961, the Institute for Scientific Information (ISI) published the Science Citation Index (SCI), sparking the interest for citation studies [53]. The SCI was presented as an ordered list of cited articles, each of which was accompanied by a list of citing articles. It was well received internationally by the research community and it rapidly grew from 600 journals in 1964 [54] to 3,700 of the world's leading scientific and technical journals in 2012. It covers a large range of journals, including the world's most prestigious scientific and technical journals in almost all disciplines. Studies have been devised to analyze links between authors as well as impact of scholarly works and journals in research fields.

2.2.1 Citation Analysis

The relationship that exits between (part of) a citing document and a cited document has been the focus of many previous works. The study of such relationship is called citation analysis. Citation analysis uses citations in scientific and technical works to establish links between research publications as well as between authors of research articles. Citation analysis can be conducted quantitatively and qualitatively. The quantitative study of citation analysis implies that the scientists are concerned with the number of citations that occur in the paper and the qualitative study implies that scientists are also concerned with the content and information in the citation context. In information science, bibliometrics is a set of methods to quantitatively analyze scientific and technological literature. Bibliometrics uses statistical measures for the

qualitative evaluation of individuals, institutions, publications and even countries in science. Many tools are now available to calculate and present these measures. For instance, journal impact factor is a measure of the frequency with which the journal's average article is cited and citation counts and rates of individuals/institutions are used to gauge their scientific productivity. The Science Citation Index and Journal Citation Reports are notable examples of science management tools. However, many criticisms arise over the use of citation counts as a measure of quality of scientific contribution and many theoretical objections have been pointed out. For instance, Garfield [55] notes that, citation counts may be inflated by self-citation. Also, it is difficult to relate the citation rate of co-authored papers to individual authors, and articles in prestigious journals may be cited more than articles of equal quality in less prestigious journal, due to the journal's visibility. It is therefore often stated that citation analysis based on raw citation counts is not sufficient to determine the underlying reasons for the citation. Works may be cited in refutation or as negative examples. Therefore, not all citations are positive endorsements of the cited work. Such criticisms have prompted many citation studies.

2.2.2 Citation Context Analysis and Citer Motivation

In order to gain a better understanding of the citation process and the validity of citation analysis for quality assessment, Liu et al. [105] present a thorough review on citation analysis. Research in citation analysis may be categorized in a number of ways. For example, they label studies by their research objectives, giving five labels (which are not mutually exclusive): to enhance citation indexes, to describe the manifest functions of citations, to assess the quality of citations, to define concepts attributed to the cited work by the citing work and, lastly, to examine the underlying motives for citing.

Alternatively, citation studies can be categorized by methodology. In this approach, studies in citation analysis may be divided into two broad categories: whether the text of the citing document is the object of study or not. The first category is called citation context analysis [161], where "context" corresponds to the textual passage or statement that contains the citation. The second category is mainly related to the work on citer motivations. Works in this line examine the motives that authors have for citing; such motives are generally outside of the text. Brooks' investigation of citer motives [21] is cited as the first study of real authors and their motives for citing. Using surveys and interviews, this work identifies seven motivational parameters, including persuasiveness, reader alert and both positive and negative credit, and found persuasiveness to be the main motivating factor, i.e., the citing author is marshalling earlier work in order to persuade readers of the quality of his or her own claims. The scope of citation context analysis can be very broad. Small [161] subdivides citation context analyses

into those which use the citing text to abstractly classify citations and those which use it to identify the particular, concrete topics that are being attributed to the cited work. However, the two approaches are not always entirely distinct. The first may be unambiguously called citation classification and is principally concerned with the relationship between the citing and cited document. The second is concerned with the topical content of the cited document and is called citation content analysis and content analysis of citation contexts [168], [171].

2.2.3 Annotation Scheme for Citation Sentences in Scientific Research Articles

Citation classification schemes define a set of classes with which to encode the relationship between the citing and cited documents [152]. In this line of study, citation classification allows patterns in citations to be studied in finer detail, making it possible for citation analytic techniques to distinguish between types of citations. Moravcsik et al. [132] present one of the earliest classification schemes, intending to improve understanding of citations and, specifically, the extent to which citation counts can be used for various purposes in science policy. Based on the analysis of 30 research papers in theoretical high energy physics, the study proposed a classification scheme consisting of four categories [132]: *conceptual/operational*, *organic/perfunctory*, *evolutionary/juxtapositional* and *confirmational/negative*. This scheme was modified in later works in order to make the categories more appropriate to a non-scientific field and easier to code [168]. Most of the earlier classification schemes are subject to subjectivity as they seem to be only intended to be executed by their creators. They are usually tested on a small number of samples and it is difficult to generalize them to a larger corpus. Therefore, they cannot be applied automatically, and this reduces their practical use in an age when large bodies of literature are available in electronic form [152]. In more recent works, schemes have been developed to enable an automatic classification of citations. Nanba et al. [135] presented a simplified citation classification scheme involving three categories: *compare*, *based-on* and *other*. Rules are created manually based on cue words/phrases to be applied automatically by a system to support writing domain surveys. Recently, Teufel et al. [174] have presented a fine-grained sentence annotation scheme and employed machine learning techniques to achieve automatic classification of citation sentences based on that scheme. They attempted to categorize citations into their rhetorical function by pointing out the author's reason for citing a given paper. This classification scheme comprise 12 categories including: the acknowledgment of the use of the cited method, the contract between methods, or comparison between results, and so on. Teufel's categories such as CoCoGM (Contrast/Comparison in Goals or Methods), CoCoXY (Contrast between two cited methods), PUse (Author uses tools/algorithms/data/definitions)

and PModi (Author adapts or modifies tools/algorithms/data) are concerned with the author's use of methods or techniques that are described in the cited paper. These categories represent 1/3 of the 12 categories and 2/3 of the whole dataset when we remove the Neut (Neutral description of cited work) category. We can therefore infer that methods or techniques from previous research works are cited most often in research papers.

Other works on citation classification include the work by Garzone and Mercer, who presented an automated citation classifier, which involved a pragmatic grammar consisting of lexical and parsing rules that were developed based on cue words extracted from the citation and its location in the article [56]. Later, Mercer and DiMarco extended the work of Garzone and Mercer to propose the use of fine-grained cue phrases within citation sentences for classification purpose [121]. Also, Nakov et al. [134] proposed the use of the text of the sentences surrounding citations as tools for semantic interpretation of bioscience text. This work emphasized several different uses of citation sentences and showed that citation sentences are rich in domain specific concepts and terminology. Nakov et al. [134] proposed a methodology to automatically extract paraphrases of facts about a cited paper from multiple citations to it, with the eventual aim of using these sentences to automatically create summaries of the cited paper. This is in line with Small's [162] notion of cited works as concept symbols, whereby a work may come to be repeatedly and consistently cited to represent a specific idea or topic, using descriptions that converge on an almost fixed terminology for that topic, such that the cited work eventually becomes synonymous with the topic. Our work is partly based on the citation context analysis concept. While most of previous studies have focussed on citation sentences in the citing paper, the current work will be on both the citing and the cited papers.

2.3 Annotation Scheme for Sentences in Scientific Research Articles

A research article comprises a collection of sentences arranged in meaningful sequences alongside other elements such as tables, figures, diagrams. Scientific papers are presented in a cohesive formatting with comprehensive structures to enable the reader to understand the topic under investigation as well as background information and supporting materials used to carry out such investigation. Text in a running article is presented as groups of units of sentences, arranged in paragraphs, sections and sub-sections. Discourse analysis theories, such as Rhetorical Structural Theory (RST) [183], Discourse Representation Theory (DRT) [82], Segmented Discourse Representation Theory (SDRT)[95] and Discourse Lexicalised Tree Adjoining Grammar [50] have been proposed to analyze the characteristics of scientific text struc-

ture and the relationship that exists between its components. The different principles inherent to textual cohesion have been the primary focus of these theories. Existing text annotation schemes have relied on such theories for the classification of scientific article sentences into different rhetorical categories such as Introduction, Method, Result and Discussion, etc. Also, emphasis has been put on the distinction between citation and non-citation sentences in the annotation process. A number of text annotation schemes was created either for full text articles or abstracts.

2.3.1 Full Text Articles Annotation Schemes

Annotation schemes for full text articles have been the focus of many studies among which the "argumentation zoning" concept coined by Simone Teufel in her attempt to subdivide research articles into pre-defined zones in order to facilitate information extraction from scientific papers [171]. She proposed an annotation scheme of seven categories namely- *background*, which refers to sentences describing background knowledge; *aim*, which refers to sentences describing the research goal of the article; *textual*, which contains sentences referring to the textual content of the article; *own*, which contains sentences describing any aspect of the author's own work and not covered under *aim* or *textual*; *contrast*, including sentences that contrast the author's own work with other works or that point out weaknesses in other research; *basis*, containing sentences or statements that point to the use of other work as a starting point for the author's work or that explain how it gets support from the cited work; and *other* which refers to sentences that describe some aspects of other research in a neutral way [171]. A decision tree was designed to carry out the annotation. The main purpose of the argumentation zoning scheme was to identify the rhetorical status of a sentence with regard to the discourse flow presented in the paper.

Teufel's argumentation scheme was extended by Mizuta and Collier [127] for zone analysis in biology texts. The scheme comprises seven categories namely: *background*, *problem setting*, *outline*, *textual*, *own*, *difference* and *connection*. Three major modifications were made to the original argumentative zoning scheme. The category of sentences that represents the author's own work is divided into classes such as *methodology*, *results*, *insights* and *implication*. Categories are also created to cover the relations between data or findings, and sentences can also be classified into one or multiple classes.

Mizuta and Collier [128] provided further support to their zone analysis presented in [127] by proposing a theoretical and practical framework for qualitative analysis of zone identification in scientific publications relating to biology.

Langer et al. [94] presented a fine-grained annotation scheme to fit the requirements for

applications in the Semantic Web domain. The scheme comprises sixteen topic types including categories for automatic classification of text segments in scientific articles. These categories are presented in a hierarchical manner comprising eight classes at the higher level. These higher level classes are *background*, *problem*, *textual*, *framework*, *evidence*, *method*, *answers* and *resource*. Also sub-classes that represent finer features of sentences are defined at lower levels.

In order to identify passages of scientific facts, Wilbur et al. [182] proposed an annotation scheme of five categories for biomedical texts. These categories are *focus*, *polarity*, *certainly*, *evidence* and *trend*. Wilbur et al. noted the complexity of Mizuta and Collier's [128] annotation scheme and showed that such a complex system might be difficult to use.

Ibekwe-Sanjuan et al. [75] use rhetorical and lexical clues to classify sentences in articles from experimental sciences such as quantitative biology as well as information science and astronomy. They provide a framework to annotate sentences in a rhetorical scheme based on local grammars. This scheme comprises eight categories highlighting key processes of the scientific investigation and report. They are: *objective*, *new things*, *results*, *findings*, *hypotheses*, *conclusion*, *related work* and *future work*.

Shatkay et al. [159] used Wilbur et al.'s [182] annotation scheme for multi-dimensional classification of biomedical texts and they noted that such a scheme could easily be used in practice.

Liakata et al. [102] presented two complementary annotation schemes for scientific papers in Chemistry: the Core Scientific Concepts (CoreSC) annotation scheme based on the proposed CISP (Core Information about Scientific Papers) metadata scheme presented in [166] and the Argumentative Zoning-II scheme (AZ-II) [171].

The CISP scheme is validated through an online survey and is said to contain the necessary components for describing scientific investigation, namely: *Motivation*, *Goal*, *Object*, *Method*, *Experiment*, *Observation*, *Result* and *Conclusion*. The CoreSC scheme implements these concepts along with others such as *Hypothesis*, *Model* and *Background* as a three layered annotation scheme for sentences. It consists of a number of categories distributed into hierarchical layers. The first layer consists of 11 categories, which describe the main components of scientific investigation, the second layer is properties of those categories (e.g. Novelty, Advantage), and the third layer provides identifiers that link together instances of the same concept.

The CoreSC annotation scheme aims at retrieving the structure of the investigation presented in the paper in terms of generic high-level Core Scientific Concepts (CoreSC) whereas the AZ-II scheme [171] focuses on modeling the rhetorical and "argumentational" aspects of scientific writing and is concerned with rhetorical statements and the links between citing and cited papers.

Scheme	Number of categories	Details of categories
Teufel (1999)	7	Background, Aim, Textual, Own, Contrast, Basis, Others
Mizuta and Collier (2004a)	7	Background, Problem setting, Outline, Textual, Own, Difference, Connection
Langer et al. (2004)	8 (high level)	Background, Problem, Textual, Framework, Evidence, Method, Answers, Resource
Wilbur et al. (2006)	5	Focus, Polarity, Certainty, Evidence, Trend
Teufel (2006)	12	Neutral, PBas, PMet, PSup, PSim, PMod, PUse, CoCoX, CoCoR, CoCo-, CoCoG, Weak
Ibekwe-sanjuan et al. (2007)	8	Objective, New Things, Results, Findings, Hypothesis, Conclusion, Related Work, Future Work

Table 2.1: Summary of annotation schemes for full text of articles

Table 2.1 shows a summary of Annotation Schemes for Full Text Articles.

2.3.2 Annotation Schemes for Abstracts

There are studies that have focused on sentences in abstracts of articles. These studies include the following. A sentence categorization scheme is proposed in [117], by which abstract sentences are classified into four general high-level categories, namely, *introduction*, *method*, *result* and *conclusion*. Also, in [67], information extraction and sentence classification methods are used to summarize clinical trial design. The target areas for this scheme were three aspects of clinical trial, namely, *compared treatment*, *endpoint* and *patient population*. In [187], Yamamoto and Takagi classified sentences in abstracts according to their rhetorical status into six classes: *background*, *purpose*, *introduction*, *method*, *result* and *conclusion*. Sequential classification models were used by Chung in [29], to identify key sentences from abstracts of Randomized Clinical Trials (RCTs) texts. The sentences were classified into three categories: *intervention*, *participants* and *outcome measures*.

In [62], Guo et al. investigated the applicability of different annotation schemes, based on section names, argumentative zones and conceptual structure of documents to biomedical abstracts and observed that a majority of the categories defined in these schemes appeared in abstracts and that they could be identified efficiently using machine learning techniques.

Kim et al. [86] classified sentences in medical abstracts into six categories: *background*, *intervention*, *outcome*, *population*, *study design* and *other*, for supporting evidence based medicine.

2.4 Sentence-level Information Extraction

The task of information extraction is concerned about extracting specific language expressions from unstructured or semi-structured texts and present them in a more structured format. In the biomedical, bioinformatics and the biological domains, relevant "focused" information, such as gene and protein names, disorders, chemicals and species have been the focus of many studies. Many systems have been built to identify specific entity types. Such systems can be integrated in more complex biomedical information extraction systems to perform tasks such as relation extraction [194], text classification [83] or topic modeling [96]. However, it is difficult to rely on a single technique to build systems to recognize all the occurrences of names for a given entity type due to the variations of biomedical names [196].

2.5 Text Classification Techniques

Text can be modeled as quantitative data with attribute values derived from word occurrence, such as word frequency. Most of the quantitative data analysis techniques can then be used directly on text. But due to the particularities of text data such as the sparsity and the high dimensionality of word attributes, specific changes are applied often to text in order to use most of the existing classification techniques.

Decision Trees: Different text features may be used to perform a hierarchical division of the underlying data space. Decision trees can then used to classify data instances along this hierarchical model. The hierarchical division of the data space is designed in order to create class partitions which are more skewed in terms of their class distribution. For a given text instance, decision trees are used to determine the partition that it is most likely to belong to. Teufel [172] used a decision tree to classify sentences in scientific articles into the argumentative zoning rhetorical categories. Figure 2.1 shows how the Teufel decision tree is used to identify sentence rhetorical types.

Pattern (Rule)-based Classifiers: In rule-based classifiers we determine the word patterns which are most likely to be related to the different classes. A set of rules is constructed having in the left-hand side a predefined word pattern, and on the right-hand side the corresponding class label. These rules are used for the purposes of classification. Garzone [57] used rules to classify citations in biomedical articles.

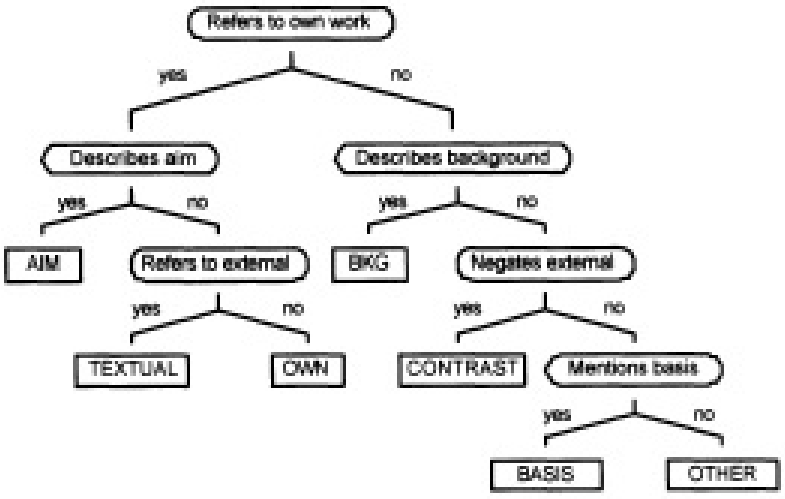

Figure 2.1: Teufel decision tree for argumentative zoning [172]

SVM Classifiers: SVM Classifiers attempt to partition the data space with the use of linear or non-linear delineations between the different classes. The key goal of such classifiers is to determine the optimal boundaries between the different classes and use them for classification purposes. Mullen et al. [133] used SVM (support vector machines) to classify sentences in full-text biomedical articles.

Bayesian (Generative) Classifiers: In Bayesian classifiers (also called generative classifiers), we attempt to build a probabilistic classifier by modeling the underlying word features in different classes. The idea is then to classify text based on the posterior probability of the documents belonging to the different classes on the basis of the word's presence in the documents. Teufel [171] used Naïve Bayes models for sentence categorization.

2.5.1 Sequential Classifiers for Sentence Classification

Sequential classifiers such as Hidden Markov Models (HMMs), Maximum Entropy Markov Models (MEMMs) and Conditional Random Fields (CRFs) have been used for sentence classification as well in [101]. The objective in this work was to show that sentence classification can be viewed as a sequential task because authors usually follow a sequential pattern when writing scientific articles. Swales [9] observed a sequential pattern in paragraphs of introduction sections in 50% of the articles he surveyed.

2.6 Semantic Similarity between Texts

The task of citation linkage will mostly consist of finding the semantic similarity between two units of text. The semantic similarity problem usually aims at deciding the degree of similarity between two text units A and B. Using quantitative measures, one can check to what extent text A has the same meaning as text B (paraphrase relation) or to what extent text A entails text B (entailment relation). Finding the best linkage candidates (i.e., the sentences having more or less the same "contentful" meaning as a given citation sentence), will amount to computing the degree of similarity between the citation sentence and each sentence in the citing paper. Generally, semantic similarity can be broadly construed as being assessed between any two texts of any size. Depending on the granularity of the texts, we can talk about the following fundamental text-to-text similarity problems: word-to-word similarity, phrase-to-phrase similarity, sentence-to-sentence similarity, paragraph-to-paragraph similarity, or document-to-document similarity. Mixed combinations are also possible such as assessing the similarity of a word to a sentence or a sentence to a paragraph. In the case of this study, we are mostly concerned with finding sentence-to-sentence similarity, but having in mind that the similarity could be at a more granular level.

Previous works on text similarity detection include paraphrase detection [42] and textual entailment [38]. On the one hand, paraphrasing methods recognize, generate, or extract (e.g., from corpora) paraphrases, phrases, sentences or longer units of text that convey the same, or "almost" the same information.

On the other hand, Textual entailment methods, recognize, generate, or extract pairs (T, H) of natural language expressions, such that a human who reads (and trusts) T would infer that H is most likely also true [38]. Both Paraphrase detection and Textual Entailment have been combined in the SemEval2012 Semantic Textual Similarity shared task [5], consisting of finding similarity between sentences in a text pair (T1 and T2) and returning a similarity score and an optional confidence score. The authors participating in the tasks used a combination of lexical and syntactical approaches as well as machine learning approaches. Lexical matching tends to be helpful in the case of simple sentences but it loses confidence in the case of complex and compound sentences. Usually, most systems based on machine learning tools performed better as they use a combination of explicit and implicit features to build a learning model.

2.6.1 Paraphrase Recognition Approaches

Paraphrase recognizers judge whether or not two given language expressions (or templates) constitute paraphrases. Methods for paraphrase recognition may operate at different levels of representation of the input expressions; some methods may treat the input expressions simply

as surface strings and other may operate on syntactic or semantic representations of the input expressions. Also there are methods that work with representations of the input expressions that combine information from different levels.

Logic-based Approaches to Recognition

The language expressions can be mapped to logical meaning representations. Bi-directional logical equivalence can be used to check for paraphrase relation between two text units. Practically, common sense knowledge obtained from resources such as WordNet [49], or extended WordNet [130] can be used to check the similarity between logical meaning representations of conceptual words. Pairs of formula (φ_{T1} ,φ_{T2}) can be generated of the pair of texts (T1, T2) and the task will consist of checking if ($\varphi_T \wedge$ B) $\models \varphi_H$, where B contains meaning postulates and common sense knowledge. For instance, since "analyze" is a hyponym (more specific sense) of "examine" in WordNet, an axiom like the following can be added to B [130, 19].

$$\forall x \forall y \; examine(x, y) \Longrightarrow analyze(x, y)$$

Recognition Approaches that Use Vector Space Models of Semantics

Each word of the input expressions (sentence pairs) is represented by a vector that shows how it strongly co-occurs with other words in corpora [104]. The vector computation can also take into account other information such as syntactic dependencies or ontological relatedness [138]. A vector-based meaning of the input expressions is determined by combining the vectors of single words using their sum, their products or some more elaborate approaches as proposed by [125], [47], [31]. Paraphrase can then be determined by computing the cosine similarity between the vectors representation of the two input expressions.

Recognition Approaches Based on Surface String Similarity

Many text similarity methods operate directly on the input surface strings, possibly after applying some pre-processing, such as part-of-speech (POS) tagging or named-entity recognition, but without computing more elaborate syntactic or semantic representations. For example, they may compute the string edit distance of the two input strings, the number of their common words, or combinations of several string similarity measures. Surface string similarity roughly speaking examines the percentage of word n-grams (sequences of consecutive words) of text1 that also occur in text2, and takes the geometric average of the percentages obtained for different values of n. Although such n-gram based measures have been criticised in machine

translation evaluation [24], for example because they are unaware of synonyms and longer paraphrases, they can be combined with other measures to build paraphrase recognizers [197].

Recognition Approaches Based on Syntactic Similarity

Text similarity recognisers can also work at the syntax level. For instance, dependency grammar parsers [119] [90] are commonly used in paraphrase recognition tasks. One approach may consist of counting the common edges of dependency trees of the input expressions [8]. Tree edit distance [157, 169, 193] can also be used to compute the similarity between two sentences.

Recognition via Similarity Measures Operating on Symbolic Meaning Representations

Paraphrases may also be recognized by computing similarity measures on graphs whose edges do not correspond to syntactic dependencies, but reflect semantic relations mentioned in the input expressions [64]; for example the relation between a *buyer* and the entity *bought*. Relations of this kind may be identified by applying semantic role labeling methods [111].

2.6.2 Paraphrase and Textual Entailment Recognition via Supervised Machine Learning

Machine learning approaches combine similarity measures computed at different levels, such as string levels, syntactic representations, dictionary, to build language expression similarity recognition models [126, 6]. Each pair of expressions that we wish to determine whether they are semantically related is converted into a feature vector. Each vector contains different similarity measures applied to the expression. In the case of the similarity detection between two text units, the feature vectors can contain cosine similarity scores, overlap measure, distance similarity scores, etc. Other features may be added as well. A supervised machine learning algorithm trains a classifier on manually annotated (as similar or dissimilar) vectors corresponding to training input pairs. Once trained, the classifier can classify unseen pairs as similar pairs or dissimilar pairs by examining their features. Figure 2.2 presents an illustration for the recognition approach using machine learning.

Table 2.2 presents the similarity methods and the resources that are used for the paraphrase recognition task [8].

2.6.3 Paraphrase Textual Similarity Evaluation Corpora

Datasets with both positive and negative instances of input pairs are necessary for the evaluation of textual similarity recognizers. Discriminative classifiers such as SVMs are often used to

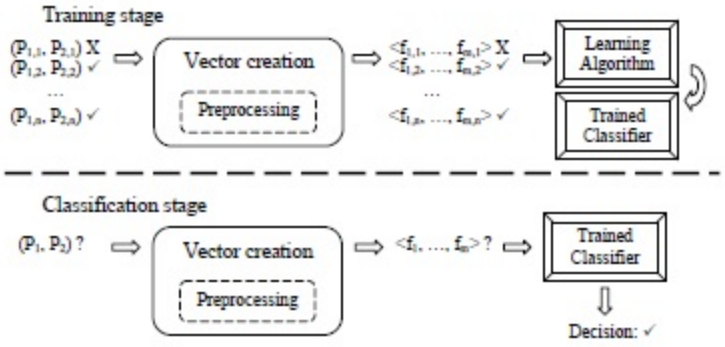

Figure 2.2: Paraphrase recognition via supervised machine learning [8]

Main ideas discussed	Main typically required resources
Logical-based inferencing	Parser producing logical meaning representations, inferencing engine, resources to extract meaning postulates and common sense knowledge from.
Vector space semantic models	Large monolingual corpus, possibly parser
Surface string similarity measures (high level)	Only preprocessing tools, e.g., POS tagger, named-entity recognizer, which are also required by most other methods
Syntactic similarity measures	Parser
Similarity measures operating on symbolic meaning representations	Lexical semantic resources, possibly parser and/or semantic role labeling to produce semantic representations.
Machine learning algorithms	Training/testing datasets, components/resources needed to compute features
Decoding (transformation sequences)	Synonyms, hypernyms-hyponyms, paraphrasing/TE rules

Table 2.2: Similarity methods and resources they typically require [8]

discriminate between the negative pairs and the positive ones.

In the case of paraphrase recognizer task, the most used benchmark dataset is the Microsoft Research Paraphrase corpus containing pairs of sentences extracted from news articles that refer to the same events [42, 43]. The pairs were first chosen using heuristics by which the word edit distance of the two sentences in each pair should be in a fixed interval in order to avoid nearly identical pairs and too many negative pairs. Also both sentences are required to be among the first three of articles among the same cluster (articles referring to the same

event). The reason is that initial sentences often summarize the events. At a second stage of the selection, an SVM-based paraphrase recognizer [20] -trained on separate manually classified pairs obtained in a similar manner, which was biased to overidentify paraphrases- was used to filter the candidate paraphrase pairs. Then, the rest of the sentence pairs were annotated by human judges to determine whether they are paraphrases or not. In the end about 67% of the initial 5,801 pairs were agreed upon by human judges to be paraphrases. The heuristics that were used in the selection process were biased towards paraphrase pairs that share many words in common as noted by Zhang and Patrick [195]. Therefore, one might argue that other types of paraphrases are excluded from the MSRP dataset.

The MSRP corpus has inspired other studies on semantic similarity datasets development.

The User Language Paraphrase Corpus (ULPC) [116], is a collection of pairs of texts obtained by asking students to paraphrase sentences extracted from biology textbooks. The responses were collected using the iSTART (an intelligent tutorial system) platform. The quality of the paraphrases between the target sentences and the student responses was assessed by human experts using a measure called "Paraphrase Quality bin" that rates the pairs along 10 dimensions of paraphrase characteristics. Each of these dimensions measures the paraphrase quality between the target-sentence and the student response on a binary scale. From a total of 1,998 pairs, 1,436 (71%) were classified by experts as being paraphrases. A quarter of the corpus is set aside as test data. The average words per sentence is 15.

The Question Paraphrase corpus [15] is extracted from the WikiAnswers web site and contains 1,000 questions along with their paraphrases (totaling 7,434 question paraphrases). The texts were taken from 100 randomly selected FAQ files in the Education category of the site. The 1,000 questions are called the target questions and the 7,434 question paraphrases are called the input questions. These are all true paraphrases (there are no explicit false paraphrase instances) which represent similar questions.

The SEMILAR corpus (formerly known as SIMILAR) [154] is richly annotated with information that can help easily assess similarity at many text levels such as word-to-word similarity, word-to-word similarity in context, sentence level paraphrase identification methods. It also includes word alignment algorithms. The SEMILAR corpus is made up of 700 pairs of sentences taken from the MSRP corpus. It contains 29,771 tokens (words and punctuation) of which 26,120 are true words and 17,601 content words. The number of content words is important because many word-to-word semantic similarity metrics available work on content words or certain types of content words, e.g. only between nouns or between verbs. The 700 pairs are fairly balanced with respect to the original MSRP judgments, 49% (344/700) of the pairs are TRUE paraphrases.

The Student Response Analysis corpus (SRA) [45] is built using students and experts

answers to tutorial questions from different science domains. The corpus contains 56 questions along with 3,000 student answers from the BEETLE corpus [44], as well as 197 assessment questions along with 10,000 answers from the ScientsBank corpus [137]. The question-answer pairs were annotated using a 5-scale annotation as opposed to the typical 2-scale annotation.

The Semantic Textual Similarity corpus (STS) [4] contains 2,250 pairs of headlines, machine translation evaluation sentences, and glosses (concept definitions). The data set is balanced and the selection process is based on string similarity techniques. The corpus comprises the STS CORE annotated on a 6-way schema ranging from 5=identical to 0=completely unrelated. This ratings were used to judge the similarity on a 0-5 scale (low to high similarity) by human judges recruited through Amazon Mechanical Turk.

2.7 Chapter Summary

We present a review of relevant work in citation research, text classification, information extraction as well as work in text similarity detection. We first provide an overview of the stream of studies focusing on the analysis of relationships between scholarly works. We also show some of the annotation schemes for citation and full text articles. Research in text similarity detection, with an emphasis on paraphrase detection was also presented. These different topics represent key components that are required for the task of citation linkage, which is the focus of this study.

Chapter 3

Test Collection and Datasets

3.1 Introduction

The evaluation of text-based applications requires the use of appropriate text collections and datasets that need to be annotated and validated by domain experts. While this is an ideal procedure, experiments have also proved that alternative methods that require minimum domain experts' input for corpus annotation might also be possible and reliable.

On the one hand, when time and resources permit, domain experts can annotate data instances by providing human judgment depending on whether the instance satisfies predefined characteristics or not. The resulting dataset or text collection forms a gold standard dataset or corpus that usually contains positive and negative instances determined by the domain experts. On the other hand, a silver standard corpus or dataset can be built relying on a combination of heuristics that don't necessary involve the full judgment of domain experts on each data instance. To build the datasets used in this thesis, we use human expert judgment as well as a combination of linguistic heuristics. We propose three datasets to be used for the following purposes:

- The first dataset is used to evaluate the task of information extraction presented in Chapter 4. This task aims at providing a framework for recognizing method mentions in biomedical texts.

- The second dataset is used to build a model for sentence classification in the biomedical domain. This dataset is used in Chapter 6: Sentence Classification for citation linkage. We assume that the target sentence will likely belong to the same category as the citation sentence. So, recognizing sentences of the method category, might help reduce the target sentence pool to this class.

- The third dataset is constructed to evaluate the task of citation linkage in scientific research papers, which is the ultimate goal of the thesis.

3.2 Corpus Building

The creation of a text collection and datasets is a necessary step in the evaluation of many text-based applications. These datasets are used by the experimenter to reproduce real-word scenarios in a laboratory setting by making use of a representative portion of real-world data. However, building the best representative dataset to emulate all the possible cases of the real world is a difficult, time-consuming and challenging task. To built experimental datasets, researchers have relied on two different methods.

In the domain of biomedical data mining, many Gold Standard Corpora (GSC) for diverse biomedical entity types have been constructed in the effort to provide relevant datasets for researchers in biomedical entity recognition and extraction tasks. Annotated corpora for gene and protein names have been built containing several thousands of annotated sentences to provide the framework for various challenges for the recognition of gene and protein names. BioCreative [163] and JNLPBA [85] challenge results show most of the advances in the biomedical Named Entity Recognizer (NER) research, especially with the machine learning based techniques. Also, corpora for the identification of disorders and species names have also been built, but the number of annotated documents is not enough to represent the whole range of names in these domains. Often, due to the difficulties to build corpora to reflect all the spectrum of entities in a given domain, efforts have been focussed on sub-entity type corpora that contain annotations for a specific category of mentions. For instance, the SCAI IUPAC corpus [25] contains only annotations of chemicals that follow the IUPAC nomenclature [141]. In many of these cases, only small sets of documents have been annotated, due to the complexity of building Gold Standard Corpora. In order to overcome this deficiency, efforts have been directed towards building large-scale datasets using automatic or semi-automatic annotation methods. Recently, the CALBC (Collaborative Annotation of a Large Biomedical Corpus) [150] has been proposed to provide a large-scale biomedical Silver Standard Corpus (SSC) automatically annotated through the harmonization of several NER systems. The resulting corpus comprises one million abstracts annotated with several biological semantic groups, such as diseases, species, chemicals and genes/proteins.

In the present thesis, we use a number of text datasets that fall into either the category of Silver Standard dataset or the category of Gold Standard dataset.

Our first text collection used in the task of information extraction reported in Chapter 4 can be characterized as a combination of Silver Standard dataset and Gold Standard dataset,

because it is created using heuristics to select sentences that have a high probability to belong to the method sentence category and then manually curated for method mention detection. The second text collection used in the sentence classification task reported in Chapter 6 is also a Silver Standard Corpus, whereas the Linkage corpus used in Chapter 8 and Chapter 9 is a Gold Standard Corpus fully annotated by domain experts.

We describe each of these datasets in the following sections.

3.3 Method Mention Extraction Dataset

One of the intermediate goals of this study is to extract the methods and techniques used in biomedical research papers in order to build a lexical resource comprising the name, the variants and definition of such methods and techniques as they are mentioned and used in research articles. The first task consists of the gathering of a comprehensive corpus that is large enough to contain an important number of method mentions and information about how they are used in the papers. One way is to select some research papers and scan through each of them to extract candidate "method sentences" manually. Another way is to use filters to automatically select in a large repository of papers, "method sentences" that we believe are about the methods or techniques used in the paper as well as the context of such sentences. We define the context of a "method sentence" to be a window of sentences coming after or before them. The first option has proved in the past to be more precise, but very difficult to implement. In fact, a manual information extraction task can be tedious and time-consuming and requires many specialized skills and domain experts. It is therefore recommended to use automatic extraction techniques whenever it is possible. Since our purpose is to find as many relevant sentences as possible, it is obvious that we cannot rely on existing corpora to achieve our goal. Similar corpora have been built for the task of sentence classification from biomedical articles into the IMRaD (Introduction, Methods, Results, and, Discussion) categories. In their attempt to classify biomedical research papers into these categories, Agarwal and Yu [1] used a corpus of 1131 sentences. Of these sentences, 389 were labelled Introduction, 363 were labelled Methods, 273 were labelled Results and 106 were labelled Discussion. Even though the corpus contained "method" sentences, very few contain the mention of the techniques used in the paper. Also, the context of the sentence was not retrieved. In a similar classification task, Liakata et al. [101] used a corpus of 265 articles from biochemistry and chemistry annotated at the sentence level by experts in the domains. Even though the corpus contains 8404 method sentences, many sentences belong to the same paper and are about the same methods or techniques. For instance 10 consecutive sentences can be annotated to belong to the method category. Those sentences usually refer to the same method or technology and some of them may not even contain any mention of a

method. Besides, few of the sentences contain a definition of the techniques used in the papers. Based on the study of the corpora mentioned above, we deemed it necessary to build a different corpus that contains as many method sentences as possible and a definition or a usage context of the method mention. We relied on some linguistic concepts such as anaphoric relations to produce our fine-grained method corpus. In fact, most demonstrative noun phrases are anaphoric; therefore a sentence that begins with the demonstrative noun phase "This method" is anaphoric and its antecedents are likely to be found in previous sentences. Torii and Vijay-Shankar [176] reported their work on anaphora resolution of demonstrative noun phrases in Medline abstracts, and found that nearly all antecedents of such demonstrative phrases can be found within two sentences. On the other hand, Hunston [74] reported that interpreting recurring phrases in a large corpus enables us to capture the consistency in meaning as well as the role of specific words in such phrases. So, the recurring semantic sequence "this method" in the PubMed corpus can help us to capture valuable information in the context of their usage. To build our corpus we therefore search for sentences starting with "This method" in the PubMed article repository as well as the sentences immediately preceding them. Then we collected the pairs of sentences. Sentence 1 in Example 1 contains the mention of the method and Sentence 2 contains its usage and definition context.

Example 1

Sentence 1 *: The Ortholuge method reported here appears to significantly improve the specificity (precision) of high-throughput ortholog prediction for both bacterial and eukaryotic species .*

Sentence 2 *: This method , and its associated software , will aid those performing various comparative genomics-based analyses , such as the prediction of conserved regulatory elements upstream of orthologous genes .*

We can see that Sentence 1 is talking about the method and Sentence 2 is a reference to the method mention and how it is used. Combining both sentences we therefore have sufficient information to extract the "method" mention, its usage and its benefits.

We can then derive such information to fill lexical components such as:

- Method Mention: ***Ortholuge method***
- Usage/Role: ***comparative genomics-based analyses/ prediction of conserved regulatory elements upstream of orthologous genes.***

Example2

Sentence 3 *: An alternative method for predicting protein function is the Phylogenetic profile method, also known as the Co-Conservation method, which rests on the premise that*

functionally related proteins are gained or lost together over the course of evolution [4].

Sentence 4 *: This method predicts functional interactions between pairs of proteins in a target organism by determining whether both proteins are consistently present or absent across a set of reference genomes.*

In Example 2, it is possible to extract the following information:

- Method: ***Phylogenetic profile method***
- Variant/also known as: ***Co-Conservation method***
- Usage: ***for predicting protein function / predicts functional interactions between pairs of proteins***
- How: ***by determining whether both proteins are consistently present or absent across a set of reference genomes.***

The information that we can derive from this pair is sufficient enough to create lexical resources that can be used in many natural language processing tasks.

Using the retrieval technique mentioned above, we have been able to retrieve about 6500 such pairs of sentences from 189 different journals and 2000 papers.

3.3.1 Semi-gold Standard Datasets for Method Mention Extraction

We fine-tune the corpus by creating two sets of semi-gold standard datasets with sentences taken only from BioMed Central journals. The first dataset comprises 918 pairs of sentences containing the first category of method mention, i.e., terminology units ending with a method keyword. The second dataset comprises 122 pairs of sentences of method mentions that don't contain a method keyword. In each dataset, we assumed that the method mention is in the first sentence and the other information about its usage is in the second sentence. We manually verify that those sentences meet the requirement to belong to the method sentence category by making sure that they contain at least a method mention. Table 3.1 shows the number of sentences per category.

3.4 Sentence Classification Corpus

The second goal of this study is to classify sentences from scientific research papers to match some of the IMRaD rhetorical structure classes such as Method, Result, Discussion/Conclusion, with machine learning using a self-annotating corpus.

Category (keywords)	Number of sentences	Proportion
Method	439	42%
Analysis	200	19%
Model	63	6%
Algorithm	73	7%
Approach	145	14%
Other (Machine learning corpus)	122	12%
Total	1040	100%

Table 3.1: Corpus statistics (combining both datasets).

The first task consists of the curation of a corpus that contains sentences representative of the defined categorization scheme. We have chosen to build this corpus by extracting sentences from a large repository of full-text scientific research papers, the PubMed repository. As with the previous datasets in Section 3.3, our assumption is that a sentence that appears in the co-referential context of the co-referencing phrase "This method", will likely talk about a methodology used in a research experiment and reported in the paper. Similarly, a sentence that starts with the expression "This result" is likely to refer to an experimental result context, etc.

We define the co-referential context of these phrases to be a small number of sentences preceding the sentences in which these "This" references occur. To collect sentences that belong to the "Result" category, our target candidates will be those sentences that come immediately before sentences starting with "This result…", as shown in Example 3. Similarly, to collect sentences that belong to the "Conclusion" category, our target candidates will be those sentences that come immediately before sentences starting with "This conclusion…".

Example 3

1. *We have developed a DNA microarray-based method for measuring transcript length on a genomic scale.*
   ***This method**, called the Virtual Northern, is a complementary approach to cDNA sequencing.*

2. *Interestingly, Drice the downstream caspase activated just as individualization begins (downstream caspases are typically activated by upstream caspases such as Dronc and Dredd) was not affected by inhibition of Dronc and Dredd.*
   ***This result** , along with the fact that Dronc and Drice were activated at different times and places, suggests that some other mechanism activates Drice.*

3. *We obtained a long-range PCR product from the latter interval, that appeared to encom-*

Category	Number of sentences	Proportion
Method (Met)	3163	31.9%
Result (Res)	6288	62.69%
Conclusion (Con)	534	5.39%
Total	9985	100%

Table 3.2: Corpus statistics

pass the breakpoint on chromosome 2 (Figure 1D, E).
***This conclusion** , however , was regarded with caution , since the region is rich in copy number variants (UCSC, Structural Variation track) providing a possible alternative explanation for the FISH results.*

Table 3.2 shows the number of sentences per category. We present the evaluation of this corpus in Chapter 6.

3.5 Building a Citation Linkage Corpus

As the ultimate aim of the study is to provide a framework for the identification of pairwise similarity between citation sentences and the target sentences they refer to in the cited paper, we deem it necessary to build a corpus that is suitable, as much as possible, for such a task. The different steps leading to the construction of this dataset can be divided into four main parts, namely:

- The definition of the scope of the targeted citation types.
- The definition of the annotation guidelines.
- The annotation process.
- The annotator's feedback.

As the content of citation references is often expressed in full sentences containing specific terminology that may represent a set of methods, tools or techniques used in scientific experiments, we assume that targeting citation sentences containing method mentions and the context of their usage might help researchers to understand how the methods have been used in the cited paper. So the scope of the corpus is limited to citation sentences of this category.

We have given an annotator some annotation guidelines (see Appendix C) that have been generated with the following intended objectives underlying the instructions:

- To what extent can the person who reads a citation sentence taken in isolation be able to determine the candidate sentences that have been cited in a reference paper.

- For a given citation sentence from a citing paper *A*, there are one or more sentences from the cited paper *B* that are similar in terms of content (our assumption is that the determination of similarity of content by a domain expert could be achieved by word similarity including the use of synonyms, the use of domain knowledge possessed by the domain expert, the use of the surrounding textual context, etc.) and we would like a domain expert to be able to identify such candidate sentences.

- Candidate sentences are chosen from the full article *B*. They are presented chronologically as they appear in the article, thus providing the textual context that is mentioned previously.

- For each sentence in the cited paper, a score will be given. This score will indicate the confidence that the annotator had in making his/her choice of candidate sentences. For those sentences not chosen as candidate sentences, a score of 0 is given indicating that the annotator is confident that there is no similarity in content with the citing sentence. For those sentences chosen as candidates, a score is given ranging from 1 (low confidence that similarity in content exists) to 5 (the annotator is confident that there is strong similarity between the candidate and citing sentences).

We choose the papers to annotate in such a way that they belong to a citation network that shows the links between cited and citing papers. For this purpose, we use the BioMed Central's research articles corpus which is an open access corpus ideally suited for data mining research. The richness of BioMed Central's XML format also makes the content especially suitable for information extraction and textual analysis.

Figure 3.1 shows a graph of a citation network among papers drawn from the BioMed Central corpus. It is a directed graph in which the source nodes represent citing papers and target nodes represent cited papers. For instance, node 10-381.xml represents a paper that cites 9 other papers in the bioinformatics journal of the biomedical corpus. Table 3.3 presents an example of an annotation produced by a domain expert.

We have limited the current research to the biomedical domain and our final corpus is curated with papers from this domain. The final corpus is comprised of 26 papers among which 4 have only similarity ranging from 1–2. After an analysis of these papers we then conclude that the candidate linkage sentences cannot be included into the final experimental corpus. The annotators are also instructed to give the reason why they chose the candidate sentences.

Citation Sentence
We have been able to amplify 200 bp fragments of DNA obtained from Bouin's fixed and paraffin wax embedded tissues only after a specific restoration method to produce longer reconstructed DNA fragments.

Candidate Sentences	Rating
To obtain longer stretches of DNA, a pre-PCR restoration treatment was required , by filling single strand breaks, followed by a vigorous denaturation step.	4
The development of this simple treatment allowed the analysis of longer fragments of DNA obtained from archival postmortem paraffin wax embedded tissues.	3
A partial restoration and reconstruction of DNA length in these cases is possible.	1
Here , we show that it is possible to analyze human postmortem paraffin wax embedded tissues amplifying a 287 bp sequence of apolipoprotein E (ApoE) and 291 bp of the prealbumin gene (TTR).	3
DNA was extracted from 6 m sections of paraffin wax embedded tissues.	1
The final sample of DNA was obtained by precipitation with ethanol using glycogen as the carrier.	1
DNA samples were incubated for one hour at 55C in 100 l of solution containing 10mM Tris/HCl (pH 8.3), 1.5 mM MgCl2 , 2 % Triton X-100 , and 200 M of each dNTP.	2
After this incubation, 1 U Taq DNA polymerase (Amersham) was added and DNA polymerisation was performed at 72C for 20 minutes.	2
The polymerase reaction restores the nicks after DNA rehybridisation, using the other strand as the template.	3
We have developed a method for amplifying longer DNA sequences, ranging up to 300 bases , from postmortem formalin fixed and paraffin wax embedded tissues, with no modification to the usual DNA extraction procedures.	5
Our restoration method is based on the fact that DNA degradation results from random single strand breaks and polymerase reaction restores the nicks, using the other DNA strand as a template.	2
Our restoration method is based on the fact that DNA degradation results from random single strand breaks and PCR restores the nicks, using the other DNA strand as a template.	2
The method proposed can be used to obtain longer amplification fragments of around 300 bp of DNA from normally extracted postmortem paraffin wax embedded tissues.	4

Table 3.3: Example of an annotation

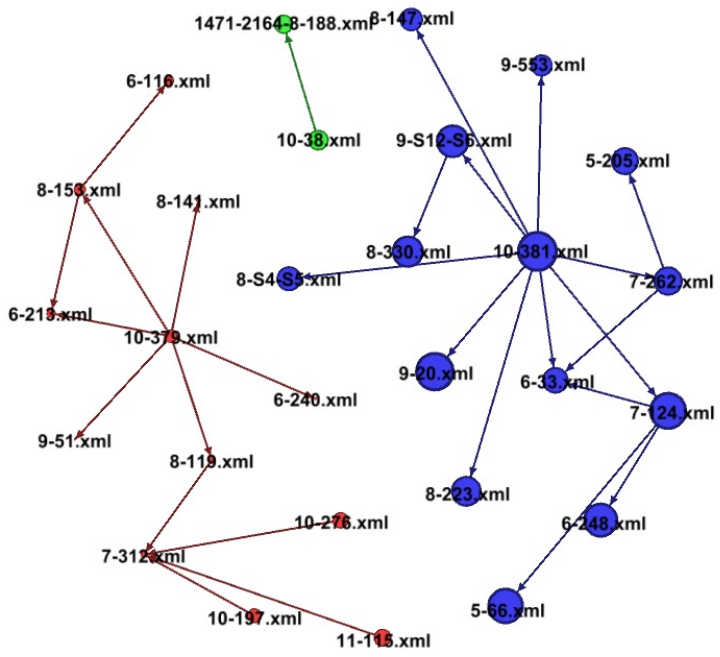

Figure 3.1: Graph showing citation link between papers

3.5.1 Annotators' Feedback

The annotation task for the citation linkage corpus requires the annotator to read the citation sentence (with no context) and then to choose citation sentence candidates from all of the sentences in the referenced article. The annotator then gives a numeric confidence score (1–5; 1 is the lowest confidence, 5 is the highest) indicating how certain the annotator is that the sentence is a citation candidate. The annotator also gives comments about his/her choices.

During the process of the selection of candidate sentences by the annotators, different factors have been taken into account. Many selections are done with the assumption that the underlying citation makes reference to many sentences that point to specific domain knowledge that are not necessarily visible in the citation sentences. Such assumptions lead to choices that require deep understanding of the subject matter. We present below some of the feedback collected from annotators.

Remark 1: Selection based on subtle similarity

"I selected sentence 49 because it mentions the use of SNP assays for identity conformation. The candidate sentence also discusses SNP assays being used for identity conformation.

I gave it a weak rating because the sentence is not as much about the use of SNP assays for identity conformation as it is about techniques used in addition to SNP assays for identity confirmation."

From this remark, we notice that the annotator made her selection based on the fact that even though the citation sentence and the candidate sentence make mention of the same technique, the context of usage is different. The citation sentence is really about using the target technique referred to as "SNP assays" , whereas the candidate sentence is more about another technique used "in addition" to the "SNP assays" technique.

Remark 2: Selection based on Inference

"This paper was more centered around the protocol, so I think that is why I have so many more sentences. I had to make inferences about several. For sentence 35, I had to infer that the buffer described was one of the buffers referred to as "such buffers" in the sentence on the left. Sentence 39 was a more detailed description of the precipitation/concentration protocol described in the sentence on the left. 40, 44, and 45 were tricky because I had to infer two things. First, I had to infer from the text leading to it that the nucleic acids were the ones that had been extracted and concentrated. As well, the left sentence says that some nucleic acids are extracted organically and then used for PCR and cDNA work. It does not explicitly say that the ones that have to be further concentrated are used for the same thing, but it is implied. (A lot of times in biochemistry, you don't get as much DNA, RNA, etc as you want using an initial technique so you have to further concentrate it to do things that you would have done had you gotten a high yield with the initial technique. That's what is happening here, I think). So with that in mind, I inferred that this concentrated nucleic acid (described in the sentence on the right) being used for the techniques described in the sentence on the left fit the description."

From this remark, we can understand that many inferences were needed for the annotator to choose candidate sentences. We can notice that, while selecting these sentences, the domain expert "thinks" that the description of the target technique and the experiments in which it is used, rely on some aforementioned " information" that is not present at a surface level in the cited paper.

Remark 3: Selection based on Background knowledge

"I made bigger inferences for 54, 60, and 98 and 67, 98, 100, 108, 119, 130, 132, 134, 135, and 153 required some inference and knowledge of biochemistry terminology and techniques. As you can see, I selected a lot of sentences. This is because the sentence on the left is talking about how TAE1 is involved in translation or biosynthesis. The paper looked at many ways

Number of Papers	Total Number of Sentences	Number of Sentences per Category
26	4318	0: 4018 5: 33 4: 48 3: 42 2: 42 1: 44

Table 3.4: Linkage corpus statistics

that TAE1 is involved in these processes in order to come to this conclusion and they gave a detailed description of their results, so the reader could infer what these results meant, and then explained what the results meant as well. This is a bit unusual as most papers let the reader look at the figures to see the results, rather than describing them in the degree of detail given here. They did this several times, so I wound up selecting a lot more sentences than usual."

Here the annotator needed to make "bigger inferences" to be able to link the citation sentence to the candidate linkage sentences. Specific domain knowledge terminology and technique are required for the annotation to be possible. Not only should the reader have an advanced knowledge of the domain, but they should also be able to grasp relevant information from supporting figures presented in the target paper.

3.5.2 Corpus Statistics

Based on these remarks, we can rightly say that the task of a citation linkage corpus annotation is very challenging and time consuming. Making a computer reproduce such a task by building automatic citation linkage systems will imply incorporating in the learning model most of the features that can be derived from the annotators' remarks, such as ***inference***, ***background information***, etc. However, it is virtually impossible to incorporate inference features, efficient domain specific knowledge and background information in the learning model, due to the scarcity of such resources. To make the problem more tractable we will rely on state-of-the-art text similarity techniques and resource to infer the linkage between the cited and citing sentences. This corpus is used in Chapters 8 and 9.

3.6 Chapter Summary

We have presented in this chapter different corpora and datasets used in the experiments reported in the thesis. Two different approaches are engaged to build these datasets and text

collection. We used a semi-automatic approach based on linguistic heuristics to construct the datasets used in the information extraction and the sentence classification phases. The corpus used in the linkage experiments is annotated by domain experts. These datasets and corpora are evaluated in Chapter 4 and Chapter 5, and used in Chapter 8 and Chapter 9.

Chapter 4

Information Extraction from Scientific Publications

4.1 Information Extraction Techniques

While scientific texts usually follow well defined argumentation structures, most of the information they contain is often hidden under more complex language expressions that require further processing in order to extract from them some specific information at a granular level. In fact, a scientific publication is often an eventful discourse aiming at presenting the way different agents interact to achieve a desired outcome. For instance, in the process of proving how two proteins interact, the argumentation may be about describing such proteins, the event that binds them, and the conditions in which such events occur. A reader might later be concerned with extracting specific proteins mentioned in a research publication as target entities and the event that binds them as a type of relation they engage in. The task of information extraction is then concerned about extracting these kinds of specific language expressions from unstructured or semi-structured texts and presenting them in a more structured format. In the biomedical, bioinformatics and the biological domains, relevant "focused" information, such as gene and protein names, disorders, chemicals and species have been the focus of many studies. Systems have been built to identify specific entity types. Such systems can be integrated in more complex biomedical information extraction systems to perform tasks such as relation extraction [194], text classification [83] or topic modeling [96]. Many techniques usually come into play in the process of building systems to recognize all the occurrences of names for a given entity type due to the various variations of some entity names in the biomedical domain [196]. Such variations may be of diverse forms:

- Many entity names are descriptive (e.g. **"*normal thymic epithelial cells*"**)

- Two or more entities can share one head name (e.g.***"91 and 84 kDa proteins" refers to "91 kDa protein" and "84 kDa protein"***)
- One entity name can have several spellings (e.g. ***"N-acetylcysteine", "N-acetylcysteine", and "NAcetylCysteine"***)
- Ambiguous abbreviations are often used (e.g. ***"TCF" may refer to "T cell factor" or to "Tissue Culture Fluid"***).

Therefore, NER systems developed for the biomedical domain have relied on different approaches and techniques that can broadly be divided into three categories, namely: rule-based, dictionary matching and Machine Learning (ML).

- **Rule-based approaches** are used for names with strongly defined orthographic or morphological structures.
- **Dictionary based techniques** are used for closely defined names and vocabulary entity types.
- **Machine Learning approaches** are used for names with strong variation of vocabulary.

In this chapter we will first present some of the works in named entity recognition. Then, we will show how the task of information extraction can be expanded to other entities such as method and technique mentions.

4.1.1 Rule-based Methods

A set of rules are either pre-defined by domain experts or automatically discovered. Rules are generally expression patterns that match specific entities. When a text is run against the rules and a match is found, an entity type is extracted in the form of a sequence of tokens. A representative set of features can be computed to match each token occurrence in the sequence. For instance in the recognition of protein nouns such as "***91 and 84 kDa proteins***", the target features may include: *number* (91), *conjunction* (and), specific *word shape* (kDa), and a *head-word* (protein), occurring in a specific order.

4.1.2 Dictionary-based Techniques

Dictionary based information extraction methods compare the text against vocabularies stored in databases or lexicons such as UMLS or Wordnet. When specific words or sequence of tokens in the text match terminology present in a specialized database, they are said to belong to a specific semantic type defined in the database.

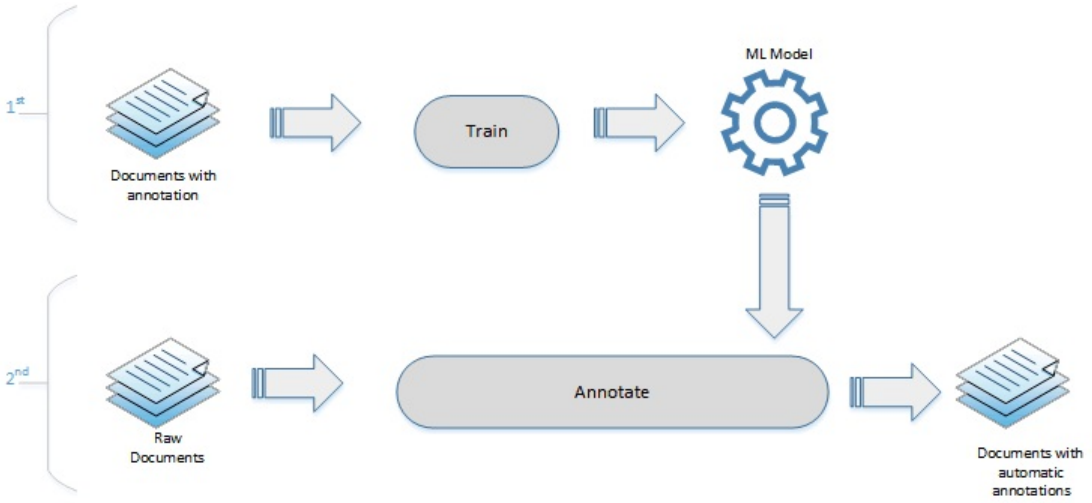

Figure 4.1: Development process of ML-based solutions

4.1.3 Machine Learning Methods

Most of the advances in NER systems are achieved with machine learning techniques. The main requirements for the information extraction applications using machine learning techniques are:

- Annotated corpus
- Feature selection
- Learning methods
- Evaluation techniques

Figure 4.2 shows a synopsis of an information extraction pipeline using machine learning.

Corpora

Annotated datasets need to be created in order to evaluate information extraction methods. When the information to be extracted is domain specific, the dataset is often taken from the target domain. For instance when genes and proteins are to be extracted, corpora such as BioCreative [163] and JNLPBA [85] are used.

Pre-processing

Pre-processing involves the following steps:

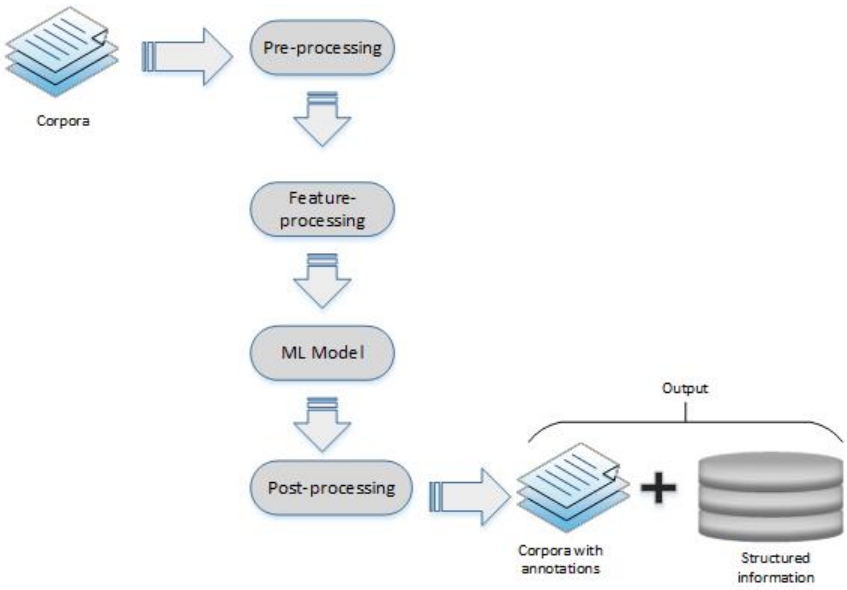

Figure 4.2: Pipeline for the ML process

- **Sentence splitting**: Text documents need to be broken down into their respective sentences. Various tools for sentence segmentation [109] have been developed to perform sentence boundary detection for biomedical documents [175].

- **Tokenization**: Each sentence needs to be broken down into its constituent meaningful units called tokens. Tools for tokenization include: GENIA TAGGER [177], JTBD [175], SPECIALIST NLP [17].

- **Annotation Encoding**: An annotation scheme is needed to represent the internal structure of each annotated entity name. The IO encoding tags each token as either being in ("I") or outside ("O"). However this simplest representation is only suitable for representation two entities next to each other. The BIO encoding has been proposed to resolve the boundary problem. In this scheme, the "in" tag is subdivided into tag "B", representing the first token or beginning of the entity name, and tag "I" for the remaining tokens.The BMEWO encoding extends the BIO encoding by distinguishing the end of an entity (tag "E") tokens from the middle entity tokens (tag "M"), and adding a new tag ("W") for entities with only one token [18].

Feature Processing

The selection or computation of the most representative features that reflect the characteristics of the entities and the context of their occurrence is essential for the recognition task. Features

should be carefully chosen to encode the linguistic characteristics and the naming convention of the target entity names. Such features include:

- **Linguistic Features:** The token itself is the primary feature as it is a surface representation of the entity type. Stemming or lemmatization can also be applied to the word to include its morphological variants so that they can be analyzed as a single item. Each token may also be associated with a grammatical category such as its Part-of-Speech (POS). Chunking can also be used to divide the text into syntactic constituents such as Noun Phrase and Verb Phrase. Features incorporating the relationship between various tokens in the text based on dependency parsing can also be added to the feature pool.

- **Orthographic Features:** They are used to capture the knowledge about how words are formed in the domain. For instance certain protein names contain uppercase and lowercase letters as well as numbers and other symbols. Therefore features that capture the presence of such indicators can be valuable attributes for the recognition task.

- **Morphological features:** They reflect common structures or the presence of subsequences of characters among several entity names. They can be used to establish the similarity between various tokens. Generally, three type of morphological features are used.

  - Suffixes and prefixes: They can be used to distinguish certain entity names. For instance, suffixes like “ase”, “ome” and “gen” frequently occur in gene and protein names.
  - Char n-grams: They are sub-sequences of n characters from a given token. This feature extends suffixes and prefixes by considering sub-sequences of characters in the middle of tokens.
  - Word shape: Such patterns can reflect how letters, digits and symbols are organized in the token.

- **Context features:** The environment in which a token occurs - such as the words surrounding it - can be used to reflect the local context of its usage. A context feature is often created by establishing a higher-level relation between a token and the preceding or/and succeeding tokens through a specific window. New features can also be created by grouping features of the surrounding tokens.

- **Lexicons:** NER systems can be further optimized by adding domain knowledge to the set of features. In the biomedical domain, specific domain terms and entity names can be matched to specific semantic class tags that can be used as feature.

4.1.4 Information Extraction Machine Learning Models

In the biomedical domain, the task of biomedical NER can be treated as a sequence labeling problem. Many natural language processing tasks have been successfully modeled with sequence labeling algorithms. Part-of-speech tagging, chunking and other have used sequence labeling techniques. It is formulated as follow:
Given a sequence of observations x = (x_1, x_2,..., x_n) , we would like to assign a label y_i to each observation x_i based on the assumption that each label y_i depends not only on its corresponding observation x_i, but also possibly on other observations and other labels in the sequence. The close neighborhood of the current position i is often considered to provide enough information to model this dependency.
Each word in a sentence is treated as an observation, and the class labels should clearly indicate both the boundaries and the types of named entities within the sequence. The BIO notation is usually used and for each entity type T, two labels - B-T and I-T - are created. A token labeled with B-T is the beginning of a named entity of type T while a token labeled with I-T is inside (but not the beginning of) a named entity of type T. In addition, there is a label O for tokens outside of any named entity.

Hidden Markov Model

In probabilistic reasoning, the best label sequence $y = (y_1, y_2, \ldots, y_n)$ for a corresponding observation $x = (x_1, x_2, \ldots, x_n)$ is the one that minimizes the conditional probability $p(y|x)$ or equivalently, the one that maximizes the joint probability $p(x, y)$. To model the joint probability, the Markov process assumes that the generation of a label or an observation is dependent only on one or a few previous labels and/or observations. By treating y as hidden states, we get a hidden Markov Model [46].

Maximum Entropy Markov Model

One problem with the hidden Markov models described above, is that being generative they model the probability $p(x|y)$, and tend to give a higher prediction error rate compared to discriminative models that directly model $p(y|x)$. When training data is sufficient, discriminative models are preferable over generative models [179]. For the task of named entity recognition, researchers have been using more generative models than discriminative models. A commonly used discriminative model for named entity recognition is the maximum entropy model [14] coupled with a Markovian assumption (MEMM). Existing work using such a model includes [13, 36].

Conditional Random Fields

Conditional random fields (CRFs) are state-of-the-art discriminative models for sequence labeling [92]. The main difference between CRFs and MEMMs is that in CRFs the label of the current observation can depend not only on previous labels but also on future labels. Also, CRFs are undirected graphical models while both HMMs and MEMMs are directed graphical models. CRFs are said to resolve the "label bias" problem in the sense that they use a single exponential function to model the joint probability of the entire label sequence, whereas MEMMs use a per-state exponential model.

Evaluation measures

The evaluation of the recognition models is performed by comparing automatic annotations with the one provided by domain expert annotators. This will enable the experimenter to understand the behavior of the system by measuring the accuracy of the machine generated annotations. For this purpose, each automatic annotation must be classified as being a:

- True Positive (TP): the system provides an annotation that exists in the annotated corpus.
- True Negative (TN): the non existence of an annotation is correct according to the annotated corpus.
- False Positive (FP): the system provides an annotation that does not exist in the annotated corpus.
- False Negative (FN): the system does not provide an annotation that is present in the annotated corpus.

Three important measures are used evaluate the system's performance: **precision, recall and F-measure**. These measures assume values between 0 (worst) and 1 (best). Precision measures the ability of a system to present only relevant names, and it is formulated as:

$$Precision = \frac{relevant\ names\ recognized}{total\ names\ recognized} = \frac{TP}{TP + FP} \tag{4.1}$$

On the other hand, recall measures the ability of a system to present all relevant names, and is formulated as:

$$Recall = \frac{relevant\ names\ recognized}{relevant\ names\ in\ corpus} = \frac{TP}{TP + FN} \tag{4.2}$$

F-measure is the harmonic mean of precision and recall. The balanced F-measure is most commonly used, and is formulated as:

$$F\text{-}measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.3}$$

The system's accuracy can also be computed to access the proportion of both true positives and true negatives among the total number of instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.4}$$

The system's error rate is the proportion of cases where the prediction is wrong.

$$Error\ Rate = \frac{FP + FN}{TP + TN + FP + FN} \tag{4.5}$$

4.2 Method Mention Extraction

The methods and techniques used during scientific experiments and reported in scientific papers are often expressed in different forms. They can be in the form of a semantic sequence (see [74] containing the word *method ("rank-based normalization method" in Sentence 1)*, the word *technique ("Non-negative matrix factorization technique" in Sentence 2)*, the word *analysis ("discriminant analysis" in Sentence 3)*, and so on.

Sentence 1: *In order to compare U133A and U133 Plus 2.0 data we further normalized the data with a* ***rank-based normalization method****.*

Sentence 2: *In the Bioinformatics field a great deal of interest has been given to* ***Non-negative matrix factorization technique NMF*** *due to its capability of providing new insights and relevant information about the complex latent relationships in experimental data sets.*

Sentence 3: *The relative performance of the four IBDQ dimensions in distinguishing best patients with minor symptoms from those with severe was studied by* ***discriminant analysis****.*

A method mention can be a terminology term such as the phrase **Multiplex Ligation dependent Probe Amplification** in Sentence 4.

Sentence 4: *Recently* ***< Multiplex Ligation dependent Probe Amplification > < MLPA >*** *has also been used to quantify copy number classes.*

A method expression can also be a verb phrase referring to an action performed during an experiment, such as the verb phrase *to search for* in Sentence 5.

Sentence 5: *These sequences were used **to search for** the nearly invariant nucleotides of the inverse core AAC and core GTT sites separated by a distance typical of previously identified attC sites to bp.*

In some cases, the context in which the method terminologies are used in the text can contain valuable information about the method mention, such as synonyms, definition, and other relations with entities in the text. The automatic extraction of terminologies has been the focus of many studies in the past [113, 192], but not many works have focussed on automatic extraction of method mentions.

We believe that the extraction of method expressions from scientific research papers can help to build lexical resources that can be used in various NLP tasks on using scientific research papers. For example, the automatic recognition of method expressions can help to easily detect method sentences and classify them into their rhetorical categories as in [1]. In addition, the automatic extraction of method mentions and the information surrounding them can be useful in dictionary building, ontology population and glossary creation. Also, it can help in the task of knowledge discovery from scientific papers, as any mention of methods and techniques used in that paper can easily be presented to readers without them having to read the whole text.

The extraction of contextual information around the method mentions can be useful in building question-answering and focussed text summarization systems. In the study presented below, we examine our method to extract method terminologies from sentences. Also we show how we can use a terminology context to extract other relevant information about the term. We define "other relevant information" to be the syntactic and semantic relationship between the term and other words. It may be a definition, a variant terminology and so on. This study doesn't take into account the extraction of verbs as method mention in scientific research papers.

4.2.1 Experiments

We used grammatical rules to extract the first category (method mentions that contain keywords, such as "algorithm", "technique", "analysis", "approach" and "method") and machine learning techniques to extract the second category of method mention (those that don't contain the above keywords). When a method keyword is not explicitly mentioned in a sentence containing a method mention, it is not obvious to apply general grammatical rules to recognize it as the words composing it can be of various forms. In the following sentences:

Sentence 10 : *Enault and colleagues proposed an improved **< phylogenetic profile >** based on a **< normalized Blastp bit score >**.*

Enault and colleagues proposed an improved **< phylogenetic profile >** based on a **< normalized Blastp bit score >**	Enault	O
	and	O
	colleagues	O
	proposed	O
	phylogenetic	B-method
	profile	I-method
	based	O
	on	O
	a	O
	normalized	B-method
	Blastp	I-method
	bit	I-method
	score	I-method
	.	O

Table 4.1: Representation of a method sentence in the BIO format.

Sentence 11 : *Another way to obtain suboptimal solutions from a* ***< HMM >*** *is to do* ***< HMM sampling >***.

Sentence 12 : *In this paper we introduce a* ***< Bootstrap procedure >*** *to test the null hypothesis that each gene has the same relevance between two conditions where the relevance is represented by the Shapley value of a particular coalitional game defined on a microarray data set.*

we can notice that the different method mentions—*< phylogenetic profile >, < normalized Blastp bit score > < HMM >, < Bootstrap procedure >*—are all of different word shapes. Also when most of them are nouns phrases, they can be confused with other noun phrases in the sentence. We thus considered that the extraction of this type of method mention can be viewed as a named entity recognition task [114, 140].

For this second task, we transformed each sentence into the BIO format (Table 4.1). There are 284 manually tagged terms, 122 sentences, 2871 words and punctuation marks.

In Table 4.2 we show the number of sentences in each dataset.

4.2.2 Rule-based Extraction

Most of the method mentions in the first category can be represented by the following examples:

1. *rank-based normalization method*
2. *HRV power spectral analysis*
3. *Non-negative matrix factorization technique*

Category (keywords)	Number of sentences	Proportion
Method	439	42%
Analysis	200	19%
Model	63	6%
Algorithm	73	7%
Approach	145	14%
Other (Machine learning corpus)	122	12%
Total	1040	100%

Table 4.2: Corpus statistics (combining both datasets)

4. *linear regression analysis*
5. *Newton-type algorithm*
6. *tube group amplification approach*
7. *progressive alignment algorithm*
8. *metabolite profiling approach coupling mass spectrometry*
9. *profile-based HMM method*
10. *multifactor-dimensionality reduction MDR method*

As we can notice these are simple grammatical patterns that can be extracted with simple rules.

1. They can start and continue either with an adjective or a noun.
2. They can continue either with an adjective or a noun.
3. They end with a method keyword.

These rules can be represented by the following regular expression, where Adjective and Noun are Parts-of-Speech:

(Adjective | Noun)+(method | analysis | algorithm | approach| model)

To extract such patterns, we first used the Genia tagger [177], a Part-of-Speech tagger trained on a biomedical corpus, to tag every word in the sentence with its Part-of-Speech. Then, we used the rules to extract all phrases and terminologies that correspond to the above mentioned patterns. The results are presented in Section 4.2.4.

4.2.3 Machine Learning and Feature Extraction

The choice of appropriate features and learning techniques are essential for the method mention recognition task.

Feature extraction

The feature pool includes:

1. Word feature
   The word itself.

2. Part-of-speech tags
   We use a POS tagger (the Genia tagger) to tag each word in the sentence.

3. Word-shape features
   We check whether the word is lower case, upper case or has both lower case and upper case letters. These features include: isAllCaps, StartWithCap, isAllLowerCase, isMixedCase.

4. Position features
   We check if the word is at the beginning of a sentence (BOS), at the end of a sentence (EOS), Not Beginning of Sentence (!BOS), Not Ending of Sentence (!EOS).

5. Token prefixes and suffixes features
   We extract the prefixes and suffixes for each word. These include the first four prefixes and the last four suffixes for each word.

6. Bigram features
   For each sentence we extract bigrams containing only nouns and adjectives.

Model training

To train the model, we used Conditional Random Field machine learning on 90% of the dataset and we tested on 10% of the dataset.

4.2.4 Results and Discussion

The experimental results used 924 for the rule-based task and 122 sentences for the machine learning task. Table 3 shows the performance of the two tasks. As we could expect, recall is very high (100%) for the rule-based task because every sentence in the dataset contains a method keyword. Our rules were able to recognize every noun phrase and adjective phrase ending with a method keyword. Precision is 85.40%. Some of the errors resulted from mistagging by the Genia tagger. We have spotted these errors and we will remove them in future study. For the machine learning task, precision is 81.8%, recall is 75% and F-score is 78.26%. Features such as word shape, POS, and noun-bigrams performed better than the position features. Also some of the errors came from the all-lower-case terms as they tend to be confused with similar words in the sentence. Table 4.3 summarizes the performance of both systems.

System	Precision (%)	Recall (%)	F-Measure (%)
Rule-based	85.40	100	91.89
Machine Learning	81.8	75.00	78.26

Table 4.3: Precision, Recall, F-measure of the Various Methods.

We believe that the scores can be improved with better linguistic filters in the case of the first task and a better feature selection for the second task.

Our work can be compared to [191] which uses CRFs for automatic keyword extraction from documents. They reported promising results (F-score of 51.25%) on a Chinese corpus.

It can also be compared to [192], which uses a multi-level term-hood method to extract terminology candidates from a bilingual corpus. Their system achieved an F-score of 79.6% with CRFs. When both works use similar techniques, most of the terminology extraction tasks were performed using a Chinese corpus.

4.3 Chapter Summary

In this chapter we have presented an overview of the task of information extraction in the biomedical domain. Then, we have explored two established techniques to automatically extract method terminologies from method sentences. Our results showed that we can extract many of these terms using simple grammatical patterns. A few other terms can be extracted with machine learning techniques (Conditional Random Fields). A brief study of the corpus showed that the context of the method mentions can help in the extraction of important information about the method term. The whole corpus can be used to extract information that is essential in the building of NLP resources such as glossaries, ontologies and specialist lexicons.

Chapter 5

Features for Text Categorization

5.1 Introduction

In Chapter 4 we discussed the need for features of written text that can be used as the basic components of methods that infer information implicit in the text. In this chapter we present some of the techniques used in feature selection for text classification. We show how they can be applied for the task of categorization of sentences into the IMRaD (Introduction, Methods, Results, and Discussion) scheme for classifying the rhetorical purpose of a sentence in scientific writing. We use some of these methods in the selection of features for the text classification task presented in Chapter 6. The purpose of feature selection is to select from the possible candidates the subset that provides higher accuracy (see Chapter 4) and good scalability (performance on previously unseen text) to the model. A scalable feature selection technique can deal with any amount of data while limiting the amount of resources needed for computation. Scalability can be attained with algorithms that use parallel computing techniques [160] or a feature filtering methodology such as Fast Correlation Based Filtering (FCBF) technique [27]. When the classifier is built with the most discriminative features, its accuracy can be improved. Noisy features are filtered out in order to reduce the error rate and increase F-measure score. The need to eliminate irrelevant features is even more acute with the Naïve Bayes algorithm, one of the most used text classifiers [26]. Moreover, when the dataset is large, the possible feature pool can comprise hundreds of thousands of distinct values. The computational resources that are needed to build a classifier can be demanding on conventional machines. A reduced feature set can have the effect of decreasing the computational cost and reduce the time required to build learning models. This benefit can be noticed even more when the classifying model requires the combination of many learning algorithms.

The rest of the chapter is organized as follows. The next section presents the principle for feature engineering. Then, we show some of the state-of-the-art feature selection approaches

that make use of statistical distribution of terms in the different classes. We conclude the chapter with a summary.

5.2 Text Feature Engineering

Feature engineering for text classification relies mostly on lexical and morphological attributes of the underlining domain. Relevant tokens are selected, either by focusing on individual words or a combination of words. Other grammatical features at the syntactical level can also be added to improve accuracy.

5.2.1 Unique Words

Relevant features can be selected from the set of individual words in the corpus. One way to reduce the feature space may consist of the selection of only the content words, which are words such as nouns, verbs and adverbs that refer to some objects, actions and characteristics. Content words in a domain specific corpus may also include special word forms such as a combination of letters and numbers like “U133A” or hyphened words like “Iso-Sensitest” in the biomedical domain.

5.2.2 Word Combination

Multiple related words can be merged together to reduce the feature space. A common practice is to force all words to lowercase, or to use stemming algorithms to merge related words. Inflected words can be replaced with their common base form or lemma. A lookup in a dictionary can help find the common lemma for related inflected words. Sometimes, when words having the same base forms mean different things, a common practice is to associate the base form with its part-of-speech. The combination of word - part-of speech (lexeme) can then be used as a single feature.

5.2.3 Word Phrase

One might want to collapse words that represent meaningful grammatical units such as noun phrases, phrasal verbs, prepositional verbs etc. Grammatical triples derived from dependency parse trees [37] have also proved to be relevant feature units for text classification [100].

5.2.4 Consecutive Sequence of Words: N-grams

Many consecutive words can also be combined together to form a single feature. Word sequence of length n (2 or more) can also be generated and used as a single feature. The advantage of using such a technique is that it helps incorporate context in the feature set, thus enabling the learning model to leverage on the position of words closed to one another in the text.

5.3 Principles of Feature Selection

The task of text classification requires that each data instance be transformed into its representative feature vector. In most situations, relevant features need to be selected from a large feature pool. Feature selection techniques then attempt to determine the set of features that are more likely to discriminate between the different classes. Most text classification features are word features, and some of them are more likely to correlate to a particular class distribution than others. Many feature selection methods have been proposed in order to make the feature pool as relevant as possible.

Feature selection is important in text classification tasks due to the high dimensionality of text features among which many are irrelevant. The representation of text instances into feature vectors can be done in two ways:

- Text can be represented as a bag-of-words, a method by which a document is tokenized into a set of words, along with their frequency. However, such a representation does not take into account the order in which words occur in the collection.

- Text can also be represented as strings in which each document is a sequence of words. Such sequences can be the occurrence of specific terms or keywords in the text.

The bag-of-words representation is used in most text classification tasks. In most text applications, the most basic feature selection technique is the removal of common words or stop words, which are words that are not specific to a given class. The incorporation or not of such words in the feature pool may not affect the classification outcome. Stop words include articles, prepositions, pronouns, etc. An example of stop words in shown in table 5.3.

Also stemming, which is used to merge different word forms into a single word, can serve to reduce the feature pool. Generally, singular, plural and different tenses can be collapsed into a single representative token.

Stemming and stop word removal are not limited to the problem of text classification, and are also applied in unsupervised applications such as indexing and clustering. While simple

a	an	are	as	at
and	be	by	for	from
hs	he	in	is	it
its	of	on	that	the
to	was	were	will	with

Table 5.1: Stop Words List

stemming or the removal of stop words may be sufficient to filter out noise from most text data, a more accurate and discriminatory feature selection method may be necessary in some cases. The most efficient technique would be the one that makes use of the class labels in the selection process. It is essential to make sure that features which are particularly skewed towards the presence of a particular class label be chosen for the learning process [2].

5.4 Feature Selection Approaches

Feature selection aims at deriving a subset of features from all possible features. The purpose of these approaches is to replace a complex and resource intensive classification model with a simpler and equally efficient model. We present in the following sections some of the most used feature selection approaches for text classification. We use some of these techniques in Chapter 6.

5.4.1 Gini Index

The gini-index measure [60] is used to quantify how a feature is different from other features. In the task of the classification of text in different categories, the conditional probability that a document is in a class c, given that it contains the feature f, is used to determine the importance of a feature for the underlying classes.

Let $p(1\,|\,t) \ldots p(k\,|\,t)$ be the fraction of class-label presence of the k different classes for the term t. In other words, $p(i\,|\,t)$ is the conditional probability that a document belongs to class i, given the fact that it contains the term t. Therefore, we have:

$$\sum_{i=1}^{k} p(i\,|\,t) = 1 \tag{5.1}$$

We compute, the gini-index $G(t)$ for the term t, as shown in 5.2:

$$G(t) = \sum_{i=1}^{k} p(i\,|\,t)^2 \tag{5.2}$$

The value of the Gini-index $G(t)$ is always in the range $(1/k, 1)$. Higher values of $G(t)$ are an indication that the discrimination power of the term t is high. A maximum value of $G(t)$ is obtained when a term (t) appears only in all documents that belong to a particular class. In this case $G(t)$ is 1. On the other hand, when documents containing a term (t) are evenly distributed among k different classes, the value of $G(t)$ is $1/k$.

One problem with this method is that it doesn't accurately reflect the discriminative power of a feature when the global class distribution is skewed. A normalized Gini-index can then be computed to reflect the discriminative power of an attribute.

Let $P_1 \ldots P_k$ represent the global distributions of the documents in the different classes. Then, we determine the normalized probability value $p'(i|\,|!t)$ as follows:

$$p'(i|t) = \frac{p(i|t)/P_i}{\sum_{j=1}^{k} p(j|t)/P_j} \tag{5.3}$$

Then, the normalized Gini-index is computed in terms of these normalized probability values as:

$$G(t) = \sum_{i=1}^{k} p'(i|t)^2 \tag{5.4}$$

By using global probabilities P_i, the Gini-index will likely reflect the class-distribution more accurately in the case that the class distributions in the document collection are biased. [2].

5.4.2 Information Gain

Information Gain [69] or entropy is another related measure used for text feature selection. Let $P(i)$ be the global probability of class i, and $p(i\,|\,t)$ be the probability of class i, given that the document contains the term t. Let $F(t)$ be the fraction of the documents containing the term t. The information gain measure $I(t)$ for a given term t is defined as follows:

$$\begin{aligned} I(t) = &- \sum_{i=1}^{k} p(i|t)\log(P(i)) + \\ &F(t)\sum_{i=1}^{k} p(i|t)\log(p(i|t)) + (1 - F(t))\sum_{i=1}^{k}(1 - p(i|t))\log(1 - p(i|t)) \end{aligned} \tag{5.5}$$

The greater the value of the information gain $I(t)$, the greater the discriminatory power of the term t.

5.4.3 Mutual Information

Mutual information is derived from information theory [158] and is used to model how the presence or the absence of a term helps to make the correct decision as whether an instance belongs to a class i or not. It computes the mutual information (MI) that exists between the features and the classes.

The point-wise mutual information $M_i(t)$ between the term t and the class i, is based on the degree of co-occurrence between term t and class i. The expected co-occurrence of class i and term t on the basis of mutual independence is given by $P_i F(t)$. Note that the true co-occurrence is of course given by $F(t)p_i(t)$. In practice, the value of $F(t)p_i(t)$ may be much larger or smaller than $P_i F(t)$, based upon the level of correlation between the class c and term t. The mutual information is defined in terms of the ratio between these two values.

$$M_i(t) = \log\left(\frac{F(t)p(i|t)}{F(t)P(i)}\right) = \log\left(\frac{p(i|t)}{P(i)}\right) \tag{5.6}$$

The term t is positively correlated to the class i, when $M_i(t) > 0$, and the term t is negatively correlated to class i, when $M_i(t) < 0$. $M_i(t)$ is specific to a particular class i. The overall mutual information needs to be computed as a function of the mutual information of the term t with the different classes. These are defined with the use of the average and maximum values of $M_i(t)$ over the different classes.

$$M_{avg}(t) = \sum_{i=1}^{k} P(i)M_i(t) \tag{5.7}$$

$$M_{max}(t) = \max_i\{M_i(t)\} \tag{5.8}$$

Either of these measures may be used in order to determine the relevance of the term t. The second measure is particularly useful, when it is more important to determine high levels of positive correlation of the term t with any of the classes [2].

5.4.4 χ^2-Statistic (Chi-Squared)

The Chi-Squared (χ^2) statistic is a different way to compute the lack of independence between the term t and a particular class. Let n be the total number of documents in the collection, $p(i|t)$ be the conditional probability of class i of documents which contain t, $P(i)$ be the global fraction of documents containing the class i, and $F(t)$ be the global fraction of documents which contain the term t. The χ^2-statistic between term t and class i is defined as follows:

$$\chi^2(t) = \frac{nF(t)^2(p(i|t) - P(i))^2}{F(t)(1 - F(t))P(i)(1 - P(i))} \tag{5.9}$$

As in the case of the mutual information, we can compute a global χ^2-Statistic from the class-specific values. We can use either the average or maximum values in order to create the composite value:

$$\chi^2_{avg}(t) = \sum_{i=1}^{k} P(i)\chi^2_i(t) \tag{5.10}$$

$$\chi^2_{max}(t) = \max_i \{\chi^2_i(t)\} \tag{5.11}$$

We note that the χ^2-statistic and mutual information are different ways of measuring the correlation between terms and categories. One major advantage of the χ^2-statistic over the mutual information measure, is that it is a normalized value, and therefore these values are more comparable across terms in the same category [2].

5.5 Chapter Summary

Feature selection is an important stage of text classification tasks. When simple stemming and stop word removal are not enough to improve the performance of the learning model, a more complex selection technique may be required to filter out noisy features from the feature pool. The most used methodologies rely on a probability distribution over the dataset to compute the discriminative factor of potential features. In Chapter 6, we will make use of some of these techniques in the task of sentence classification for citation linkage.

Chapter 6

Sentence Classification for Citation Linkage

6.1 Introduction

Classifying sentences into the four rhetorical categories, Introduction, Methods, Results, and Discussion (IMRaD), has been an important pre-processing step when trying to understand the structure of scientific text. For instance, sentences that are deemed to be in the discussion category and then further sub-classified as conclusions of a research paper can be used to validate or refute an hypothesis contained in the introduction sentences in that paper. Therefore, in order to understand the argumentation flow in scientific publications, we need to understand how different sentences fit into the complete rhetorical structure of scientific writing.

For our purposes here, we hypothesize that citation linkage should primarily be viewed as the matching of units of text that convey meaningful information when taken in isolation. An underlying assumption that will reduce the scope of the citation linkage task that is investigated in later chapters is that citation sentences that make (contain) reference to method mentions will likely be linked to sentences with similar method terms in the cited paper. So classifying sentences into rhetorical categories that include the method class may be of great value for the citation linkage task. To allow sentence level rhetorical classification of scientific discourse to be a step in the citation linkage task we need to build a rhetorical classifier because none are publically available. In doing so, we make an improved classifier.

To perform sentence classification, we would like to engage supervised machine learning techniques. One necessary requirement for this task is to have a large corpus annotated with the appropriate classification tags. The building of large-scale corpora of text for use in some specific text analytic purpose is typically affected by the following constraints: budgets, time

constraints, and quality control. These constraints make it difficult for researchers to build domain-specific corpora that suit their needs. In the biomedical domain, some corpora already exist, but many of these corpora are still limited and cannot be generalized to every context. The task of sentence classification in various rhetorical categories is often performed on *ad hoc* corpora derived from a limited number of papers that don't necessarily represent all of the text in the biomedical domain. Annotators often agree on a limited number of sentences to be included in the corpus. For instance, the corpus used in [1] for the task of sentence classification into the IMRaD categories is composed of only 1131 sentences. These sentences simply can't represent all possible sentences in biomedical publications, hence, the need for a corpus that is large enough to include as many relevant sentences as possible.

As presented in Chapter 3, we hypothesize that using a simple linguistically-based heuristic, we can build a significantly larger corpus that will be less resource-consuming and will represent a wider range of publications in the biomedical literature. We believe that we can use some simple linguistic filters to collect a large corpus comprising sentences that belong to specific categories of the IMRaD rhetorical structure of the biomedical research text without having domain experts annotate them. For this purpose, we have collected pairs of sentences where the second sentence begins with "This method...", "This result...", "This conclusion...". Our hypothesis is that the first sentence in each pair is a sentence that can be categorized respectively as "Method", "Result" and "Conclusion" sentences. We are aware that the co-reference could be referring to a sentence prior to the immediately precedent sentence, but we believe that this is rare.

We have a number of motivations for this work. First, sentences are the basis for most text mining and extraction systems. The second motivation is that biomedical texts are the reports of scientific investigations and their discourse structures should represent the scientific method that drives these investigations. The third and last motivation is that categorizing sentences into the IMRaD categories can help in the task of extracting knowledge discovery elements from scientific papers, as any mention of results or the methods used, as well as the claims mentioned in the paper, can easily be summarized and presented to readers without them having to read the whole text.

The contribution of this work is twofold. First, we have used a simple linguistic filter to automatically select thousands of sentences that have a high probability of being correctly categorized in the IMRaD scheme, and second, we have used machine learning techniques to classify sentences in order to validate our hypothesis that this linguistic filter works. The rest of the chapter is organized as follows. The next section reviews some related work. In Section 6.3, a detailed methodology of corpus construction and sentence classification techniques is presented. In Section 6.4, the results are described. We conclude the chapter by a summary

and directions for future work.

6.2 Related Work

The classification of sentences from scientific research papers into different categories has been investigated in previous works. Many schemes have been used and currently no standard classification scheme has been agreed upon. In [171], a classification scheme termed Argumentative Zoning (AZ) is used to model the rhetorical and argumentative aspects of scientific writing in order to easily detect the different claims that are mentioned in a scientific research paper. AZ has been modified for the annotation of biology articles [129] and chemistry articles [173].

Scientific discourse has also been studied in terms of speculation and modality by Kilicoglu et al. [84] and Medlock et al. [118]. Also, Shatkay et al. [159] and Wilbur et al. [181] have proposed an annotation scheme that categorizes sentences according to various dimensions such as focus, polarity, and certainty. Many annotation units have also been proposed in previous studies. Sentence level annotation is used by Teufel [171] for the task of citation function classification, whereas de Waard et al. [39] used a multi-dimensional scheme for the annotation of biomedical events (bio-events) in scientific text.

Also, Liakata et al. [101] attempt to classify sentences into the Core Scientific Concept (CoreSC) scheme. This classification scheme consists of a number of categories distributed into hierarchical layers. The first layer consists of 11 categories which describe the main components of a scientific investigation, the second layer is comprised of properties of the first layer categories (e.g., Novelty, Advantage), and the third layer provides identifiers that link together instances of the same concept. Some other recent works have focussed on the classification of sentences from biomedical articles into the IMRaD (Introduction, Methods, Research, and, Discussion) categories. Agarwal and Yu [1] use a corpus of 1131 sentences to classify sentences from biomedical research papers into these categories. In this study, sentence level annotation is used and multinomial Naïve Bayes machine learning has proved to perform better than simple Naïve Bayes. The authors report an overall F-measure score of 91.55% with a mutual information feature selection technique. The present study provides an alternative way to build a larger IMRaD annotated corpus, which combined with existing corpora achieves a better performance.

6.3 Methodology

As presented below in Section 6.4, the training corpus used in the sentence classification task is a silver standard corpus extracted from scientific text using linguistic heuristics intended to

recognize rhetorical schemes. The method to create this silver standard corpus has been detailed in Section 3.4. Let's recall that we work with the belief that a sentence that appears in the co-referential context of the co-referencing phrase "This method", will likely talk about a methodology used in a research experiment and reported in the paper. Similarly, a sentence that starts with the expression "This result" is likely to refer to an experimental result context, etc. We define the co-referential context of these phrases to be a small number of sentences preceding the sentences in which these "This" references occur. Since most of the demonstrative pronouns are co-referential, a sentence that begins with the demonstrative noun phrase "This method" or "This result" is co-referential and its antecedents are likely to be found in previous sentences. Torii et al. [176] has reported that nearly all antecedents of such demonstrative phrases can be found within two sentences. On the other hand, Hunston [74] reported that interpreting recurring phrases in a large corpus enables us to capture the consistency in meaning as well as the role of specific words in such phrases. So, the recurring semantic sequences "this method" or "this result" in the PubMed corpus can help us to capture valuable information in the context of their usage.

For instance, to collect sentences that belong to the "Result" category, our target candidates will be those sentences that come immediately before sentences starting with "This result…", as shown in Example 1. Similarly, to collect sentences that belong to the "Method" category, our target candidates will be those sentences that come immediately before sentences starting with "This method …" etc. A similar technique was used in Section 4.2 (and also provided in [72], to build a dataset for method mention extraction from biomedical research papers.

Category	Number of sentences	Proportion
Method (Met)	3163	31.9%
Result (Res)	6288	62.69%
Conclusion (Con)	534	5.39%
Total	9985	100%

Table 6.1: Corpus statistics

As stated in Chapter 5, a number of feature selection methodologies have proved to increase the accuracy of sentence classification models. Chi-Squared and Mutual Information use the statistical distribution of terms in the text corpus to reduce the feature pool into a more relevant and discriminatory set. In [1], Argawal and Yu used Chi-Squared and Mutual Information to extract features from a biomedical text corpus in their attempt to classify sentences into the IMRaD rhetorical categories. In experimenting with different top features, they obtained better performance using the top 2500 features.

We have used these sets of features extracted from the Agarwal and Yu constructed IMRaD corpus [1]. The reason for this choice is to be able to validate our claim against previous works in which this set of features proved to achieve high accuracy and F-measure scores. We also used verb tense features as some categories may be associated with the presence of the present tense or the past tense in the sentence. We used the Stanford parser [87] to identify the presence of these tenses.

The feature pool comprises a combination of individual words as well as bi-grams and tri-grams. A feature that indicates the presence of citations in the sentence is also used as it can be an important feature for distinguishing some categories; for example, citations are more frequently used in Background/Introduction than in Results. All numbers were replaced by a unique symbol #NuMBeR. Stop words were not removed since certain stop words are also more likely to be associated with certain IMRaD categories. Words that refer to a figure or table are not removed, since such references are more likely to occur in sentences indicating the outcome of the study. Methods for training supervised machine-learning systems on non-annotated data, were presented in [188], which assumed that in a full-text, IMRaD-structured article, the majority of sentences in each section will be classified into their respective IMRaD category. For example, sentences that are used in the Method section are assumed to belong to that category, unless proved otherwise. Also Agarwal and Yu used the same method to build a baseline classifier that achieved about 77.81% accuracy [1].

Our method for categorizing sentences into the IMRaD categories does not work for the Introduction category, so we do not attempt to find sentences in this category. From the Agarwal and Yu IMRaD dataset [1], we extracted instances belonging to the Method, Result and Conclusion categories. We have used this dataset to build a model. This model is used to classify instances of the unannotated PubMed dataset. When the confidence level of the classification is greater than a given value (98%), this instance is added to the model validated self-annotated corpus. The fine-grained machine-validated dataset is presented in Table 6.2.

Category	Number of sentences	Proportion
Method (Met)	878	23.59%
Result (Res)	2399	64.50%
Conclusion (Con)	443	11.90%
Total	3719	100%

Table 6.2: Corpus statistics

For all supervised classifications, we used two algorithms: multinomial Naïve Bayes and Support Vector Machine (SVM). Both multinomial Naïve Bayes and SVMs are widely used supervised machine-learning algorithms. The probabilistic framework of Multinomial Naïve

Classification with Multinomial Naïve Bayes

Class	Prec.	Recall	F-Meas.
method	0.923	0.661	0.77
result	0.627	0.813	0.708
conclusion	0.68	0.821	0.744
Average	0.779	0.74	0.744

Classification with SVM

Class	Prec.	Recall	F-Meas.
method	0.818	0.521	0.636
result	0.511	0.908	0.654
conclusion	0.923	0.226	0.364
Average	0.72	0.621	0.604

Table 6.3: Precision, Recall, F-measure: Classifier trained with the auto-annotated dataset and tested with the IMRAD dataset (Method, Result, Conclusion)

Bayes represents a multinomial distribution of words in a sentence that captures word frequency. The multinomial model has been shown to perform well in document classification [115]. We used the implementation of both algorithms provided by the open-source Java-based machine-learning library Weka 3.7 [65].

6.4 Results and Discussion

In the first classification task, we have used the self-annotating dataset to build a model and we test it with instances from the Agarwal and Yu corpus [1]. Only three categories (Method, Result, Conclusion) are used. The results are presented in Table 6.3. Because the features extracted from the training and test sets are lexically based, the small corpora involving different biomedical subject matter in the PubMed dataset have an effect on these results. To reduce this effect the two corpora are merged into what we call the machine validated dataset. The self-annotating dataset does not contain instances categorized as Background (Introduction), so instances of this category in the Agarwal an Yu corpus are not included. In the second classification task, a classifier is trained and tested with the machine validated dataset using 10 fold cross-validation. The cross validated models achieve an average F-measure score of 97%. See Table 6.4. In the third classification task, a classifier is trained and tested with the machine validated dataset together with the Background category instances taken from the Agarwal and Yu dataset. We applied 10 fold cross-validation and achieved an overall F-score of 93.6% with Multinomial Naïve Bayes and 95% with SVM. See Table 6.5.

When we select only the sentences that are classified into the correct categories using the Agarwal and Yu corpus as a training set and the Agarwal and Yu self-annotated dataset as a test set, we notice that more than 3700 sentences are classified in the right categories (more than three times the Argawal and Yu data set size). A 10-fold cross validation classification on this data using Naïve Bayes and SVM achieve an overall score of 95%, i.e., more than 3 percentage points above the Agarwal and Yu score.

Classification with Multinomial Naïve Bayes

Class	Prec.	Recall	F-Meas.
method	0.981	0.957	0.969
result	0.966	0.992	0.979
conclusion	0.98	0.885	0.93
Average	0.971	0.971	0.971

Classification with SVM

Class	Prec.	Recall	F-Meas.
method	0.986	0.984	0.985
result	0.988	0.995	0.992
conclusion	0.986	0.95	0.968
Average	0.987	0.987	0.987

Table 6.4: Precision, Recall, F-measure: Classifier trained with the machine validated dataset using 10 fold cross-validation (Method, Result, Conclusion)

Classification with Multinomial Naïve Bayes

Class	Prec.	Recall	F-Meas.
background	0.884	0.711	0.788
method	0.961	0.949	0.955
result	0.952	0.991	0.971
conclusion	0.853	0.83	0.842
Average	0.937	0.938	0.936

Classification with SVM

Class	Prec.	Recall	F-Meas.
background	0.796	0.765	0.78
method	0.97	0.983	0.976
result	0.985	0.988	0.986
conclusion	0.854	0.844	0.849
Average	0.95	0.95	0.95

Table 6.5: Precision, Recall, F-measure: Classifier trained with the machine validated dataset using 10 fold cross-validation (Background, Method, Result, Conclusion)

A comparison with the results obtained in [1] using a Multinomial Naïve Bayses classifier and a similar feature selection technique leads us to suggest that the proposed self-annotating corpus building method has great potential over human annotation methods and should therefore be used whenever possible.

6.5 Chapter Summary

Sentence classification is important in determining the different components of argumentation. We have suggested a self-annotation method to annotate sentences from scientific research papers classes into their IMRaD categories providing an automatic method to build large annotated corpora. We used a linguistic heuristic to extract a corpus from the PubMed repository. Our results show that it is possible to extract automatically a self-annotated corpus from a large repository of scientific research papers. One advantage of such method is that, it is less time-consuming than a human annotated corpus. In Chapter 8 we will use the classification model learned with this corpus to classify sentences from cited papers into the IMRaD's method category. We believe that citing sentences that are in the "Method" rhetorical category will likely be linked to sentences of this category in the cited paper.

Chapter 7

Text Similarity Measures and Evaluation

7.1 Introduction

The linkage between the citation text and the target text it refers to in the cited paper requires the choice of appropriate text comparison techniques that evaluate the degree of similarity between two spans of textual content. In this perspective, finding the best matching citing sentences for a given citation sentence will involve the use of one or more text similarity measures to discriminate between the candidate sentences.

Previous works have shown that text similarity can be determined at the word level [145], as well as at longer text unit levels such as clause, sentence, paragraph and document [156]. Similarity between words can be a initial step for computing the similarity between longer units of text. Words can be lexically or semantically related or similar.

Most of the similarity measures and techniques presented in this chapter are derived or adapted from Gomaa et al.'s survey of text similarity approaches [61]. In that work, they provided a comprehensive survey of these measures and organized these approaches into four main categories such as String-Based Similarity, Corpus-Based Similarity, Knowledge-Based Similarity and Hybrid Similarity Measures. This chapter, drawing heavily on this work, summarizes most of the approaches detailed in this survey and maintains the organization that the authors provide. Text summary related measures have been added to their string-based similarity measures.

Lexically similar words share a common sequence of characters and are said to be syntactically related. In this regard, the degree of similarity between words is determined with string-matching algorithms that operate on string sequences and character composition.

Semantic similarity measure is concerned with the actual meaning of the words and the context of their usage. Words are semantically similar if they mean the same thing, are opposite to each other, are used in the same way, are used in the same context, or one is a type of another.

Semantic similarity can be corpus-based or knowledge-based. Corpus-based similarity between words is determined by using information extracted from large corpora, whereas knowledge-based similarity techniques compute the degree of similarity between words using information derived from semantic networks or ontologies.

The present thesis uses a combination of similarity measures as the basis for the linkage operation that links text contents in the scientific research literature. Different experiments using these similarity measures or techniques are reported in Chapter 8.

7.2 Similarity Measures using String Matching

String-based similarity can be divided into character-based and term-based distance measures. Character-based similarity focuses on the difference between individual characters and the term-based distance computes the difference between string tokens of more complex structures such as words, word-variants, topical terms, etc.

7.2.1 Character-Based Similarity Measures

Character-based similarity measures include:

Longest Common Substring (LCS) calculates the distance between two inputs by comparing the length of the longest consecutive sequence of matching characters[63].

Damerau-Levenshtein computes the distance between two strings by counting the minimum number of operations needed to transform one string into the other. An operation is defined as an insertion, deletion, substitution of a single character, or a transposition of two adjacent characters [66, 146].

Jaro is computed based on the number of characters that are common to the input strings as well as the order in which these characters appear in the strings. The computation takes into account word spelling variations and it is commonly used in record linkage [79].

Jaro-Winkler [184] is an extension of the Jaro distance metric technique. It was proposed for name comparison and considers the exact match between the initial characters of the two inputs [185].

Monge-Elkan [131] is a hybrid method which tokenizes two inputs and finds word pairs that have the highest string similarity scores (usually greater than zero). It then sums up and normalizes these scores and assigns it as the distance between the inputs.

Needleman-Wunsch algorithm uses a dynamic programming method to perform a global alignment of two sequences. It applies intermediary alignment searches over parts of the sequences to find the best alignment over the entire text inputs. It is suitable when the two

sequences are of similar length, with a significant degree of similarity throughout the string sequences [136].

Smith-Waterman is another example of dynamic programming. A local alignment approach is used to find the best alignments for similar chunks of two sequences. It is useful for sequences that are dissimilar, but are suspected to contain regions of similarity or similar sequence motifs within their larger sequence context [164].

N-gram is a contiguous sequence of n items from a given text sequence. N-gram similarity algorithms compare the n-grams of characters or words in two strings. To compute N-gram distance, the number of similar n-grams is divided by the maximal number of n-grams [12].

7.2.2 Similarity Measures using Term-based Approaches

A term represents a single word or a word-compound formed by two or more contiguous words. A vector can be used to represent a piece of text. When comparing two pieces of text, length n vectors are used to represent the texts. The length n is determined by the n distinct terms in the union of the two texts. A vector then indicates the existence (0 or 1) or the number of occurences of each term in one of the texts.

Block Distance, also known as Manhattan distance, and L1 distance, is the distance that would be traveled to get from one data point to the other if a grid-like path is followed. This block distance between two vectors is the sum of the (absolute value) differences of their corresponding components [89].

Cosine Similarity Two texts can be treated like two vectors in an n-dimensional space. The cosine of the angle between the two vectors is an indication of the similarity of the two texts.

Dice's Coefficient is defined as twice the number of common terms in the compared strings divided by the total number of terms in both strings [41].

Euclidean Distance or L2 distance is the square root of the sum of squared differences between corresponding elements of the two vectors representing the two text inputs.

Jaccard Similarity measure returns the proportion of the number of shared terms among all unique terms in both strings [78].

Matching Coefficient is a very simple vector based approach. Each text input is first represented as a multi-dimensional vector. Then, it returns the number of terms for which both vectors are non zero [165].

Overlap Coefficient is similar to Dice's coefficient, but it considers two strings a full match if one is a subset of the other [108].

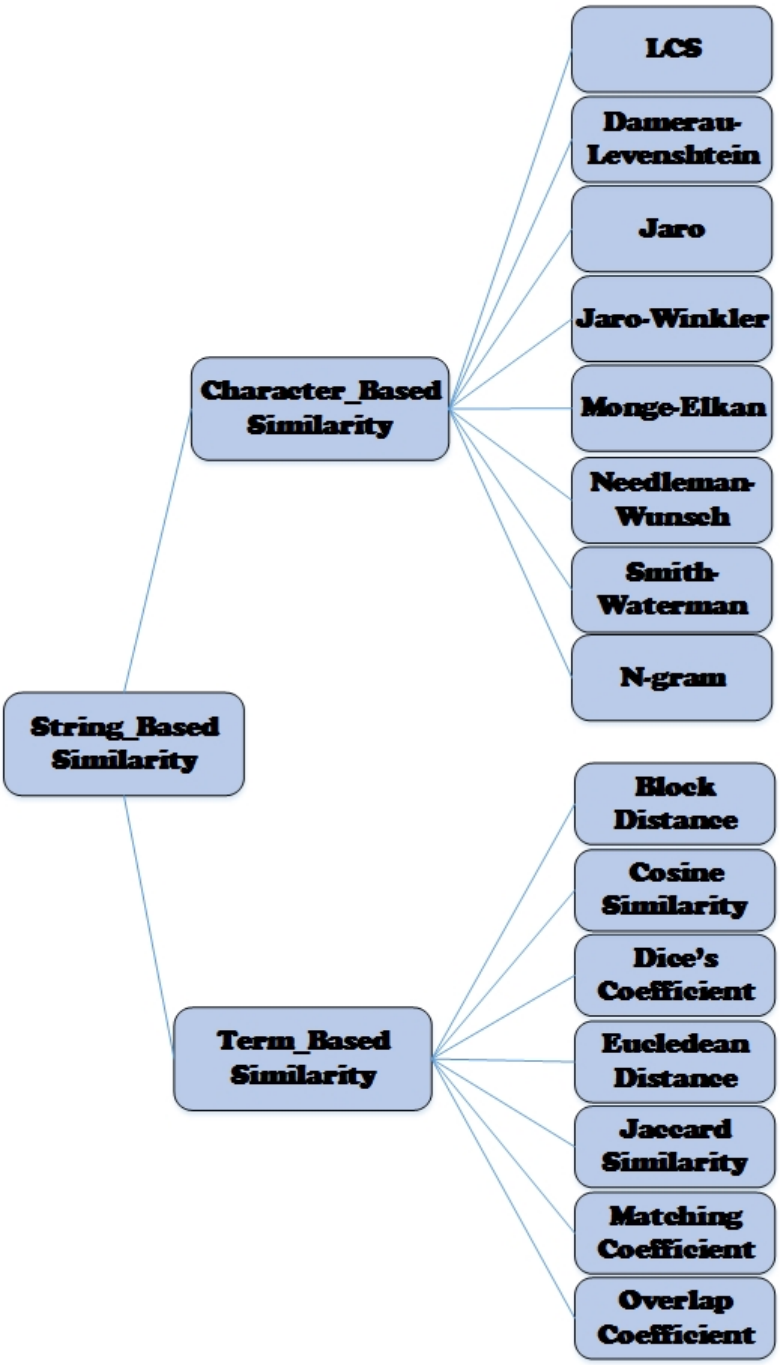

Figure 7.1: String-based similarity measures organized by character-based and term-based distance measures (adapted from [61])

7.2.3 Text Summary Related Measures

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [103] comprises a set of measures to automatically assess the quality of a summary or a translation by comparing it to the ideal summaries or translations produced by humans. These measures are based on overlapping units of text such as n-gram, word sequences, and word pairs. ROUGE measures include ROUGE-N, ROUGE-L, ROUGE-W and ROUGE-S.

ROUGE-N uses N-Gram co-occurrences to compute the similarity between a reference and candidate summaries. Summaries that share more N-Grams with the reference are favored.

ROUGE-L uses common sequences with maximum length for summarization evaluation. The technique first computes the Longest Common Subsequence (LCS) between a reference summary and machine generated summaries. The longer the LCS of two summary sentences is, the more similar the two summaries are.

ROUGE-S, also called skip-bigram-based co-occurence statistics uses overlap of word pair occurrences in reference and candidate sentences to estimate the similarity between a reference translation and a set of machine generated translations.

ROUGE-W is a weighted version of longest common subsequence. It computes the number of consecutive characters in each match, and gives a higher score for those matches that have a larger number of consecutive characters in common.

7.3 Corpus-based Similarity Measures

Corpus-based semantic similarity is based on the assumption that the relationship that exists between words can be determined within the context of their usage. Therefore, how a given word is used in a corpus can help evaluate how similar it is to other words in the corpus. The information gained from a large corpus is then used to compute a representative vector for words in the corpus. A comparison between two vectors is enough to assert whether their corresponding words are similar or not. Figure 7.2 shows the corpus-based similarity measures.

Hyperspace Analogue to Language (HAL), also known as semantic memory, relies on the assumption that words with similar meaning constantly occur closely [106, 107]. For example, in a large biomedical text corpus, we might expect to see words such as *gene* and *protein* to appear close to each other. HAL creates a semantic space from word co-occurrences. A squared matrix of unique words from a corpus is constructed, where each word is represented as columns and rows. The matrix shows the strength of association between the word represented by the row and the word represented by the column.

For each focus word in a running text, a ten word window is set to record co-occurring

words that appear in its neighbourhood. A word is then represented by a vector having for components the weighted values of the distance of the co-occurrences to the focus word. The word's meaning is mostly determined by its close neighbours. Information regarding whether the co-occurring words appear before or after the focus word is also taken into account.

Latent Semantic Analysis (LSA) [93] is the most popular technique of corpus-based similarity. LSA assumes that words that are close in meaning will occur in similar pieces of text. A matrix containing word counts per paragraph- (rows represent unique words and columns represent each paragraph)- is constructed from a large piece of text. The technique of Singular Value Decomposition (SVD) is used to reduce the number of columns. The similarity structure among rows are preserved in this reduction process. Word similarity is then determined by computing the cosine of the angle between the two vectors formed by any two rows.

Generalized Latent Semantic Analysis (GLSA) is a framework for computing semantically motivated term and document vectors [112]. This is an extension to the LSA method with emphasis on term vectors rather than the dual document-term representation. Along this line, similarity between terms is determined using dimensionality reduction techniques and a term-document matrix to provide the weights in the linear combination of term vectors.

Explicit Semantic Analysis (ESA) [52] is a measure for computing the semantic relatedness between two arbitrary texts. One technique is based on representing terms or texts as high-dimensional vectors. Each vector entry represents the TF-IDF weight between the term and a Wikipedia article. The cosine score between corresponding vectors is used to assess the semantic relatedness between two terms (or texts).

Cross-Language Explicit Semantic analysis (CL-ESA) [147] is a multilingual generalization of ESA. The idea is to represent a document as a language-independent concept vector. For instance, Wikipedia, which has the same documents in many languages, can be used to transform any text into a representative vector, irrespective of the target language . The relatedness of two documents in different languages is assessed by the cosine similarity between their corresponding vector representations.

Pointwise Mutual Information - Information Retrieval (PMI-IR) [178] is a method for computing the similarity between pairs of words. It uses AltaVista's Advanced Search query-syntax to calculate probabilities. It assumes that when two words often co-occur near each other on a web page, their PMI-IR similarity score will be high.

Second-Order Co-Occurrence Pointwise Mutual Information (SCO-PMI) [76, 77] is a semantic similarity measure that uses point-wise mutual information to sort lists of important neighbouring words of the two target words in a large corpus. SOC-PMI can be used to calculate the similarity between two words that do not co-occur frequently but co-occur with the same neighbouring words.

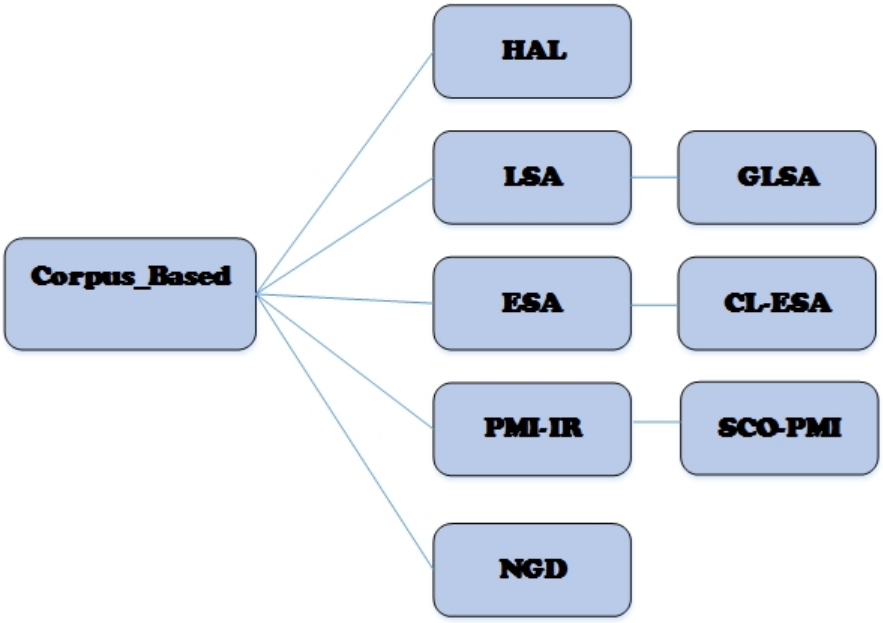

Figure 7.2: Corpus-based similarity measures (adapted from [61])

Normalized Google Distance (NGD) [30] is a semantic similarity measure computed using the number of hits returned by the Google search engine for a given set of keywords. It assumes that keywords with the same or similar meanings in a natural language tend to be "close" in units of Google distance, whereas words that are dissimilar tend to be farther apart. Specifically, the Normalized Google Distance between two search terms x and y is :

$$NGD(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log M - \min\{\log f(x), \log f(y)\}} \tag{7.1}$$

where M is the total number of web pages searched by Google, $f(x)$ and $f(y)$ are the number of hits for search terms x and y, respectively, and $f(x,y)$ is the number of web pages on which both x and y occur. An infinite NGD between two search terms x and y is attained when they never occur together on the same web page, but do occur separately. On the other hand, terms that always occur together have a zero NGD.

7.4 Similarity Measure using Knowledge-based Information

Knowledge-based semantic similarity approaches identify the similarity between words based on information derived from semantic networks [122]. WordNet [123] is the most popular semantic network for measuring the knowledge-based similarity between words. WordNet is also a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each representing a distinct concept. Synsets are inter-

connected by means of conceptual-semantic and lexical relations. Knowledge-based similarity exploits the relationship between words in a semantic network in order to compute a measure that asserts their "closeness".

Knowledge-based semantic similarity measures can be divided roughly into two groups: measures of *semantic similarity* and measures of *semantic relatedness*. In the semantic similarity metric computation, concepts that are similarly related are accessed on the basis of their likeness. Semantic relatedness, on the other hand, is a more general notion of the relationship that exists between words, and it is not specifically tied to the shape or form of the concepts.

Semantic similarity between two words can also be viewed as a kind of relatedness in a more specific sense. This can be assessed in terms of a range of relationships between the set of concepts the words identify with. This includes extra similarity relations such as is-a-kind-of, is-a-specific-example-of, is-a-part-of, is-the-opposite-of [143].

There are six measures of semantic similarity, three of which are based on information content: Resnik (res) [151], Lin (lin) [104] and Jiang & Conrath (jcn) [81]. In Wordnet, each word is associated with a synset or concept, which in turn relates to other concepts in an hierarchical pattern. The information content of a concept c is defined as the negative log of its probability, computed based on frequency counts [144]. These counts are often obtained from corpora such as the Brown Corpus[51], the Penn Treebank[110], or the British National Corpus[98].

$$IC(c) = -\log P(c) \tag{7.2}$$

The other three measures are based on path length: Leacock & Chodorow (lch) [97], Wu & Palmer (wup) [186] and Path Length (path)[35].

The value returned by *res* is equal to the information content (IC) of the Least Common Subsumer (most informative subsumer). Least Common Subsumer of two concepts A and B is "the most specific concept which is an ancestor of both A and B" [145] The *lin* and *jcn* measures augment the information content of the Least Common Subsumer with the sum of the information content of concepts (synsets) A and B themselves. The **lin** measure scales the information content of the Least Common Subsumer by this sum, while *jcn* returns the difference of this sum and the information content of the Least Common Subsumer.

The **lch** measure returns a score denoting how similar two word senses are, based on the shortest path between the senses as well as the maximum depth of the conceptual categories (taxonomy) in which the senses occur.

The **wup** measure returns a score denoting how similar two word senses are, based on the depth of the two senses in the taxonomy and that of their Least Common Subsumer. The **path** measure returns a score that shows the degree of similarity between two word senses, based on the shortest path that connects the senses in the *is-a (hypernym/hypnoym)* taxonomy.

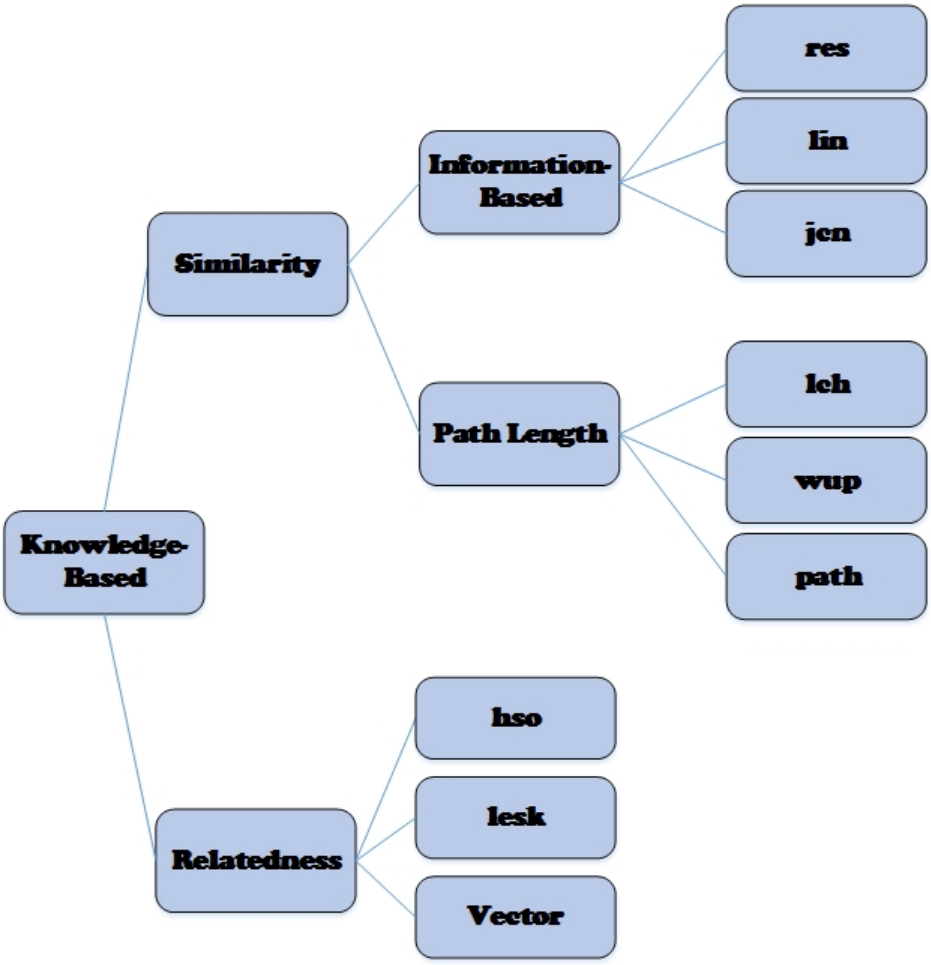

Figure 7.3: Knowledge-based similarity measures (from [61])

There are three measures of semantic relatedness: St.Onge (hso) [70], Lesk (lesk) [10] and vector pairs (vector) [142]. The **hso** measure is based on the lexical chains linking the two word senses. The technique tries to find these chains by means of three classes of relations, namely: *extra-strong*, *strong*, and *medium-strong*. The **lesk** method computes word relatedness by finding overlaps in the glosses of their two synsets. The relatedness score is the sum of the squares of the overlap lengths. To compute the **vector** measure, a co-occurrence matrix is created for each word used in the WordNet glosses. The gloss/concept in the knowledge base is then represented with a vector that is the average of these co-occurrence vectors. The relationship between two words is assessed by the similarity value between the glosses or the concepts they belong to.

7.5 Similarity Measures using Hybrid Approaches

Hybrid techniques combine many similarity measures to determine the similarity between texts. Eight semantic similarity measures were tested in [122]. Two of these measures were corpus-based measures and the other six were knowledge-based. Firstly, these eight algorithms were evaluated separately, then they were combined together. The best performance was achieved using a method that combines several similarity metrics into one. A method for measuring the semantic similarity between sentences or very short texts, based on semantic and word order information, was presented in [99]. Semantic similarity is derived from a lexical knowledge base and a corpus. The technique evaluates the impact of individual word order as well as word pair order on sentence meaning.

In the Semantic Text Similarity (STS) task [77], similarity between small units of text is determined by combining semantic and syntactic information. Two mandatory functions (string similarity and semantic word similarity) and an optional function (common-word order similarity) were used. The STS method achieved a very good Pearson correlation coefficient for 30 sentence pairs and outperformed the results obtained in [99].

The authors of [3] presented an approach that combines corpus-based semantic relatedness measures along with the knowledge-based semantic similarity scores that were obtained for words falling under the same syntactic roles in both sentences. The scores are used as features for machine learning models, such as linear regression, and bagging models. The system returns a single score that denotes the degree of similarity between sentences. This approach of combining knowledge-based similarity measures and corpus-based relatedness measures showed a significant improvement over the corpus based measure taken alone.

Promising correlation results were achieved in [22] by combining two modules. The first module calculates the similarity between sentences using N-gram based similarity, and the second module calculates the similarity between concepts in the two sentences using a concept similarity measure derived from WordNet.

A system named UKP was introduced in [11]. It used a simple log-linear regression model based on training data, to combine multiple text similarity measures. These measures were string similarity, semantic similarity, text expansion mechanisms and measures related to structure and style. The UKP final models consisted of a log-linear combination of about 20 features, out of the possible 300 features implemented. These models achieved reasonable correlation results.

7.6 Chapter Summary

We have presented a survey of the commonly used text similarity measures. The techniques used to compute these measures can be as simple as finding the number of common terms in two text inputs or as complex as combining statistical metrics and information retrieved from semantic networks. In the task of linking citation sentences to their equivalent sentences in cited papers, the use of adequate measures is necessary and it is not obvious which one of the techniques presented in this chapter is the de-facto choice for a best citation linkage. We have concluded that hybrid similarity measures would be an idea worth investigating. In Chapter 8 we will use an hybrid method to combine many measures to build machine learning linear regression models to predict the similarity strength between citation sentences and candidate cited sentences in the cited papers.

Chapter 8

Experiments and results: Citation linkage as a machine learning task

8.1 Introduction

The citation linkage task can be considered as a matching operation between a citation sentence and one or more sentences it refers to in the cited paper. In this chapter we use a machine learned model to perform this matching procedure. The machine learning techniques that we consider below are supervised, so human annotated data is required to build the citation linkage model. The data instances that have been annotated for this task are given a rating score by the human annotators reflecting their confidence in the citation match. See Section 3.5 for details. Because the data is graded and not binary, we will use regression techniques rather than classification techniques to generate the machine learned model. For this purpose, we evaluate three machine learning methods: linear regression, SVM linear regression, and ordinal regression.

As important as the human annotated data are the features that will be used by the regression techniques to build the citation linkage model. The features that we choose measure various similarity aspects of the text (character-level, word-level, syntactic-level, and semantic-level). The next section reviews various approaches that use these hybrid features to combine the many similarity features to build learned models.

The sections that follow then examine an in depth assessment of the model that best determines the linkage task. After determining which of the two learning methods produces the better learned model, it is further examined to ascertain how well it ranks the sentences that it suggests are the candidate linkage sentences.

8.2 Related Work using Hybrid Approaches for Text Similarity Detection

The state-of-the-art text similarity detection techniques have combined many measures in the form of linear combinations of more advanced machine learning algorithms. Along this line, the task of text similarity detection can employ many similarity measures to determine the similarity between texts. For instance eight semantic similarity measures were tested in [122]. Two of these measures were corpus-based measures and the other six were knowledge-based. Firstly, these eight algorithms were evaluated separately, then they were combined together. The best performance was achieved using a method that combines several similarity metrics. A method for measuring the semantic similarity between sentences or very short texts, based on semantic and word order information was presented by Li et al. [99]. First, semantic similarity is derived from a lexical knowledge-base as well as a corpus. Then, the proposed method incorporates the impact of word order on sentence meaning. The similarity derived from the word order is used to measure the impact of individual different words as well as the word pairs used in a different order.

In the Semantic Text Similarity (STS) task [77], similarity between small units of text is determined by combining semantic and syntactic information. Two necessary functions were proposed: a function that incorporates string similarity and semantic word similarity, as well as an optional function that computes common-word order similarity. Using a combination of semantic metrics and syntactic processing, Islam et al. [77] achieved a very good Pearson correlation coefficient over 30 sentence pairs of a data set. The results were better than the results obtained by Li et al. [99].

Aggarwal et al. [3] used a hybrid method that combines corpus-based semantic relatedness measures over the whole sentence along with the knowledge-based semantic similarity scores that were obtained for the words that belong to the same syntactic role labels in both sentences. All the scores were used as features for building machine learning models, such as linear regression and bagging models. A resulting model score gives the degree of similarity between sentences. This approach yielded an important improvement in the evaluation of the degree of similarity between sentences by combining the knowledge-based similarity measures and the corpus-based relatedness measures over the use of only corpus-based measures.

Furthermore, a promising correlation between manual and automatic similarity results were achieved in [22] by combining two similarity modules. The first module calculates the similarity between sentences using N-gram based similarity, and the second module calculates the similarity between concepts in the two sentences using a concept similarity measure and WordNet.

Also, the similarity detection system called UKP, proposed in [11], achieved a relatively good correlation score with a simple log-linear regression model based on training data, while combining many text similarity measures. These measures were string similarity, semantic similarity, text expansion mechanisms and measures related to structure and style. The best UKP models consisted of a log-linear combination of about 20 features, out of the possible 300 features tested.

8.3 Feature Pool

The initial pool of features that comprise the hybrid approach that is developed in this study have been discussed in detail in Chapter 7. These similarity measures are summarized below.

- Distance and term-based features
  - Overlap: It returns the number of words in common between two sets. Stop words are ignored.
  - JaroWinkler: Similarity scores are computed in the range of 0 to 1. The higher the score is, the more similar are the text strings.
  - Token_Edit: It computes the number of operations that is required to transform a text t1 to a text t2. A constant cost is fixed for the edit operations such as match, insert, substitute and delete.
  - LCS: It returns the length of the longest common sequence of characters that appear left-to-right, but not necessarily in a contiguous block.
  - Cosine: Texts are transformed into vectors having for dimensions in the euclidean space, the words in the texts. The frequency of each word corresponds to the values in the dimension.
  - ROUGEW: It is a weighted LCS-based statistics that relies on consecutive longest common subsequences to compare two texts t1 and its summary, t2.
  - ROUGES: Also called skip-bigram-based co-occurrence statistics, is used to estimate the similarity between two translations of the same texts.
  - ROUGEN: It uses N-Gram co-occurrences to compute the similarity between a reference and candidate translations.
  - Jaccard: Document similarity is measured as a size of the intersection of the two texts dived by the size of their union.

  - QGram : It is the sum of the absolute differences between N-Gram vectors of two strings. We use word N-Gram of size 2.

- Knowledge-based algorithm features
  These features rely on word to word similarity methods as intermediary steps to computing sentence to sentence similarity. A combination of the word-pair similarity values represents the score for a given pair of sentences. These measures include:

  - LIN uses WordNet LIN method for word-to-word similarity.
  - RES uses WordNet RES method for word-to-word similarity .
  - PATH uses WordNet PATH method for word-to-word similarity.

- Corpus-based features

  - LSA: LSA based word-to-word similarity. We used the TASA corpus LSA model [167].

8.3.1 Feature Selection

One step in developing the hybrid methodology is deciding the features to use in the machine learning phase. Feature selection is not a single phase in the methodology, but rather develops as the machine learning task uses various feature pools. The selection process is informed by a variety of tools. One simple technique is correlation. While such a simple elimination technique can already help us reduce the feature pool, it is not enough to make an accurate decision about the degree of importance of these features. Software packages such as Boruta [91] and machine learning utilities such as the *lm* function [58] implemented in the R language can help make such a decision. These tools are described in the next sections. The final pool is decided when different subpools are tried in the machine learning phase.

8.3.2 Feature Selection using Correlation

Feature analysis helps remove redundant features by dealing with collinearity between attributes that are highly correlated. Features can be removed or ranked by their importance using methods that report on the relationship between attributes based on specific criteria such as correlation values and Z-score. The importance of features can also be determined by building models that assess their contribution to the performance of the best predictive model. Figure 8.3 shows the correlation scores between attributes we deem important for the learning model we intend to build. A quick peek can determine that LCS and Token_Edit are highly correlated

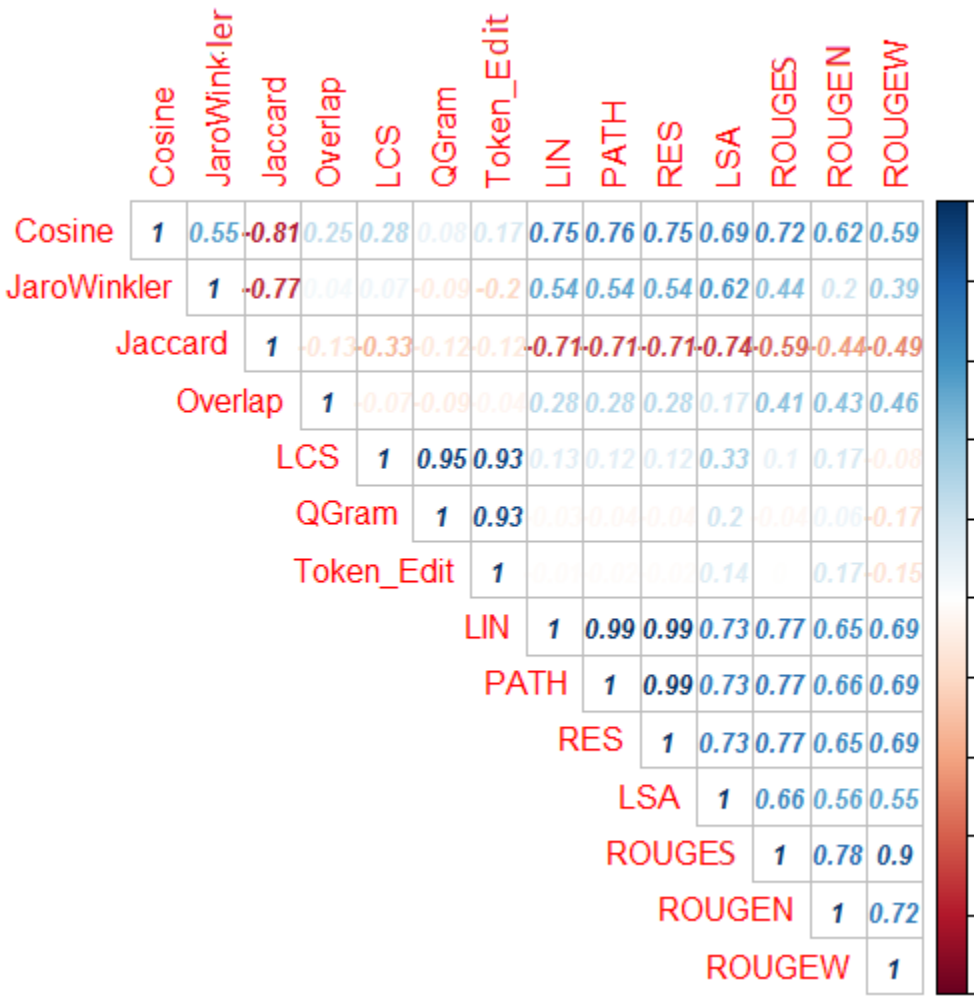

Figure 8.1: Pair-wise correlation values for feature relationship analysis

and the model should therefore incorporate only one of these features. Other highly correlated features are LIN, PATH and RES, and ROUGES and ROUGEW.

8.3.3 Feature Selection with Boruta

The Boruta package uses a Random Forest algorithm to build decision trees based on a random selection of subsets of attributes. Different bootstrap samples of the training set are used to build models that lead to the best distribution of data between nodes of the decision tree. A preliminarily step consists of calculating some "shadow" attributes whose values are obtained by random permutation of the original attributes. The importance of each attribute is estimated based on its contribution to the predicted values of the instances in the training set. Z-score values obtained by dividing the contribution score by the standard deviation value are used to rank the attributes by their importance. The attributes deemed important are those having better

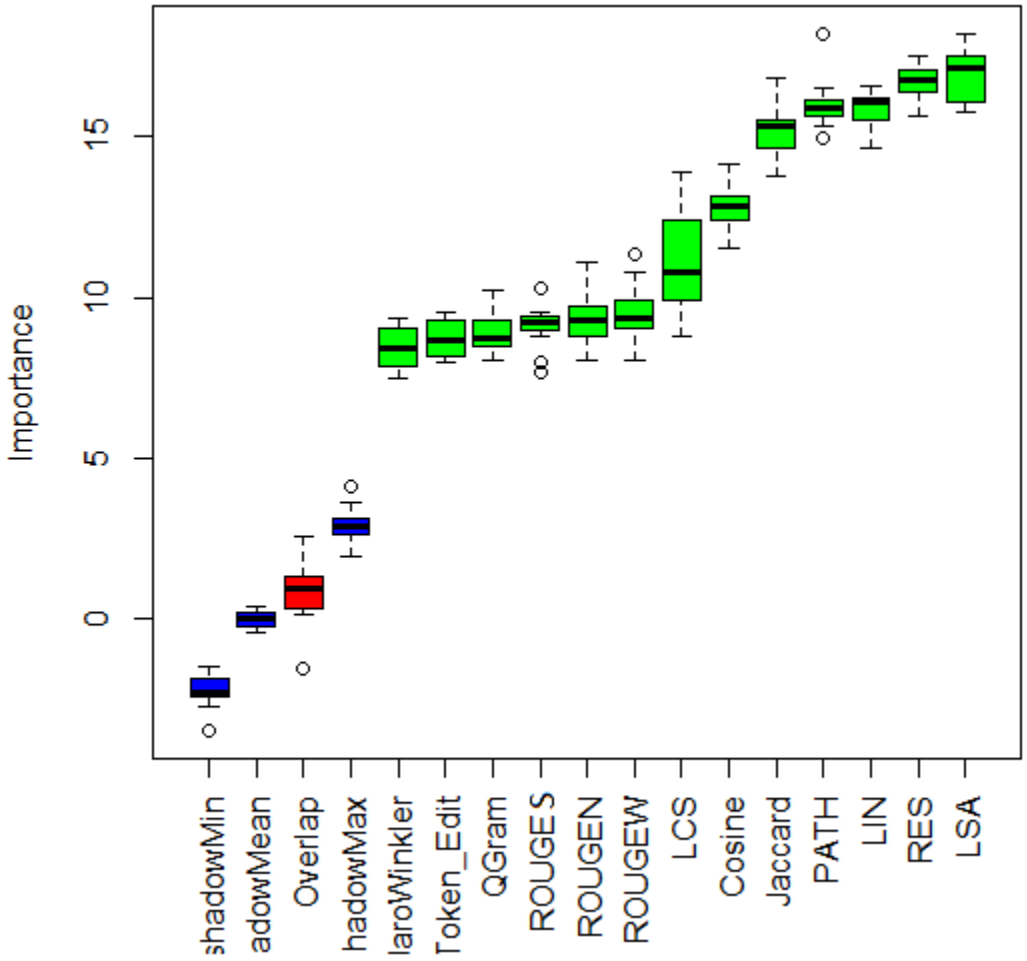

Figure 8.2: Feature importance using Boruta. The colours "green", "red", "blue" indicate "Confirmed", "Rejected" and "Shadow" attributes, respectively.

Z scores than the maximum Z score among shadow attributes.

Figure 8.2 shows feature selection using the Boruta package on *Dataset1*. We can see that one feature, Overlap, has been rejected and 13 features have been confirmed. The most important feature is LSA and the least important of the selected features is JaroWinkler. Table 8.1 presents the statistics about the importance of the features. Based on the correlation scores between the features, one may expect that PATH, LIN and RES would be collapsed into one feature to reduce collinearity. The same thing could be said of the two ROUGE features that seem quite similar. The Boruta technique produces importance scores for the different features. So, a combination of these two analyses would suggest that ROUGEW would be preferred over ROUGES and RES would be preferred over both PATH and LIN.

Features	Mean Importance	Median Importance	Minimum Importance	Maximum Importance	Decision
Cosine	12.828571	12.8577007	11.528687	14.178081	Confirmed
JaroWinkler	8.450618	8.4285305	7.484721	9.362891	Confirmed
Jaccard	15.184641	15.3142992	13.795474	16.850114	Confirmed
Overlap	0.825512	0.9813544	-1.512515	2.600881	Rejected
LCS	11.080594	10.8034608	8.783717	13.910828	Confirmed
QGram	8.907945	8.7453438	8.030900	10.213252	Confirmed
Token_Edit	8.730280	8.7057877	8.005638	9.555609	Confirmed
LIN	15.876769	16.1118099	14.682388	16.589173	Confirmed
PATH	16.023402	15.9107779	14.946889	18.193780	Confirmed
RES	16.707691	16.7601900	15.625752	17.492634	Confirmed
LSA	16.882704	17.1217413	15.792107	18.193728	Confirmed
ROUGES	9.095186	9.2124287	7.695301	10.309025	Confirmed
ROUGEN	9.345326	9.2953577	8.033144	11.130090	Confirmed
ROUGEW	9.533045	9.3567052	8.032066	11.364687	Confirmed

Table 8.1: Feature selection statistics using Boruta showing Importance values and Confirmed and Rejected Features

8.3.4 Feature Selection with *lm*

We use the *lm* function in R to build linear models with the original feature pool. The results suggest that four attributes, Cosine, PATH, LSA and ROUGEN, are really important for building the model. Table 8.2 presents the values of a model's coefficients as well as the features' importance based on their p-values (in the utmost right column).

8.3.5 Final Feature Pool

Three techniques have been used to suggest features that do not provide valuable information in the context of the other features. These different feature pools have been used with the machine learning techniques described next and a final feature pool is decided upon. Correlation has suggested LCS, QGram, and Token_Edit could be reduced. A similar suggestion could be made for the features LIN, PATH, and RES. ROUGES and ROUGEW also highly correlated. *Boruta* ranks Overlap as not important. *Lm* provides a reduced feature set: Cosine, PATH, LSA, and ROUGEN. However, when using these four features with the machine learning techniques that are discussed below, the models produced did not show improved performance. As suggested by *Boruta*, Overlap is removed from the feature pool described above in the final evaluation that is reported below.

Features	Estimate	Std. Error	t value	Pr(>|t|)	Significance
(Intercept)	-9.762e-02	3.507e-01	-0.278	0.780767	
Cosine	9.065e-01	2.048e-01	4.427	9.80e-06	***
JaroWinkler	-3.974e-01	2.665e-01	-1.491	0.136008	
Jaccard	1.553e-01	2.752e-01	0.564	0.572523	
LCS	4.886e-04	9.273e-04	0.527	0.598243	
QGram	-1.370e-04	8.548e-04	-0.160	0.872694	
Token_Edit	9.295e-05	8.244e-04	0.113	0.910235	
LIN	-6.715e-01	9.956e-01	-0.674	0.500058	
PATH	3.804e+00	1.085e+00	3.505	0.000461	***
RES	-1.236e+00	1.160e+00	-1.066	0.286499	
LSA	-5.105e-01	1.030e-01	-4.956	7.45e-07	***
ROUGES	6.655e-01	4.976e-01	1.337	0.181166	
ROUGEN	-5.332e-01	1.843e-01	-2.893	0.003837	**
ROUGEW	1.086e+00	7.124e-01	1.524	0.127609	

Table 8.2: Feature selection statistics provided by *lm*. Residual standard error: 0.6618 on 4431 degrees of freedom. Multiple R-squared: 0.08456, Adjusted R-squared: 0.08187 . F-statistic: 31.48 on 13 and 4431 degrees of freedom, p-value: < 2.2e-16. Significance codes: '***': p-value < 0.001; '**': p-value ≥ 0.001 and < 0.01; all others, p-value > 0.1

8.4 Regression Models

The relationship between two variables can be asserted using regression models. By building these models, the investigator tries to figure out to what extent the increase or decrease of one variable affects the other. Along these lines, the similarity between two text units can be modeled as a means to find out to what extent a given similarity value can help determine the degree of similarity between two text units. For a given dataset comprising pairs of sentences deemed similar or not by annotators to a certain level of confidence, ranging from 0 (the annotator is confident that there is no similarity in content between the two sentences), to 1 (low confidence that similarity in content exists), up to 5 (annotator is confident that there is strong similarity between the two sentences), can we build a regression model whose output can accurately predict the degree of similarity for any pair of sentences? We used three types of regression models to test this hypothesis, namely multi-linear regression, SVM Linear Regression, and Ordinal Regression.

8.4.1 Linear Regression

To test our hypothesis, we first opt for a multi-linear regression model by combining the scores of many similarity measures between pairs of sentences. Let y_i be the degree of similarity between a sentence pair in the data set; let $x_{i1}, \ldots, x_{ip}$ be the values for p similarity metrics

applied to this pair. The linear model over the whole dataset of n pairs of sentences

$$y_i, x_{i1}, \ldots, x_{ip}, \quad i = 1, \ldots, n \tag{8.1}$$

is defined by:

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \varepsilon_i = \beta_0 + X_i^T \beta + \varepsilon_i, \quad i = 1, \ldots, n \tag{8.2}$$

T denotes the transpose, and $X_i^T \beta$ is the inner product between the vectors X_i and β. This equation can be stacked into:

$$y = \beta_0 + X\beta + \varepsilon \tag{8.3}$$

where: $y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$, $X = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}$, $\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}$, and $\varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_p \end{pmatrix}$

The best model is the one that minimizes the residual sum of square error $\sum_{i=1}^{n} \varepsilon_i$.

8.4.2 Support Vector (Linear) Regression

The general idea of Support Vector Regression using a linear kernel is explained in the following.

First, let's begin with Support Vector Machines. Suppose we have a set of training instances, D, where each training instance is represented by a d-dimensional vector $\vec{x_i}$ representing the datum to be classified, and a number, y_i, representing the class of the datum. So, $D = \{(\vec{x_1}, y_1), \ldots (\vec{x_l}, y_l)\} \subset \mathbb{R}^d \times \mathbb{R}$. Our goal is to find the function $f(\vec{x})$ which has at most ε deviation from a target value y_i, for all training instances $(\vec{x_i}, y_i)$. In [155], such a function is shown to take the form:

$$f(\vec{x}) = \vec{w} \cdot \vec{x} + b, \quad with \quad w \in \mathbb{R}^d, \quad b \in \mathbb{R} \tag{8.4}$$

where $\vec{w}$ is the normal vector to the hyperplane defined by $\vec{w} \cdot \vec{x} - b = 0$.

The objective is to minimize the Euclidean norm $\|\vec{w}\|_2$ so that no deviation larger than ε is allowed. This can be captured as a convex optimization problem:

- minimize $\frac{1}{2}\|\vec{w}\|^2$

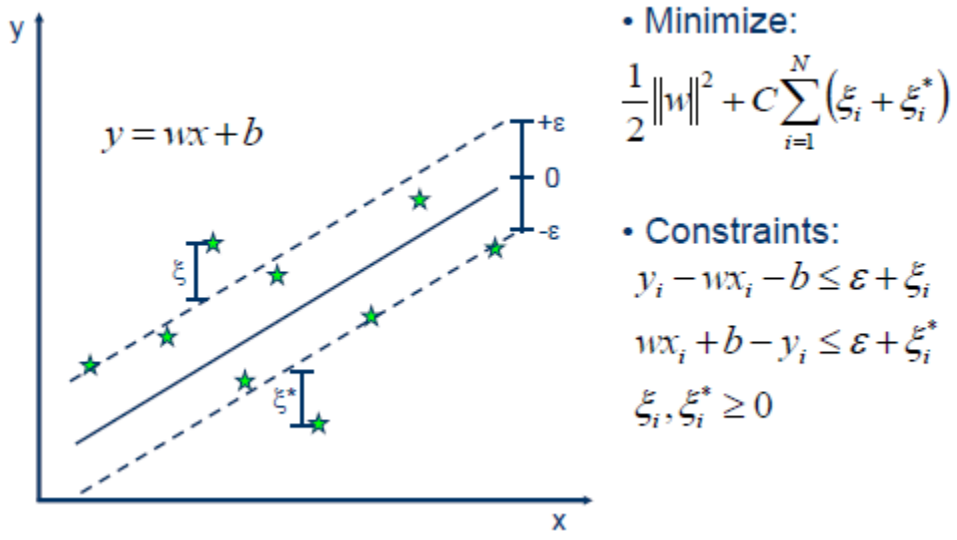

Figure 8.3: Support Vector Machine linear regression (two dimensional case)
Source: http://www.saedsayad.com/support_vector_machine_reg.htm

- subject to:

$$\left\{ \begin{array}{l} y_i - \vec{w} \cdot \vec{x_i} - b \le \varepsilon, \quad \forall i \\ \vec{w} \cdot \vec{x_i} + b - y_i \le \varepsilon, \quad \forall i \end{array} \right\} \tag{8.5}$$

In the case of Support Vector (Linear) Regression models two slack variables (for each instance) are introduced to cope with the infeasible constraints of the optimization equation. Therefore the formulation of the regression problem results in the following optimization problem:

- minimize $\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$
- subject to:

$$\left\{ \begin{array}{c} y_i - \vec{w} \cdot \vec{x_i} - b \le \varepsilon + \xi_i, \quad \forall i \\ \vec{w} \cdot \vec{x_i} + b \le \varepsilon + \xi_i^*, \quad \forall i \\ \xi_i, \xi_i^* \ge 0, \quad \forall i \end{array} \right\} \tag{8.6}$$

The constant $C > 0$ is the trade-off between f and the amount up to which deviations larger than ε are tolerated. This condition introduces a loss function $|\xi|_\varepsilon$ depicted graphically in Figure 8.3.

In resolving this equation to determine $\vec{w}$ and b, $\vec{w}$ can be described as a linear combination of training patterns $\vec{x_i}$.

$$\vec{w} = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\vec{x_i} \tag{8.7}$$

and therefore

$$f(\vec{x}) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) \vec{x_i} \cdot \vec{x_i} \tag{8.8}$$

with α_i and $\alpha_i^* \geq 0$.

$$b = y_i - \vec{w} \cdot \vec{x_i} - \varepsilon \quad \text{for} \quad \alpha_i \in (0, C) \quad \text{and} \quad b = y_i - \vec{w} \cdot \vec{x_i} + \varepsilon \quad \text{for} \quad \alpha_i^* \in (0, C) \tag{8.9}$$

8.4.3 Ordinal Regression

Ordinal regression is used to predict the dependent variable with independent variables that take values from multiple ordered categories. The aim is to facilitate the interaction of dependent variables (having multiple ordered levels) with one or more independent variables. It is a special case of linear regression by which models are built to help with the observation of the natural order in categories. In the case of our dataset, we consider the natural order of the rating to be 0, 1, 2, 3, 4 and 5 in the original form of the data. To fit the model, regression coefficients are estimated to predict the probability of the outcome of interest such that:

$$\log \frac{prob(event)}{1 - prob(event)} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_i X_i \tag{8.10}$$

where i goes from 1 to the number of features (see [88]). Each X_i corresponds to the ith data feature. In our case they range from 1 to 14. The β_i's are the model's (learned) coefficients. The dependent variables (outcomes) are the ordered categories (from 0 to 5). The independent variables (predictors) are the X_i's. The values of these variables are being used in combination with the coefficients to predict the outcome values.

8.5 Evaluation Methods

In this study we perform two types of evaluations. The first sort of evaluation is to decide which of the three regression methods performs the best and should be further studied. To do this evaluation, we use the Pearson correlation coefficient to compare human judgment with the output of the prediction models produced by the three regression methods. The details of how this correlation coefficient is calculated is presented in the next section. We expect a good model to be the one that yields a high coefficient, denoting a positive relationship between the predicted values and the expected values.

The second type of evaluation is performed to determine how well the rankings of candidate sentences provided by the best regression method matches the human annotated confi-

dence ratings. In the case of unranked retrieval, the commonly used evaluation measures such as precision, recall and F-measure (see Chapter 4 for details of these measures) are set-based measures and they don't take into account the ranking of the candidate sentences. When we have multi-level confidence judgments as is the case in the present study, an adequate evaluation measure should take into account the total utility of the k ranked candidate sentences. The two measures, Normalized Discounted Cumulative Gain at rank k (*NDCG@k*) and *Precision* at k are described below. It should be noted that these two measures are also used in the evaluation performed in Chapter 9.

8.5.1 Correlation as an evaluation metric

The performance of each system is assessed by computing the Pearson Correlation Coefficient between machine assigned scores and human ratings. This correlation coefficient measures the strength and the direction of the relationship that exists between two variables. In the case of the present study, we want to access how the machine output of the similarity scores matches the values produced by human judgment. Let X be the vector representing the confidence measures produced by human judgment and Y the corresponding output by the system. The Pearson Correlation Coefficient is defined by:

$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{X_i - \bar{X}}{s_X}\right)\left(\frac{Y_i - \bar{Y}}{s_Y}\right) \tag{8.11}$$

where

$$\frac{X_i - \bar{X}}{s_X}, \quad \frac{Y_i - \bar{Y}}{s_Y} \tag{8.12}$$

are the standard scores and

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i, \quad \bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i \tag{8.13}$$

and

$$s_X = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2}, \quad s_Y = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})^2} \tag{8.14}$$

are the sample means, and sample standard deviations, respectively. We can interpret the value of the correlation coefficient as follows:

- Range: $-1 \leq r \leq 1$
- Correlation coefficient is a unit-less index of strength of association between two variables (+ = positive association, − = negative association, 0 = no association)

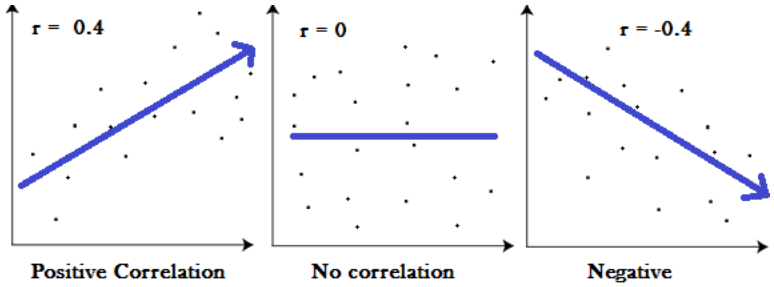

Figure 8.4: Correlation coefficient
Source: http://www.statisticshowto.com/how-to-compute-pearsons-correlation-coefficients/

- Measures the linear relationship between X and Y
- Can test for significant association by testing whether the correlation is zero.
- High correlation: 0.5 to 1.0 or -0.5 to -1.0
- Medium correlation: 0.3 to .5 or -0.3 to -0.5
- Low correlation: 0.1 to 0.3 or -0.1 to -0.3

8.5.2 Normalized Discounted Cumulative Gain at rank k (*NDCG@k*)

The *NDCG* evaluation measure is computed to reflect the ideal position of the very relevant, the marginally relevant as well as the non-relevant sentences in the ranking. Before providing a more formal definition, let's first look at an example. Assume:

- a dataset of three relevance levels such as: $r = 1$ (non-relevant), $r = 2$ (marginally relevant), $r = 3$ (very relevant)
- 9 sentences rated "3" in the article
- 1 sentence rated "2" in the article
- The ideal ranking of the first 10 sentences would assume that the first 9 sentences would be rated "3" and the 10^{th} would be rated "2". Therefore:
- The Ideal Cumulative Gain (*IDCG*) is: $3 + 3 + 3 + \ldots + 3 + 2$
- The Discounted Cumulative Gain (*DCG*) is: $3+3/\log 2+3/\log 3+\ldots+3/\log 9+2/\log 10$

If the output of a given ranking model is for instance 3 2 1 1 3 3 1 1 2 1, the *DCG* at rank 10, $DCG@10 = 3 + 2/\log 2 + 1/\log 3 + \ldots + 1/\log 10$ and the *Normalised DCG*@10 is:

$$NDCG@10 = \frac{DCG@10}{IDCG@10} \tag{8.15}$$

The Normalized Discounted Cumulative Gain (*NDCG*) principle can be summarized as follows:

- It is applicable to multi-level judgments in a scale of $[1, r]$, $r > 2$
- It measures the total utility of the top k sentences to a user
- The utility of a lower ranked sentence is discounted
- The score is normalized to assure comparability across queries

$$DCG@k = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{\log_2 i} \tag{8.16}$$

$$IDCG@k = \sum_{i=1}^{k} rel_i \tag{8.17}$$

rel_1 is the graded relevance of the sentence at position 1 and rel_i is the graded relevance of the sentence at position i.

$$NDCG@k = \frac{DCG@k}{IDCG@k} \tag{8.18}$$

8.5.3 Precision at *k*

- **Precision at *k***:
  In the the computation of the output of some search operations, we are interested in the proportion of good results among the first k answers (say the first 3 sentences).
- Evaluating the system based on the precision at a fixed level may be enough to assure a reliability of the system as there is no need for an estimate of the whole set of the relevant sentences.

8.6 Experiments

The machine learning techniques that are used in this study are supervised methods. Supervised methods require human annotated data for learning. Preparation of the required data has been

discussed previously in Section 3.5. The data consists of 22 citation sentences referencing 22 papers. On average the number of sentences per paper is approximately 194. Treating each citation-paper pair in isolation might not provide enough data to build relevant learning models to successfully retrieve the target cited sentences. In order to build a consistent dataset, we deemed it necessary to combine citation sentence-target sentence instances from individual papers. We then group the pairs of sentences having the same rating into the same category irrespective of their provenance. In the experiments presented below, we use different sizes of datasets to build learned models to predict the human rating of candidate linkage sentences.

We conducted experiments that can be divided into two categories:

***Category1*:** experiments that use the original dataset comprising pairs of sentences built by matching citation sentences with all the sentences in the cited papers.

***Category2*:** experiments that use a reduced dataset comprising pairs of sentences built by matching the citation with the Method sentences derived from the cited papers. Using our rhetorical classification method that was presented in Chapter 6, the 0-category in the original dataset is reduced to contain only Method sentences.

In *Category2* experiments, we have made the assumption that citations of a method mention will likely be linked to similar method sentences in the cited paper. For instance in paper [59], the citation sentence *Quantitative RT-PCR was carried out using a Rotor-Gene 2000 centrifugal real-time cycler (Corbett Research) using the Platinum SYBR Green qPCR SuperMix-UDG (Invitrogen) as described previously [62].*[1] contains the method mention *Quantitative RT-PCR* and is talking about how it is used in the cited paper [71]. The reduction of the possible candidates for citation linkage is done with an automatic method, so it could be a preprocessing step for the citation linkage method when deployed in a real situation. Reducing the set of candidate sentences has a further consequence: the set of negative instances (those rated as 0 in the annotated corpus) has been decreased in size lessening the problem of unbalanced data. The original dataset ratio of 0-rated to non-0-rated sentences is 20:1, whereas for the reduced dataset it is 6:1.

Another aspect of the data is the score given each sentence by the annotator. The annotator was asked to assign a score of 1 to 5 for those sentences that the annotator considered as candidates for citation linkage. The score indicates the level of confidence the annotator had making the decision (1 for low confidence up to 5 for high confidence). One other modification of the data was made to reduce any possible arbitrariness in the scores given by the annotator. For each experimental category, we use five datasets, one comprising the six original rating scores

[1] The cited paper [62] is the reference in the original citation.

0	1	2	3	4	5	Total
4049	44	42	42	48	33	4258

(a) Dataset1-Category1

0	1	2	3	4	5	Total
1244	44	42	42	48	33	1453

(b) Dataset1-Category2

Table 8.3: Dataset1 (0)-(1)-(2)-(3)-(4)-(5)

0	1	2	3	4-5	Total
4049	44	42	42	81	4258

(a) Dataset2-Category1

0	1	2	3	4-5	Total
1244	44	42	42	81	1453

(b) Dataset2-Category2

Table 8.4: Dataset2 (0)-(1)-(2)-(3)-(4-5)

0	1-2	3	4-5	Total
4049	86	42	81	4258

(a) Dataset3-Category1

0	1-2	3	4-5	Total
1244	86	42	81	1453

(b) Dataset3-Category2

Table 8.5: Dataset3 (0)-(1-2)-(3)-(4-5).

0	1-2-3	4-5	Total
4049	128	81	4258

(a) Dataset4-Category1

0	1-2-3	4-5	Total
1244	128	81	1453

(b) Dataset4-Category2

Table 8.6: Dataset4 (0)-(1-2-3)-(4-5)

(0)-(1)-(2)-(3)-(4)-(5), and four others built by combining some of the rating categories in the following patterns: (0)-(1)-(2)-(3)-(4-5), (0)-(1-2)-(3)-(4-5), (0)-(1-2-3)-(4-5), and (0)-(1-2-3-4-5). These combinations have been chosen for the following reasons: the first combination combines the two highest confidence scores, the second also combines the scores of those sentences deemed as candidates with the two lowest confidence scores, the third combination groups the middle confidence level providing two confidence classes (low and high) for the candidate sentences, and the fourth combines all candidate sentences into one class giving a binary classification problem. Another consequence of this pooling of instances into new rating score groupings is to increase the number of instances in some of the groups. Tables 8.3–8.7 provide the number of instances in each class in each of these datasets.

We use three sets of features to build the learning models:

- Distance-based features only (DB)
- Distance-based and knowledge-based features (DB + KB)

Negative (0)	Positive (1-5)	Total
4049	209	4258

(a) Dataset5-Category1

Negative (0)	Positive (1-5)	Total
1244	209	1453

(b) Dataset5-Category2

Table 8.7: Dataset5 (0)-(1-2-3-4-5)

- Distance-based, knowledge-based, and corpus-based features. (DB + KB + CB)

As we said previously, we use different datasets formed using the individual rating categories or a combination of two or more categories. We then built various regression models to evaluate the degree of similarity between the citation sentences and candidate sentences in the cited papers. The experimental results for the various combinations of learned model, feature systems, datasets, and dataset categories are presented in Section 8.7.

8.7 Results and Discussion

We conduct experiments using two types of linear regression models in which the response variables are assumed to be continuous in the range [0, 5]. We would expect the predicted values to be in such a range as well and that the correlation between the gold standard (human-rated) values and the predicted values to be close to 1.

In a third set of experiments, we consider the response variables to be nominal and ordered. We assume that the distance between the variable denoting the nominal and ordered degree of confidence given by the annotator when deciding whether a citation sentence links to a sentence in the cited paper to be on a gradual scale. In our case, the rating scores are nominal and ordered in a 0 (no linkage), 1 (low confidence of linkage), 2, 3, 4, 5 (high confidence of linkage) pattern.

The results of applying SVM Linear Regression and Linear Regression on the two dataset categories are presented in Tables 8.8 and 8.9. The results presented are the Pearson Correlation Coefficient values between the expected and predicted scores. The values given in Table 8.8 show that the best results are obtained with Linear Regression when distance-based, knowledge-based, and corpus-based features are used. Also, in this column dataset combinations with the patterns (0)-(1)-(2)-(3)-(4)-(5), (0)-(1)-(2)-(3)-(4-5), (0)-(1-2)-(3)-(4-5), (0)-(1-2-3)-(4-5) show similar performance.

The maximum value of 0.2762 (low correlation) is obtained with a multi-linear regression model. The low performance of the models may be due to:

- The unbalanced nature of the data (the number of 0-rated instances is twenty times the

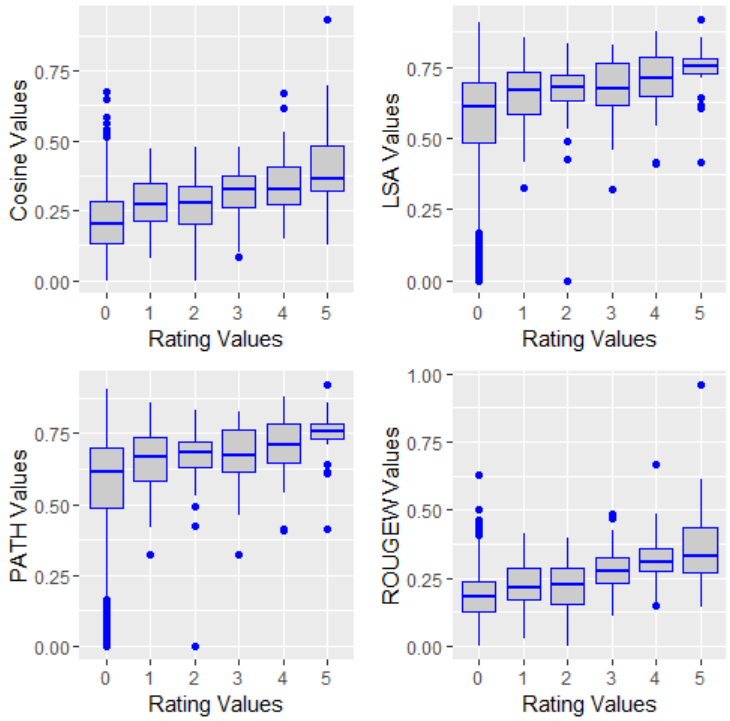

Figure 8.5: Four examples of the distribution of features per Rating based on the summary statistics: minimum, first quartile, median, third quartile, and maximum

number of instances rated 1–5). We tried stratified remove folds, but it didn't add any improvement to the models.

- Too many feature ranges overlap. The range for the attribute values are not easily separable between the response variables in many cases (see Figure 8.5).

In Table 8.9, we notice an improvement to the model as the best correlation value is 0.4217 (medium correlation). The improvement can possibly be explained by the reduction of negative instances when only Method sentences are taken into account. This further indicates that the unbalanced nature of the original dataset can affect the accuracy of the linkage models. We can also notice that the choice of the learning algorithm is very important. There is a major difference between the higher values obtained with the SVM Linear Regression models and the ones with Linear Regression models. It is therefore important to try multiple algorithms and choose the one that best fits the underlying data. Also, the combination of features from different similarity measures can help improve the systems significantly. However due to the resource intensive nature of the computation, it is important to choose features that optimize the systems while minimizing the learning cost.

	DB		**DB + KB**		**DB + KB + CB**	
Data	SVMReg (r)	LinearReg (r)	SVMReg (r)	LinearReg (r)	SVMReg (r)	LinearReg (r)
0-1-2-3-4-5	0.2016	0.27	0.1862	0.2652	0.2183	0.2759
0-1-2-3-45	0.2136	0.2658	0.1852	0.2614	0.2018	0.2737
0-12-3-45	0.1926	0.2667	0.1816	0.2651	0.1613	**0.2762**
0-123-45	0.1521	0.2443	0.1968	0.2426	0.1837	0.2711
0-12345	0.1521	0.2443	0.1968	0.2426	0.1848	0.2442

Table 8.8: Evaluation (Pearson Correlation Coefficient) using SVM Linear Regression and Linear Regression with the original dataset comprising pairs of sentences built by matching citation sentences with all the sentences in the cited papers (*Category1*).

	DB		**DB + KB**		**DB + KB + CB**	
Data	SVMReg (r)	LinearReg (r)	SVMReg (r)	LinearReg (r)	SVMReg (r)	LinearReg (r)
0-1-2-3-4-5	0.3626	0.4209	0.3659	0.4206	0.3553	**0.4217**
0-1-2-3-45	0.3673	0.4167	0.339	0.4162	0.3581	0.4184
0-12-3-45	0.3325	0.4143	0.3269	0.4143	0.3483	0.4185
0-123-45	0.3478	0.4114	0.3491	0.4105	0.3439	0.4139
0-12345	0.3605	0.3819	0.3029	0.3788	0.3055	0.3778

Table 8.9: Evaluation (Pearson Correlation Coefficient) using SVM Linear Regression and Linear Regression with the reduced dataset comprising pairs of sentences built by matching the citation with the Method sentences derived from the cited papers (*Category2*).

	DB	**DB + KB**	**DB + KB + CB**
Data	Ordinal Reg	Ordinal Reg	Ordinal Reg
0-1-2-3-4-5	0.174	**0.212**	**0.212**
0-1-2-3-45	0.1567	0.194	0.192
0-12-3-45	0.131	0.161	0.161
0-123-45	0.0829	0.102	0.102

Table 8.10: Evaluation (Pearson Correlation Coefficient) using Ordinal Regression with the original dataset comprising pairs of sentences built by matching citation sentences with all the sentences in the cited papers (*Category1*).

Table 8.10 shows results for the ordinal regression models. The Pearson correlation coefficient scores between the expected values and the predicted values are reported. The scores are essentially the same for the feature sets containing the distance-based and knowledge-based features and all features for all of the datasets. The highest score is **0.212** with the original data and the two feature sets just mentioned. The reason for the poor performance might also be due to the unbalanced nature of the dataset and the overlapping ranges of the feature values.

8.8 Ranking with the Linear Regression Model

Despite the limited performance of all the models, the Linear Regression model using all of the features trained on the data with the original six rating scores performed the best. We therefore choose the Linear Regression model as the ranking algorithm for the linkage operation and the data in its original rating format as a machine learning task.

In order to rank the sentences in a target paper, we performed a Leave-One-Out (LOO) cross validation by building a learning model with 21 papers. The resulting model is then used to predict the scores for the (citation, sentence) pairs in the target paper. These scores are then ranked in descending order.

To determine the effectiveness of the result, we compare the ranking with the relevant candidate sentences. Two evaluation metrics are used: (1) the Normalized Discounted Cumulative Gain at rank k score (*NDCG@k*) [80] is computed by taking into account the position of the sentences in the ranking as well as their relevance values as provided by human judgment; (2) the *Precision* at k score is computed as the ratio of the top k retrieved sentences irrespective of their position in the ranking. Table 8.11 shows the results per paper for the Linear Regression model when using all sentences in an article as candidate referenced sentences. Table 8.12 shows the results per paper for the Linear Regression model when using only those sentences belonging to the Method rhetorical category are used as candidate referenced sentences.

Tables 8.13 and 8.14 show six summary statistics namely: minimum, maximum, 1st quartile, median, mean, 3rd quartile, and maximum values over all the papers. The average mean *NDCG@k* is 0.3009 when using all sentences in an article as candidate referenced sentences and 0.4098 when using only those sentences belonging to the Method rhetorical category as candidate referenced sentences. Similarly, the average mean *Precision@k* is 0.2949 when using all sentences in an article as candidate referenced sentences and 0.4225 when using only those sentences belonging to the Method rhetorical category as candidate referenced sentences.

8.8.1 Example of Ranking Output

Table 8.15 shows the top six sentences from Paper 18 (full article) as ranked by the linear regression model. We can notice that three sentences out of six are among the top human-ranked sentences. When we compare this result with the original annotation 8.16, we can notice that most of the higher rated sentences (rated 5 and 3), are retrieved at the right positions as expected. However, the lower rated sentences are not retrieved in the top ranked positions. This implies that, we will need some extensive features to accurately discriminate between lower rated sentences and non-relevant sentences.

Paper Number	Number of sentences	Precision number (k)	*Precision@k*	*NDCG@k*
1	126	19	0.4737	0.4612
2	166	16	0.5	0.3811
3	150	12	0.5833	0.5257
4	162	3	0.3333	0.4693
5	194	18	0.2778	0.3169
6	185	3	0	0
7	169	4	0.25	0.1681
8	291	7	0.2857	0.3665
9	233	3	0	0
10	224	8	0.625	0.7031
11	315	31	0.2258	0.2812
12	89	5	0.4	0.4704
13	239	2	0	0
14	236	12	0.25	0.2136
15	249	14	0.2143	0.1679
16	189	8	0.125	0.0843
17	112	15	0.4	0.4968
18	143	6	0.5	0.4593
19	185	4	0.25	0.1952
20	165	3	0.3333	0.2961
21	266	3	0	0
22	170	13	0.4615	0.5633

Table 8.11: Evaluation of the Linear Regression model using all features for each left out paper with all sentences as possible candidates

It is interesting to note that 18 out of 22 papers have at least one sentence retrieved at the top k position (k being the number of relevant sentences per paper) and 10 papers have *NDCG@k* scores greater than 47%. These results are very promising and can be improved with better ranking algorithms as we will show in Chapter 9. The overall statistics of the ranked candidate sentences in the top k are presented in Table 8.17.

Table 8.18 presents the predicted number of sentences per category per paper. Most of the features used to build the learning models rely on word overlap techniques or some of their variants in the computation of the similarity scores between citation sentences and their targets. We can notice that fewer sentences human-rated as 5 and 4 have been rated as a 0 compared to sentences human-rated as 3, 2 and 1. However, the fact that 51% of the sentences human-rated as 5 and 46% of the sentences human-rated as 4 are found, suggests that there is still room for improvement. We can assume given the annotator's comments that higher rated candidate sentences share common surface level information with citation sentences. We might

Paper Number	Number of sentences	Precision number (k)	*Precision@k*	*NDCG@k*
1	59	19	0.6316	0.678
2	65	16	0.625	0.5407
3	59	12	0.6667	0.611
4	24	3	1	1
5	60	18	0.5556	0.5744
6	44	3	0	0
7	53	4	0.5	0.3633
8	92	7	0.4286	0.4644
9	97	3	0	0
10	65	8	0.75	0.6381
11	93	31	0.3548	0.4397
12	32	5	0.4	0.3452
13	47	2	0	0
14	103	12	0.3333	0.3149
15	40	14	0.5	0.4092
16	99	8	0.25	0.2063
17	53	15	0.6	0.6892
18	65	6	0.5	0.429
19	62	4	0.25	0.1952
20	39	3	0.3333	0.4693
21	119	3	0	0
22	83	13	0.6154	0.6482

Table 8.12: Evaluation of the Linear Regression model using all features for each left out paper with Method sentences as possible candidates

Metrics	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Precision@k	0.0000	0.2172	0.2817	0.2949	0.4462	0.6250
NDCG@k	0.0000	0.1679	0.3065	0.3009	0.4673	0.7031

Table 8.13: Statistics for *Precision@k* and *NDCG@k* over all the papers

Metrics	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Precision@k	0.0000	0.2708	0.4643	0.4225	0.6115	1.0000
NDCG@k	0.0000	0.2334	0.4344	0.4098	0.6018	1.0000

Table 8.14: Statistics for *Precision@k* and *NDCG@k* over all the papers reduced to Method sentences

expect the linkage operation to be straightforward in many of such cases. However it doesn't generalize to all the papers as papers 6, 13, and 21 didn't yield the expected results for their higher ranked sentences.

Citation Sentence		
Quantitative RT-PCR was carried out using a Rotor-Gene 2000 centrifugal real-time cycler (Corbett Research) using the Platinum SYBR Green qPCR SuperMix-UDG (Invitrogen) as described previously.		
Ranked Sentences	Rating	Score
Quantitative RT-PCR was carried out using a Rotor-Gene 2000 centrifugal real-time cycler (Corbett Research) using the Platinum SYBR Green qPCR SuperMix-UDG (Invitrogen).	5	2.73
Amplification was performed in 0.1 ml real-time PCR tubes (Corbett Research) placed in the 72-well rotor of the Rotor-Gene instrument.	3	0.66
Each reaction contained: 12.5 μl of the Platinum SYBR Green qPCR SuperMix-UDG 200 nM 300 nM or 400 nM of forward and reverse primers and 5 μl cDNA (1:40 RNA dilution) to a final volume of 25 μl.	3	0.617
The threshold cycle (Ct) values of the Rotor-Gene software version 6.0 (Corbett Research) were exported to qBase version 1.3.5 a free program for the management and automated analysis of qPCR data for further analysis.	0	0.432
Real-time quantitative PCR (qPCR) has become a very powerful tool for gene expression studies.	0	0.39
Total RNA was isolated using the RNeasy Midi Kit (Qiagen) according to manufacturer's instructions.	0	0.36

Table 8.15: Example of Ranking Output provided by the Linear Regression model. This ranking output is for Paper 18. All other sentences in Paper 18 are ranked as non-candidate sentences.

On the one hand, the outcomes may suggest that the sentences human-rated as 1, 2, and 3 may require biochemical knowledge and more background information for the linkage to be effective. On the other hand many higher ranked sentences may also require more preprocessing and domain knowledge for the linkage to be possible. LSA and Wordnet based features such as PATH and LIN do incorporate some kind of synonym information, however this is not domain specific information that human annotators have relied on during the annotation process.

Citation Sentence	
Quantitative RT-PCR was carried out using a Rotor-Gene 2000 centrifugal real-time cycler (Corbett Research) using the Platinum SYBR Green qPCR SuperMix-UDG (Invitrogen) as described previously.	
Candidate Sentences	Rating
Quantitative RT-PCR was carried out using a Rotor-Gene 2000 centrifugal real-time cycler (Corbett Research) using the Platinum SYBR Green qPCR SuperMix-UDG (Invitrogen).	5
Amplification was performed in 0.1 ml real-time PCR tubes (Corbett Research) placed in the 72-well rotor of the Rotor-Gene instrument.	3
Each reaction contained: 12.5 μl of the Platinum SYBR Green qPCR SuperMix-UDG 200 nM 300 nM or 400 nM of forward and reverse primers and 5 μl cDNA (1:40 RNA dilution) to a final volume of 25 μl.	3
The cycling conditions were as follows: 50°C for 2 min initial denaturation at 95°C for 2 min followed by 45 cycles of 15 s at 95°C 30 s at 60°C and 30 s at 72°C (gain set at 8 for SYBR Green).	2
The Rotor-Gene software allows automatic melting curve analysis for all tested samples in a given run.	1
SYBR Green fluorescence of the generated products was continuously monitored throughout the temperature ramp from 60 to 99°C.	1

Table 8.16: Annotation for Paper 18. These sentences are the ones annotated as candidate sentences with the confidence rating scores provided by the annotator. All other sentences in Paper 18 are annotated as non-candidate sentences.

Category	Predicted	Expected	Percent
5	17	33	(51 %)
4	22	48	(46 %)
3	16	42	(38 %)
2	9	42	(21 %)
1	8	44	(18 %)

Table 8.17: Statistics of the ranked candidate sentences over all the papers.

Paper #	Category 1 Pred/Exp	Category 2 Pred/Exp	Category 3 Pred/Exp	Category 4 Pred/Exp	Category 5 Pred/Exp
1	1/4	1/3	4/6	1/4	2/2
2	0/0	3/4	3/4	1/3	1/5
3	0/0	0/0	3/7	1/1	3/4
4	1/1	0/0	0/1	0/1	0/0
5	0/6	0/1	1/1	4/9	0/1
6	0/0	0/0	0/1	0/1	0/1
7	0/1	0/0	0/1	0/0	1/2
8	0/3	0/0	1/3	1/1	0/0
9	0/1	0/1	0/1	0/0	0/0
10	0/0	0/0	0/3	2/2	3/3
11	1/2	1/17	1/4	2/3	3/5
12	1/3	0/0	0/1	1/1	0/0
13	0/1	0/0	0/0	0/1	0/0
14	1/8	0/0	0/0	1/3	1/1
15	2/4	0/2	0/2	1/5	0/1
16	0/2	0/0	1/3	0/2	0/1
17	0/1	1/6	0/1	3/5	2/2
18	0/2	0/1	2/2	0/0	1/1
19	0/2	1/2	0/0	0/0	0/0
20	0/0	0/0	0/0	1/2	0/1
21	0/0	0/0	0/1	0/0	0/2
22	1/3	2/5	0/0	3/4	0/1

Table 8.18: Statistics of predicted and expected sentences per paper. Pred/Exp: Predicted/Expected

8.9 Chapter Summary

In this chapter we present the citation linkage problem as a machine learning task and show that moderate correlations can be achieved when we apply regression models to the datasets. A first set of experiments shows that the unbalanced nature of the dataset could affect the models. This is even confirmed by a second set of experiments in which the data are more condensed to reflect some of the rhetorical categories pertaining to scientific papers in the biomedical domain. We have also shown how we can use linear regression models to determine the degree of similarity between citation sentences and their targets in the cited paper. These results prove that the linkage operation is more achievable with higher rated sentences than the lower rated ones. This is expected as most lower rated sentences have less surface level information in common with the citations and they rely on more background information and domain specific knowledge. In Chapter 9 we will consider how using information retrieval techniques on the datasets can achieve a linkage between citation sentences and cited sentences in a scientific paper.

Chapter 9

Experiments and Results: Citation Linkage as an Information Retrieval Task

9.1 Introduction

In this chapter we hypothesize that citation sentences, when used as queries in a given retrieval model, should likely point to relevant sentences in the cited paper. For this purpose, we use many established retrieval algorithms to check whether the task of citation linkage targeting sentences in a cited paper could be performed by ranking the sentences as would do a search engine. We define the domain of a given search result to be limited to the cited paper. Therefore, the task of citation linkage as a text retrieval operation is confined to a single paper and a query is a citation sentence or part of it, targeting a cited paper. The justification for such an hypothesis can be drawn from the necessary requirements for text retrieval experiments that the linkage task satisfies, such as:

- Information need: Retrieval from a given article, the sentences that match a specific citation sentence.[1]

- Test collections: 22 articles, each containing relevant and non relevant sentences with multi level judgments.

- Evaluation methods: Evaluation measures for ranked retrieval such as *Precision@k* and Normalized Discounted Cumulative Gain at rank k (*NDCG@k*).

[1]Text (or information) retrieval normally speaks of retrieving documents from a collection of documents. Our purpose here is to retrieve sentences from articles. Hence, we will simply substitute "sentence" for "document" and "article" for "collection" whenever the latter words are typically used in the text (or information) retrieval literature.

9.2 Motivation

When a researcher reading a science article wants to establish the link between what the content of a sentence that cites a paper and what it is actually citing, she either has the option to read the whole referenced paper, or try to find out a set of sentences that best fit her needs in that paper. In the process of easing this kind of task, we already previously showed that the former option might just be too time consuming and unnecessary, as the latter option, if available, might just be enough to satisfy the user's specific information need. For a given citation sentence, we believe that there is at least one sentence in the cited paper that corresponds to the searcher's request. We therefore think that treating the linkage task as a text retrieval experience might be a promising method for finding the best match to the citation sentence. Along this line, matching a citation sentence to relevant sentences in the cited paper can be compared to how search engines rank documents based on a user's specific queries. Each article is segmented into sentences and each sentence is compared to the citation sentence using a ranking measure. The position of each candidate sentence is determined by ranking the similarity scores with regard to the citation sentence.

9.2.1 Beyond binary relevance

When information retrieved by search engines is evaluated on a binary relevance basis, each sentence is treated as being either relevant or irrelevant. The commonly used evaluation measures in such a case are: Recall and Precision.

However, relevant sentences are not always evaluated on a binary basis. In some retrieval applications an ideal model should take into account the multi-level relevance of candidate sentences.

When the relevance of sentences in the article can be captured by having more than two classes, new measures are needed to capture these degrees of relevance of each retrieved sentence. The overall score is obtained by combining relevance values and the position of the sentence in a ranked search result. Such measures include the Normalized Discounted Cumulative Gain at rank k ($NDCG@k$) [80]. This measure and $Precision@k$ are defined in Chapter 8.

9.3 Ranking Functions

The design of a ranking function is necessary for building a retrieval model. Such a function should respond to the following criteria: We have a query composed of a sequence of words

Notation	Description
x_w^q	Number of occurrences of w in query q
x_w^s	Number of occurrences of w in sentence s
t_w^s	Normalized version of x_w^s
y_s	Length of sentence s in terms[2] (words)
m	Average sentence length in terms (words)
L	Length of article a in terms (words)
N	Number of sentences in the article
M	Number of unique terms (words) in the article
F_w	Number of occurrences of term w in article: $F_w = \sum_s x_w^s$
N_w	Number of sentences containing w: $N_w = \sum_s I(x_w^s \geq 0)$ where I is an indicator function which equals 1 when its argument is true and 0 otherwise
Z_w	$Z_w = F_w$ or $Z_w = N_w$

Table 9.1: Information Retrieval Parameter Notations

$q = q_1 \dots q_m$ and a set of sentences $s = s_1, \dots, s_n$; each sentence is also a sequence of words. We define a function $f(q, s)$ that computes a score based on the query and the sentences. We target the ideal ranking function to be the one that ranks relevant sentences better than the non-relevant ones. This function measures the likelihood that a sentence s in a cited paper is relevant to the citation c used as query q. In the case of our linkage task, we consider the relevant sentences to be the sentences rated as such by human annotators. We expect that a good retrieval model will rank these sentences above all other sentences in the paper.

9.4 Ranking Models

Retrieval functions satisfy heuristic constraints that are proved to cause good information retrieval [48]. Such retrieval constraints are used to compute ranking scores that incorporate parameters described in Table 9.1.

Depending on the retrieval model being considered, the form of the target function will involve a set of variables that will serve as normalization parameters and/or smoothing constraints. Generally, a retrieving score is of the form:

$$RSV(q, s) = \sum A(x_w^q) h(x_w^s, y_s, Z_w, \theta) \tag{9.1}$$

[2]The Information Retrieval research literature usually speaks of "terms" because the language models used are n-gram based. All of the research done in this chapter uses a unigram language model, so "term" and "word" are used interchangeably.

where: θ is a set of parameters and h is a function, the form of which depends on the Information Retrieval model [32]; A is often the identity function.

9.4.1 BM25 Models

BM25 [153] is a probabilistic retrieval technique that uses the term frequency-inverse sentence frequency (TF-ISF) statistic as the weighting factor in the ranking function. Given a query q of terms $q_1, \ldots, q_n$, the BM25 score of a sentence s is:

$$score(q, s) = \sum_{i=1}^{n} ISF(q_i) \frac{x^s_{q_i})(k+1)}{x^s_{q_i} + k\left(1 - b + b\frac{y_s}{m}\right)} \tag{9.2}$$

k and b are free parameters; $k \in [1.2, 2.0]$ and $b = 0.75$, in absence of an advanced optimization[28]. $ISF(q_i)$ is the ISF (inverse-sentence-frequency) weight of the query term q_i. It is computed as [153]:

$$ISF(q_i) = \log \frac{N - N_{q_i} + 0.5}{N_{q_i} + 0.5} \tag{9.3}$$

9.4.2 Divergence from Randomness Models

The Divergence From Randomness (DFR) ranking method [7] is a probabilistic model that derives the term-weighting factor by measuring the divergence of the actual term distribution in an article from that obtained under a random process [170] such as the binomial and Bose-Einstein distributions. It uses two term-frequency-normalized term weights to produce optimal sentence-query matching. In a first normalization process, sentences are considered to have the same length and the information gain of an observed term is measured only when it has been accepted as a good descriptor of the observed sentence. A second normalization process is related to the sentence length and to other statistics such as uniform distribution of term frequency [40], term frequency density inversely related to sentence length[180], term frequency normalization provided by a Dirichlet prior[190], or term frequency normalization provided by a Zipfian relation[124]. The weight *weight*(t,s) of a term is the function of two probabilities $Prob_1$ which is the probability of term t that defines the notion of randomness in the whole article, and $Prob_2$ which is obtained by observing the set of all sentences in which a query term appears. This set is defined as the elite set of the term [7]. The weight of term t in a sentence s is defined as:

$$weight(t, s) = (1 - Prob_2)(-\log_2 Prob_1) \tag{9.4}$$

The ranking score of a sentence s given a query q is defined as:

$$R(q, s) = \sum_{t \in q} weight(t, s) \tag{9.5}$$

9.4.3 Information Based Similarity Models (IBS)

Information based retrieval models are based on the importance of the information brought by different terms in the sentence. Harter [68] notices that the behavior of 'significant' or 'specialty' words of a sentence tends to deviate from their average behavior in the whole article. Terms with low probability of occurrence in a sentence according to the distribution in the article are given higher consideration as they tend to convey more information in the context of the sentences they appear in.

These models are characterized by:

- a normalization function, that takes into account the number of occurrences of terms in a sentence, as well as the length of sentences,
- a probability distribution,
- a retrieval function

$$RSV(q, s) = \sum_{w \in q} -x_w^q \log Prob(X_w \geq t_w^s | \lambda_w) = \sum_{w \in q \cap s} -x_w^q \log Prob(X_w \geq t_w^s | \lambda_w) \tag{9.6}$$

λ_w is the probability distribution parameter. It is either the average number of occurrences of word w in the article, or to the average number of sentences in which w occurs. An implementation of the IBS model in the Lucene framework is of the form:

$$\sum_{w \in q \cap s} -x_w^q \log(\frac{\lambda_w}{t_w^s + \lambda_w}) \tag{9.7}$$

In this case $Prob(X_w \geq t_w^s | \lambda_w)$ is a Log-Logistic Distribution [68].

9.5 Language Models

A language model for retrieval is based on the probability distribution over the words in the language. In this study, these probability distributions are derived from the article. Both the query text and sentences in the article are assigned probability values based on the language

model. Smoothing is used for estimating the probability for missing (unseen) words.

$$P(q|s) = \sum_{q_i \in q} P(q_i|q) \log \frac{P(q_i|q)}{P(q_i|s)} \tag{9.8}$$

In this study we use two smoothing methods:

- Jelinek-Mercer smoothing

$$P(q_i|s) = (1 - \lambda)\frac{(f_{q_i}, s)}{y_s} + \lambda \frac{c_{q_i}}{L} \tag{9.9}$$

  (f_{q_i}, s) is the frequency of a query term q_i in the sentence; c_{q_i} is the query term-frequency in the article; A value of $\lambda \approx 0.1$ is suitable for short queries, and larger values (e.g. $\lambda = 0.7$) are more suitable for longer queries [189].

- Dirichlet priors smoothing

$$P(q_i|s) = \frac{(f_{q_i}, s) + \mu \frac{c_{q_i}}{L}}{y_s + \mu} \tag{9.10}$$

  μ is an article length parameter that is sentence-dependent [190].

9.5.1 Vector Space Model – Similarity based methods

The vector space model describes sentences and queries as multidimensional vectors. Each dimension corresponds to a unique term t in the article.

Each component of vector $\vec{s}$ (representing sentence s), and vector $\vec{q}$ (representing query q) is computed using a weight factor applied to TF-ISF (Term Frequency-Inverse Sentence Frequency) values of terms in sentences and queries. So,

$$\vec{s} = (W_1, W_2, ..., W_M) \tag{9.11}$$

and

$$\vec{q} = (W'_1, W'_2, ..., W'_M) \tag{9.12}$$

One popular formula used in the Lucene framework [16] modified to compute the weight of a term t (occupying location i in the vectors) in a sentence s is:

$$W_i = weight(t, s) = TF^*(t\ in\ s) \cdot ISF^*(t) = [x_w^s]^{1/2}(1 + \log(\frac{N}{F_w + 1})) \tag{9.13}$$

TF* and ISF* are the normalized versions of TF and ISF. The query term weights, W'_i, are

computed based on how many of the query terms are found in a specific sentence and a query normalization factor used to make scores comparable between queries.

Cosine similarity is often used to compute the similarity score between vector $\vec{s}$ and vector $\vec{q}$.

$$sim(\vec{s},\vec{q}) = \frac{\vec{s}\cdot\vec{q}}{||\vec{s}||||\vec{q}||} = \frac{\sum_{i=1}^{M} W_i W'_i}{\sqrt{\sum_{i=1}^{M} W_i^2}\sqrt{\sum_{i=1}^{M} W_i'^2}} \tag{9.14}$$

9.6 Experiments

We conducted experiments that can be divided into two categories:

1. linkage operations using all of the sentences comprising an article

2. linkage operations using a subset of sentences found in an article. The subset is constructed using the machine learned model for determining rhetorical categories of sentences which has been built using the method described in Chapter 6. The subset is comprised of those sentences in the article that are deemed to be in the Method rhetorical category by this model.

For each category, we set up two sets of sub-experiments: (1) in the first set of experiments, citation sentences are used as query terms, and (2) in the second set of experiments, the queries are reduced to noun phrases. This second set mimics query reformulation to improve the result when the desired information is not found in the first attempts. We experiment with different ranking techniques and the results of the rankings are compared. The Lucene framework [16] which has the implementation of these techniques is used for this purpose.

9.6.1 Lucene Indexing Process

During the indexing process, each article is segmented into sentences, as the linkage is done at the individual sentence level. Sentences are transformed into fields of content (words). An analysis step is then performed to remove stop words which are not required for the search operations. An index writer creates indexes as required. A directory is then created to store indexes (see Figure 9.1).

9.6.2 Lucene Search Operation

The searching process is described in Figure 9.2. A query expression is derived either from the unmodified citation text or a reformulation of it. An index searcher module points to the

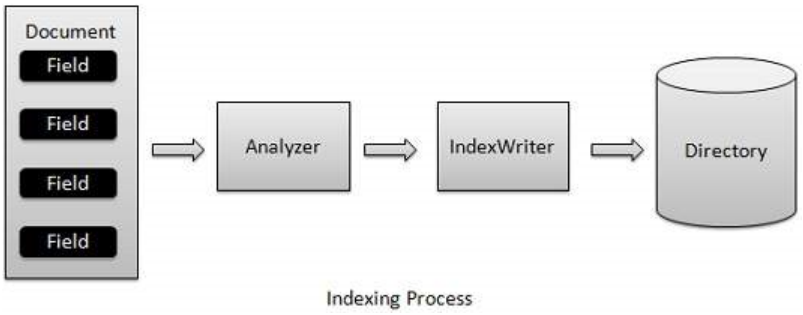

Figure 9.1: Lucene indexing process
Source : https://www.tutorialspoint.com/lucene/lucene_indexing_process.htm

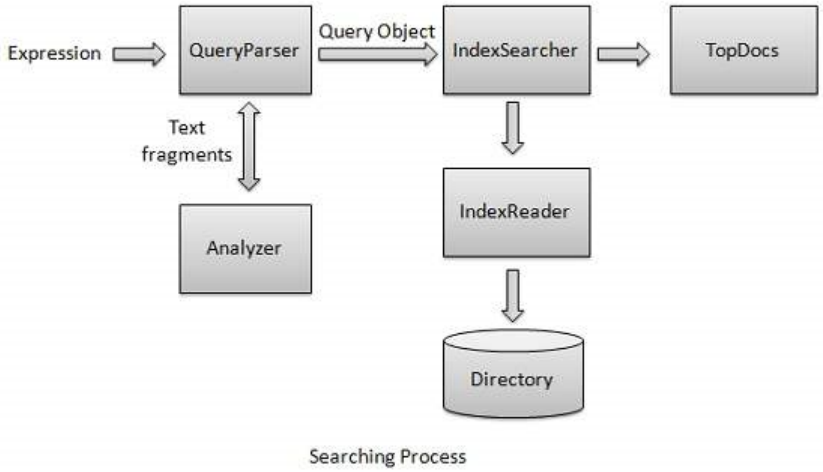

Figure 9.2: Lucene searching process
Source : https://www.tutorialspoint.com/lucene/lucene_indexing_process.htm

location where indexes are stored. A query parser operation is performed after the query terms are analyzed for stop word removal. The query expression is then passed to the searcher module which returns ranked sentences based on a ranking model.

9.7 Results and Discussion

We use two evaluation metrics for this task, namely: Normalized Discounted Cumulative Gain at rank k (*NDCG@k*) and *Precision@k*. *NDCG@k* takes into account the multi-level utility score of each sentence, whereas *Precision@k* only takes into account the binary relevance of each sentence. We put more emphasis on the *NDCG@k* metric due to the multi-level relevance of the dataset.

Due to the particular nature of this retrieval task, we assume that for a given (query, article) pair, the evaluation score will depend on the number of candidate sentences, which varies

depending on the paper being considered. Therefore the computation of an overall average score for all the papers is done after we compute the individual score for each paper, taking into account different numbers of candidate sentences. For a total number of n papers, if paper$_i$ has k_i candidates, we first compute $Precision@k_i$ for each paper$_i$, before computing the average precision *Avg. Precision*.

$$Avg.\ Precision = \frac{\sum_{i=1}^{n} Precision@k_i}{n} \tag{9.15}$$

Similarly:

$$Avg.\ NDCG = \frac{\sum_{i=1}^{n} NDCG@k_i}{n} \tag{9.16}$$

Besides the average scores (Mean), we also present five summary statistics for each retrieval model, namely: minimum, maximum,1st quartile, median and 3rd quartile over all the papers.

Following the presentation of these summary statistics, we present the $Precision@k$ and $NDCG@k$ values for each paper for one of the retrieval models, LMJ. LMJ has been chosen simply to highlight the per paper statistics, as the best retrieval models show similar performance in each experimental category.

9.7.1 Experiments with full papers

In this set of experiments, the linkage operation is performed between the citation text as the query and all of the sentences of the target paper as potential candidate referenced sentences.

Table 9.2 shows the results for the experiments using the full citation sentence as query input and all of the sentences in an article as candidate referenced sentences. Table 9.3 shows the results of the experiments using noun phrases extracted from the citation sentence as query input and all of the sentences in an article as candidate referenced sentences.

Most of the retrieval models, i.e., Vector Space Model (VSM), Information Based System (IBS), Divergence From Randomness (DFR), Language Model with Jelinek-Mercer smoothing (LMJ) and Language Model with Dirichlet priors smoothing (LMD), show similar performance, except for Probabilistic BM25 (BM25), which has much lower Mean and Median values.

The scores shown in Table 9.2 vary between 0 and 0.66 (66%) for the $Precision@k_i$, and between 0 and 0.7751 (77%) for the $NDCG@k_i$ for the experiments using full citation sentences as query input.

The scores shown in Table 9.3 vary between 0 and 1 (100%) for the $Precision@k_i$, and between 0 and 1 (100%) for the $NDCG@k_i$ values for the experiments using noun phrases for query input. Therefore, models with query reduction using noun phrases have overall slightly

Precision@k

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
IBS	0.0000	0.2319	0.3229	0.3212	0.4000	0.6667
VMS	0.0000	0.2319	0.3542	0.3395	0.4821	0.6667
BM25	0.0000	0.0178	0.2283	0.2506	0.4125	0.6667
DFR	0.0000	0.0982	0.3333	0.2972	0.4214	0.6667
LMJ	0.0000	0.2500	0.3542	0.3412	0.4904	0.6667
LMD	0.0000	0.2500	0.3205	0.3309	0.3833	0.6667

NDCG@k

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
IBS	0.0000	0.1362	0.3293	0.3168	0.4825	0.7751
VMS	0.0000	0.1778	0.3271	0.3237	0.4703	0.7751
BM25	0.0000	0.0158	0.2984	0.2514	0.3766	0.7654
DFR	0.0000	0.0732	0.3222	0.3042	0.4601	0.7751
LMJ	0.0000	0.1360	0.3412	0.3247	0.4856	0.7751
LMD	0.0000	0.2037	0.2961	0.3227	0.3966	0.7751

Table 9.2: Evaluation of six retrieval methods with Citation Sentence as queries and all of the sentences in an article as candidate referenced sentences — Statistics for *Precision@k* and *NDCG@k*

Precision@k

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
IBS	0.0000	0.2231	0.3542	0.3557	0.5000	1.0000
VMS	0.0000	0.2270	0.3333	0.3476	0.4571	1.0000
BM25	0.0000	0.0000	0.1905	0.2362	0.4125	0.6667
DFR	0.0000	0.2231	0.3333	0.3386	0.4821	1.0000
LMJ	0.0000	0.2319	0.3333	0.3518	0.4256	1.0000
LMD	0.0000	0.2007	0.3095	0.3114	0.3962	0.6842

NDCG@k

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
IBS	0.0000	0.1812	0.3120	0.3375	0.4552	1.0000
VMS	0.0000	0.1888	0.3040	0.3376	0.4694	1.0000
BM25	0.0000	0.0000	0.2224	0.2411	0.3712	0.7654
DFR	0.0000	0.1827	0.3024	0.3218	0.4407	1.0000
LMJ	0.0000	0.2198	0.3206	0.3590	0.4668	1.0000
LMD	0.0000	0.1792	0.2662	0.2932	0.3774	0.7654

Table 9.3: Evaluation of six retrieval methods with Noun Phrase as queries and all of the sentences in an article as candidate referenced sentences — Statistics for *Precision@k* and *NDCG@k*

better performances. But this better performance doesn't generalize to all the papers.

The best performance is shown by the LMJ model using noun phrases extracted from the citation text as the queries (see Table 9.3, row in light cyan). For this model and type of query, we can report the following observations for the *NDCG@k* metric:

- **1st Quartile (25%)** is 0.2198 ($\approx$ 22%). This implies that 75% of the papers (18 out of 22) have their *NDCG@k* scores above 0.2198.
- **2nd Quartile (Median 50%)** is 0.3206 ($\approx$ 32%). This implies that 50% of the papers (12 out of 22) have their *NDCG@k* scores above 0.3206.
- **3rd Quartile (75%)** is 0.4668 ($\approx$ 78%). This implies that about 25% of the papers (6 out of 22) have their *NDCG@k* scores above 0.4668.

Table 9.4 shows the results per paper for the LMJ model using the full citation sentence as query input and all of the sentences in an article as candidate referenced sentences for the LMJ model. Table 9.5 shows the results per paper using noun phrases extracted from the citation sentence as query input and all of the sentences in an article as candidate referenced sentences for the LMJ model. Rows in light cyan represent papers with the highest *Precision@k* and *NDCG@k*. Rows in orange represent papers having the lowest values of *Precision@k* and *NDCG@k* (with one exception). We can see that paper #4 has the highest *Precision@k* for both full citation sentence and noun phrases only as query inputs. On a paper by paper comparison, the experiments with noun phrases as queries have slightly better *Precision@k* and *NDCG@k* values. Specifically, 8 papers do better with noun phrases as queries, 7 do worse, and 7 are equal on the *Precision@k* metric. With the *NDCG@k* measure, the respective number of papers are 8, 6, and 6.

9.7.2 Experiments with Method sentences

In this section, we investigate how the reduction of the number of candidate sentences can affect the linkage effectiveness. To this end, we have reduced the sentences in the articles to only Method sentences using the techniques described in Chapter 6. Recalling the numbers from Chapter 8, the dataset containing all of the sentences in all of the papers has 4258 sentences and the dataset containing only the Method sentences contains 1453 sentences. Table 9.6 shows the summary results for the first set of experiments using full citation sentence as query input and only the Method sentences as candidate referenced sentences. Table 9.7 shows the summary results for the second set of experiments using noun phrases extracted from the citation sentence as query input and only the Method sentences as candidate referenced sentences.

Paper Number	Number of sentences	Precision number (k)	*Precision@k*	*NDCG@k*
1	126	19	0.5263	0.5027
2	166	16	0.375	0.3547
3	150	12	0.5833	0.4874
4	162	3	0.6667	0.7654
5	194	18	0.0556	0.0506
6	185	3	0	0
7	169	4	0.5	0.4144
8	291	7	0.4286	0.2042
9	233	3	0.3333	0.2961
10	224	8	0.625	0.7031
11	315	31	0.2581	0.3525
12	89	5	0.4	0.3452
13	239	2	0	0
14	236	12	0.25	0.2361
15	249	14	0.1429	0.1132
16	189	8	0.25	0.3373
17	112	15	0.4	0.4839
18	143	6	0.6667	0.7751
19	185	4	0	0
20	165	3	0.3333	0.2346
21	266	3	0	0
22	170	13	0.4615	0.4862

Table 9.4: Example of evaluation results per paper using citation sentence as queries and full paper sentences as candidate referenced sentences with the Language Model with Jelinek-Mercer smoothing (LMJ)

The best performing model is LMD (see Table 9.7, row in light cyan). We can report the following observations with the *NDCG@k* metric:

- **1st Quartile (25%)** is 0.3645 ($\approx$ 36%). This implies that 75% of the papers (18 out of 22) have their *NDCG@k* scores above 0.3645.
- **2nd Quartile (Median 50%)** is 0.4862 ($\approx$ 49%). This implies that 50% of the papers (12 out of 22) have their *NDCG@k* scores above 0.4862.
- **3rd Quartile (75%)** is 0.5794 ($\approx$ 58%). This implies that about 25% of the papers (6 out of 22) have their *NDCG@k* scores above 0.5794.

The *Precision@k* and *NDCG@k* scores for all of the models using both citation sentences and noun phrases as queries have increased when compared with the results in Tables 9.2 and

Paper Number	Number of sentences	Precision number (k)	*Precision@k*	*NDCG@k*
1	126	19	0.6842	0.7135
2	166	16	0.3125	0.2716
3	150	12	0.4167	0.3792
4	162	3	1	1
5	194	18	0.2222	0.2828
6	185	3	0.3333	0.2346
7	169	4	0.25	0.1952
8	291	7	0.4286	0.4791
9	233	3	0.3333	0.2961
10	224	8	0.625	0.7134
11	315	31	0.2258	0.2911
12	89	5	0.4	0.3452
13	239	2	0	0
14	236	12	0.25	0.2149
15	249	14	0.1429	0.1116
16	189	8	0.375	0.4171
17	112	15	0.6	0.6498
18	143	6	0.5	0.4593
19	185	4	0	0
20	165	3	0.3333	0.4693
21	266	3	0	0
22	170	13	0.3077	0.3733

Table 9.5: Example of evaluation results per paper using noun phrases as queries and full paper sentences as candidate referenced sentences with the Language Model with Jelinek-Mercer smoothing (LMJ)

9.3. As an example, for the LMJ model using noun phrases as queries, the mean *Precision*@k has increased from 0.3518 to 0.4704 and the mean *NDCG@k* has increased from 0.3590 to 0.4371. Also, a maximum value of 100% for *Precision@k* and a maximum value of 100% for *NDCG@k* are obtained for all the ranking models for one paper (Paper #4).

Table 9.8 shows the results per paper for the LMJ model using the full citation sentence as query input and only the Method sentences in an article as candidate referenced sentences. Table 9.9 shows the results per paper using noun phrases extracted from the citation sentence as query input and only the Method sentences in an article as candidate referenced sentences for the LMJ model. Rows in light cyan represent papers with the highest *Precision@k* and *NDCG@k*.

On a paper by paper comparison, the experiments with noun phrases as queries have slightly better *Precision@k* values. The *NDCG@k* values show no better performance by citations as

Precision@k

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
IBS	0.0000	0.3472	0.4808	0.4550	0.6116	1.0000
VMS	0.0000	0.4042	0.5000	0.4826	0.6250	1.0000
BM25	0.0000	0.3333	0.4476	0.4064	0.5781	1.0000
DFR	0.0000	0.3438	0.4727	0.4741	0.6116	1.0000
LMJ	0.0000	0.4042	0.5000	0.4854	0.6299	1.0000
LMD	0.0000	0.3438	0.4919	0.4976	0.6188	1.0000

NDCG@k

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
IBS	0.0000	0.3185	0.4830	0.4569	0.5968	1.0000
VMS	0.0000	0.3860	0.5197	0.4906	0.5981	1.0000
BM25	0.0000	0.2500	0.4644	0.3958	0.5479	1.0000
DFR	0.0000	0.3176	0.4906	0.4722	0.5968	1.0000
LMJe	0.0000	0.3807	0.5074	0.4798	0.5863	1.0000
LMD	0.0000	0.3336	0.5163	0.4913	0.6152	1.0000

Table 9.6: Evaluation of six retrieval methods with Citation Sentence as queries and only the Method sentences as candidate referenced sentences — Statistics for *Precision@k* and *NDCG@k*

Precision@k

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
IBS	0.0000	0.3333	0.4722	0.4451	0.5692	1.0000
VSM	0.0000	0.3462	0.5000	0.4799	0.6250	1.0000
BM25	0.0000	0.2125	0.4446	0.3962	0.5417	1.0000
DFR	0.0000	0.3333	0.4365	0.4540	0.5692	1.0000
LMJ	0.0000	0.3333	0.4643	0.4704	0.6250	1.0000
LMD	0.0000	0.3500	0.5000	0.4988	0.6458	1.0000

NDCG@k

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
IBS	0.0000	0.2961	0.4464	0.4207	0.4937	1.0000
VMS	0.0000	0.3184	0.4748	0.4514	0.5335	1.0000
BM25	0.0000	0.1796	0.3986	0.3702	0.5469	1.0000
DFR	0.0000	0.2961	0.4450	0.4321	0.4944	1.0000
LMJ	0.0000	0.2961	0.4596	0.4371	0.5335	1.0000
LMD	0.0000	0.3645	0.4862	0.4886	0.5794	1.0000

Table 9.7: Evaluation of six retrieval methods with Noun Phrase as queries and only the Method sentences as candidate referenced sentences — Statistics for *Precision@k* and *NDCG@k*

queries or noun phrases as queries. Specifically, 10 papers do better with noun phrases as queries, 6 do worse, and 6 are equal on the *Precision@k* metric. With the *NDCG@k* measure,

Paper Number	Number of sentences	Precision number (k)	*Precision@k*	*NDCG@k*
1	59	19	0.6316	0.6741
2	65	16	0.6875	0.5847
3	59	12	0.6667	0.5676
4	24	3	1	1
5	60	18	0.5556	0.576
6	44	3	0	0
7	53	4	0.5	0.6367
8	92	7	0.5714	0.4791
9	97	3	0.3333	0.2961
10	65	8	0.625	0.4785
11	93	31	0.4516	0.5278
12	32	5	0.4	0.4704
13	47	2	0	0
14	103	12	0.4167	0.3857
15	40	14	0.4286	0.379
16	99	8	0.5	0.5869
17	53	15	0.5333	0.5908
18	65	6	0.6667	0.7619
19	62	4	0.25	0.2463
20	39	3	0.6667	0.5307
21	119	3	0.3333	0.2961
22	83	13	0.4615	0.4869

Table 9.8: Example of evaluation results per paper using citation sentences as queries and Method sentences as candidate referenced sentences with the Language Model with Jelinek-Mercer smoothing (LMJ)

the number of papers that do better with citations as queries are 9, that do better with noun phrases as queries are 9, and that perform equally are 4.

Rows in orange represent papers having the lowest values of *Precision@k* and *NDCG@k*. In each of these latter cases, the values are significantly lower than the other *NDCG@k* values. We can see that paper #4 has the highest *Precision@k* for both full citation sentence and noun phrases only as query inputs. Experiments with citation sentences as queries tend to achieve the higher *Precision@k* and *NDCG@k* values.

On a paper by paper comparison, the experiments with citation sentences as queries have slightly better *Precision@k* and *NDCG@k* values. Specifically, 7 papers do better with citation sentences as queries, 4 do worse, and 11 are equal on the *Precision@k* metric. With the *NDCG@k* measure, the number of papers that do better with citations as queries are 8, that do better with noun phrases as queries are 5, and that perform equally are 9.

Paper Number	Number of sentences	Precision Number (k)	*Precision@k*	*NDCG@k*
1	59	19	0.6842	0.7339
2	65	16	0.625	0.5004
3	59	12	0.6667	0.5676
4	24	3	1	1
5	60	18	0.5	0.5344
6	44	3	0.3333	0.2346
7	53	4	0.25	0.1952
8	92	7	0.5714	0.4791
9	97	3	0.3333	0.2961
10	65	8	0.625	0.4974
11	93	31	0.3548	0.4487
12	32	5	0.4	0.4704
13	47	2	0	0
14	103	12	0.3333	0.2618
15	40	14	0.4286	0.3774
16	99	8	0.625	0.5591
17	53	15	0.7333	0.7721
18	65	6	0.5	0.429
19	62	4	0	0
20	39	3	0.6667	0.5307
21	119	3	0.3333	0.2961
22	83	13	0.3846	0.4321

Table 9.9: Example of evaluation results per paper using noun phrases as queries and Method sentences as candidate referenced sentences with the Language Model with Jelinek-Mercer smoothing (LMJ)

A comparison with the scores for full papers shows an improvement by 13 percentage points (from 0.3590 to 0.4886) over all the papers. Based on these results we can say that a better citation linkage can be achieved when the candidate sentences are reduced to sentences of the same category as the citation sentences. Also, the length of the query can affect the results. For all the experiments, when the citation sentences are reduced to noun phrases, we can notice some slight improvement. But this can not be generalized to all the papers. However some questions arise as why this reduction has no effect on some papers whose evaluation scores have not improved. For instance:

- papers 13 and 19 did not yield any relevant sentence for *Precision@k* and *NDCG@k* for both full sentence articles and reduced sentence ones.

This can be due to the following reasons:

Citation Sentence	
A biochemical reaction system is parameterized in terms of molecular capacities and reaction resistances, by using a thermodynamic kinetic modeling (TKM) formalism that enjoys a number of advantages over the ones suggested.	
Candidate Sentences	Rating
In contrast to ad-hoc rate laws such as linlog or generalised mass-action kinetics the convenience kinetics is biochemically justified as a direct generalisation of the Michaelis-Menten kinetics; it is saturable and allows for activation and inhibition of the enzyme.	4
The parameters kM kA and kI represent concentrations that lead to half-maximal (or in general -maximal effects: the kM values also indicate the threshold between low substrate concentrations that lead to linear kinetics and high concentrations at which the enzyme works in saturation.	1

Table 9.10: Annotation for Paper 13. These sentences are the ones annotated as candidate sentences with the confidence rating scores provided by the annotator. All other sentences in paper 13 are annotated as non-candidate sentences.

1. The number of relevant sentences is too small. This number is 2 and 4 for paper 13 and 19 respectively, whereas the average number of candidate sentences is 8.

2. The choice of most candidate sentences involves the use of some inference that is not easily translated into retrieval models.

3. It is difficult to find the best match for some citations when the matching operation involves external domain specific resources that are not yet available.

These reasons can be illustrated by looking at the annotations of these two sentences.

In the case of Paper 13 (see Table 9.10), only 2 candidate sentences were provided; one with high rating and the other with very low rating. The linkage here is also at a conceptual level. The *Michaelis-Menten kinetics* in the first candidate sentence is a **kind of** *"thermodynamic kinetic modeling (TKM) formalism"* in the citation sentence. The linkage can be done based on this semantic relation between citation terms and candidate sentence terms. However, this semantic resource is not yet available.

In the case of Paper 19 (see Table 9.11), the annotation rating is very low, in the range 1–2. This means that even though matching candidate sentences exist in the cited paper, they share very little information with the citation.

Based on these observations, some linkage operations will require some deep semantic preprocessing in order to be achievable. However despite these shortcomings, the overall results prove that citation linkage as a retrieval task might be a good way to provide the adequate solution for linking citation sentence and the sentences they refer to in the cited paper.

Citation Sentence	
Hybrid laccases combining Ascomycotina sequences and positively selected sites identified in Basidiomycotina could prove useful for testing new physico-chemical properties for biotechnology applications.	
Candidate Sentences	Rating
A model that allows omega ratio to vary among sites and lineages was therefore developed to improve the detection of positive selection.	1
In order to test these connections we first focused on an analysis of the well-described fungal lipase/feruloyl esterase A family.	1
Taken together these results confirmed that positively selected sites are unambiguously involved in the functional shift.	2
In conclusion biological data from mutagenesis experiments confirmed that the positively selected sites had been robustly identified and made it possible to establish the connection between positively selected sites and functional shifts.	2

Table 9.11: Annotation for Paper 19. These sentences are the ones annotated as candidate sentences with the confidence rating scores provided by the annotator. All other sentences in paper 19 are annotated as non-candidate sentences.

Citation Sentence		
This complex expression can also be obtained using our KAPattern package.		
Ranked Sentences	Rating	Score
Our stand-alone KAPattern package is developed using MATLAB GUI.	1	7.08
We present here a simple stand-alone computer program written in MATLAB GUI called KAPattern for generating rate equations in complex enzyme systems.	4	5.15
We have described a systematic method and the corresponding computer program called KAPattern for generating rate equations for any complex enzyme systems.	3	5.15

Table 9.12: Ranking Example for Paper 4. These sentences are the ones ranked as candidate sentences with the scores provided by the IBS model. All other sentences in Paper 4 are ranked as non-candidate sentences.

9.7.3 Examples of a Ranking Output

Table 9.12 shows the ranking for Paper 4 (using all of the sentences in the article as candidate sentences). All the candidate sentences are retrieved in the top k (3) positions as expected; thus the highest value of precision is attained (Precision = 100%). However the expected order is not totally achieved as the first ranked sentence is expected to be the third. Despite this outcome, we can still say that the information need that we tend to provide users is realized as all of the expected sentences are retrieved.

Table 9.13 shows the ranking for Paper 18 (using all of the sentences in the article as candi-

Citation Sentence		
Quantitative RT-PCR was carried out using a Rotor-Gene 2000 centrifugal real-time cycler (Corbett Research) using the Platinum SYBR Green qPCR SuperMix-UDG (Invitrogen) as described previously.		
Ranked Sentences	Rating	Score
Quantitative RT-PCR was carried out using a Rotor-Gene 2000 centrifugal real-time cycler (Corbett Research) using the Platinum SYBR Green qPCR SuperMix-UDG (Invitrogen).	5	20.33
Each reaction contained: 12.5 μl of the Platinum SYBR Green qPCR SuperMix-UDG 200 nM 300 nM or 400 nM of forward and reverse primers and 5 μl cDNA (1:40 RNA dilution) to a final volume of 25 μl.	3	6.49
Real-time quantitative PCR (qPCR) has become a very powerful tool for gene expression studies.	0	5.51
Dauers from N2 were obtained as described in Houthoofd et al.	0	4.74
SYBR Green fluorescence of the generated products was continuously monitored throughout the temperature ramp from 60 to 99°C.	1	4.59
The cycling conditions were as follows: 50°C for 2 min initial denaturation at 95°C for 2 min followed by 45 cycles of 15 s at 95°C 30 s at 60°C and 30 s at 72°C (gain set at 8 for SYBR Green).	2	3.22

Table 9.13: Ranking Example for Paper 18. These sentences are the ones ranked as candidate sentences with the scores provided by the IBS model. All other sentences in paper 18 are ranked as non-candidate sentences.

Category	Ranked	Expected	Percent
5	20	33	(60 %)
4	21	48	(44 %)
3	18	42	(43 %)
2	6	42	(14 %)
1	8	44	(18 %)

Table 9.14: Statistics of the Information Based Similarity model (IBS) ranked candidate sentences over all the papers.

date sentences). Four out of six sentences are retrieved, but not in the right order. However the fact that many highly rated sentences are in the top position, we can say that users' information needs are somewhat realized. The overall statistics of the ranked candidate sentences in the top k are presented in Table 9.14.

Table 9.15 presents the predicted number of sentences per category and per paper as given by the VSM model with noun phrases as query input.

We can notice that fewer papers rated 5 and 4 by the human annotator have been ranked as a 0 compared to papers rated 3, 2 and 1. This means that the papers rated 1, 2, and 3 may require biochemical knowledge and more background information for the linkage to be effective. 60%

Paper #	Category 1 Pred/Exp	Category 2 Pred/Exp	Category 3 Pred/Exp	Category 4 Pred/Exp	Category 5 Pred/Exp
1	1/4	1/3	5/6	4/4	2/2
2	0/0	2/4	2/4	1/3	1/5
3	0/0	0/0	1/7	1/1	3/4
4	1/1	0/0	1/1	1/1	0/0
5	1/6	0/1	1/1	2/9	0/1
6	0/0	0/0	0/1	0/1	1/1
7	0/1	0/0	0/1	0/0	1/2
8	0/3	0/0	2/3	1/1	0/0
9	0/1	0/1	1/1	0/0	0/0
10	0/0	0/0	0/3	2/2	3/3
11	1/2	0/17	0/4	3/3	3/5
12	1/3	0/0	0/1	1/1	0/0
13	0/1	0/0	0/0	0/1	0/0
14	1/8	0/0	0/0	1/3	1/1
15	1/4	0/2	0/2	0/5	0/1
16	0/2	0/0	3/3	0/2	1/1
17	0/1	2/6	0/1	3/5	2/2
18	0/2	0/1	2/2	0/0	1/1
19	0/2	0/2	0/0	0/0	0/0
20	0/0	0/0	0/0	0/2	0/1
21	0/0	0/0	0/1	0/0	1/2
22	1/3	1/5	0/0	1/4	0/1

Table 9.15: Statistics of predicted and expected sentences per category and per paper as given by the Vector Space Model with noun phrases as query input. Pred/Exp: Predicted/Expected

and 44% of the papers rated 5 and 4 are ranked respectively in the top positions. The ranking for the papers rated 5 have improved over the results in Chapter 8.

9.7.4 Comparison with other work

Our results show that it is possible to find the set of sentences a citation refers to in a cited paper with reasonable performance. Even though, there have been previous works to match citation sentences and related cited papers, this current work is first of its kind. In [34], an attempt to match citation text and cited spans in biomedical literature proved to be a difficult task with limited performance. In that work, a matching candidate can be formed by more than one sentence and relevant spans can overlap. Their best performing system achieve only 0.224 precision. However, the authors reported promising results with query reduction to noun phrases and UMLS expansion. A comparison with our results does show that query reduction might

Algorithm	p-value	Mean (Linear Regression)	Mean (Retrieval models)	Difference in Means
IBS_NDCG	0.8132	0.3009	0.3167	0.0158
IBS_NDCG_NP	0.5915	0.3009	0.3375	0.0366
VSM_NDCG	0.7371	0.3009	0.3237	0.0228
VSM_NDCG_NP	0.6011	0.3009	0.3375	0.0366
BM25_NDCG	0.4406	0.3009	0.2514	−0.0494
BM25_NDCG_NP	0.3785	0.3009	0.2411	−0.0597
DFR_NDCG	0.9613	0.3009	0.3041	0.0032
DFR_NDCG_NP	0.7538	0.3009	0.3218	0.0209
LMJ_NDCG	0.7284	0.3009	0.3246	0.0237
LMJ_NDCG_NP	0.4075	0.3009	0.35891	0.0580
LMD_NDCG	0.7245	0.3009	0.3226	0.0217
LMD_NDCG_NP	0.9008	0.3009	0.2931	−0.0077

Table 9.16: Comparison statistics between the Linear Regression Model and the Information Retrieval Models using full paper sentences as candidate referenced sentences. NDCG refers to the model retrieving with full citation as queries. NDCG_NP refers to model retrieving with noun phrases as queries.

be of great value in most cases. However, the current study can only partially be compared to [34] as they limit their work to only binary relevance.

Also, [33] uses similar techniques for citation-based summarization of biomedical literature and shows that such a task is difficult compared to regular text retrieval tasks. Based on our results with multilevel relevance we do think that we can obtain good performance with the appropriate retrieval method.

9.7.5 Comparison with Ranking with LM

In Chapter 8, we presented the linkage task as a machine learning task and used Linear Regression to build models to rank sentences in the target paper. In this section we compare the Linear Regression models with the retrieval models built in this chapter.

Tables 9.16 and 9.17 show the statistics for full article experiments and Method sentences only experiments, respectively. In all of these cases the p-values show that there is no statistically significant improvement between the Linear Regression models and the best retrieval models. Improvements are considered statistically significant if $p < 0.05$. All of the p-values (absolute values) are greater than 0.05. However, a look at the Mean Average values shows that all but the BM25 retrieving models outperformed the Linear Regression model in each experimental category. For experiments using full paper sentences, the highest difference in means (0.0580, about 5.8%) is obtained with the LMJ_NDCG_NP (Language Model with Jelinek-

Algorithm	p-value	Mean (Linear Regression)	Mean (Retrieval models)	Difference in Means
IBS_NDCG	0.5440	0.4098	0.4569	0.0470
IBS_NDCG_NP	0.8875	0.4098	0.4206	0.0108
VSM_NDCG	0.2890	0.4098	0.4906	0.0808
VSM_NDCG_NP	0.5786	0.4098	0.4514	0.0416
BM25_NDCG	0.8599	0.4098	0.3957	−0.0140
BM25_NDCG_NP	0.6246	0.4098	0.3701	−0.0396
DFR_NDCG	0.4090	0.4098	0.4721	0.0624
DFR_NDCG_NP	0.7636	0.4098	0.4321	0.0223
LMJ_NDCG	0.3513	0.4098	0.4797	0.0699
LMJ_NDCG_NP	0.7184	0.4098	0.4370	0.0273
LMD_NDCG	0.2651	0.4098	0.4912	0.0815
LMD_NDCG_NP	0.2849	0.4098	0.4885	0.0787

Table 9.17: Comparison statistics between the Linear Regression Model and the Information Retrieval Models using Method sentences as candidate referenced sentences. NDCG refers to the model retrieving with full citation as queries. NDCG_NP refers to the model with noun phrases as queries.

Mercer smoothing and noun phrases as query input). Similarly, for experiments using Method sentences as candidate referenced sentences, the highest difference in means (0.0814, about 8.1%) is obtained with the LMD_NDCG (Language Model with Dirichlet smoothing and full citation sentence as query input).

9.8 Chapter Summary

The citation linkage task aims at providing *focused* information to the reader and one best way to do that is to treat the problem as an information retrieval task. While most retrieval techniques usually apply to large collections of documents, they all involve text matching based on document content similarity. The same matching operation can be achieved at the sentence level as is in the case of the linkage between a citation sentence and its cited sentences in a cited article. For this purpose, we have used six retrieval techniques to rank sentences in cited research articles, based on citation sentences taken from citing papers. Our results show that the information needs that the linkage task tends to achieve, can be obtained with the appropriate ranking technique. We notice that while most of the retrieval techniques provide some good results, the information based models and the language based models seem to perform better. We are aware that some papers don't give good scores; this can be due to the fact that their annotation requires more inferential and background information. We intend to investigate how to include these features in the ranking models in future work.

Chapter 10

Thesis Summary and Conclusion

The goal of the study is to design a framework for finding sentences that are cited in a given article, a task we have called *citation linkage*. For this purpose, the following objectives were defined for the study:

- Building of a citation linkage dataset.
- Identify method mentions from scientific sentences.
- Identifying target sentences belonging to the same rhetorical category as the citation sentence.
- Investigating machine learning techniques for citation linkage.
- Investigating information retrieval techniques for citation linkage.

We present in this chapter a summary of the results attained for each objective and discuss future work.

10.1 Building Datasets and Corpora for Citation Linkage

In Chapter 3, we present the overall setting for the creation of different datasets that we deem necessary for the task of matching citation sentences and cited sentences in a target article. Three datasets were created:

- **Method mention extraction dataset**: The aim was to build a framework for extracting methods and techniques used in biomedical articles. A comprehensive "method sentences" dataset was built using a linguistic-based heuristic and manual curation. This

results in two complementary datasets comprising sentences mentioning method terminologies. The first dataset comprises 918 pairs of sentences containing the first category of method mention, i.e., terminology units ending with a method keyword. The second dataset comprises 122 pairs of sentences of method mentions that don't contain a method keyword. We have shown that the rhetorical structure of scientific research articles usually predisposes sentences to belong to categories that can be detected using linguistic rules. However, we still need human input to validate the dataset to be suitable for research experimentations, albeit at a lower cost.

- **Sentence classification corpus**: A corpus was created to classify sentences from scientific research papers to match the IMRaD rhetorical structure classes such as, Introduction, Method, Result, and Discussion/Conclusion. As with the previous datasets in Section 3.3, our assumption is that a sentence that appears in the co-referential context of the co-referencing phrase "This method", will likely talk about a methodology used in a research experiment and reported in the paper. Similarly, a sentence that starts with the expression "This result" is likely to refer to an experimental result context, etc. We define the co-referential context of these phrases to be a small number of sentences preceding the sentences in which these "This" references occur. To collect sentences that belong to the "Result" category, our target candidates are those sentences that come immediately before sentences starting with "This result…". Similarly, to collect sentences that belong to the "Conclusion" category, our target candidates are those sentences that come immediately before sentences starting with "This conclusion…". The benefits of these techniques are manifold: It relies on the sentence patterns presented in large repositories of scientific research papers; it is less time-consuming than a human annotated human corpus; it can be validated by established machine-learning techniques.

- **Building of a Citation Linkage corpus**: An annotation guideline was defined to match a given citation sentence with candidate cited sentences based on the following criteria:

  - To what extent can the person who reads a citation sentence taken in isolation be able to determine the candidate sentences that have been cited in a reference paper, i.e., for a giving citation sentence from a citing paper *A*, there are one or more sentences from the cited paper *B* that are similar in terms of word content, domain knowledge, etc. and we would like a domain expert to be able to identify such candidate sentences.
  - Candidate sentences are chosen from the full article and presented chronologically as they appear in the article.

– After the selection of the candidate sentences, each sentence is given a similarity strength ranging from 0 (no) to 5 (strong) similarity. Annotator feedback is collected and point to the fact that candidate sentences were chosen based on surface level similarity as well as non explicit factors such as background domain knowledge, and inferential deduction.

10.2 Method Mention Extraction from Scientific Papers

In Chapter 4, we use a rule-based method to extract method mentions that contain keywords, such as "algorithm", "technique", "analysis", "approach" and "method") and machine learning techniques to extract the second category of method mention (those that don't contain the above keywords). We achieved a precision of 85.40% for the rule-based method. For the machine learning task, precision is 81.8%, recall is 75 and F-score is 78.26%. We then showed that we can extract many of these terms using simple grammatical patterns. A few other terms can be extracted with machine learning techniques. This is an interesting contribution to the scientific community in terms of research in named entity recognition (NER) and information extraction from biomedical unstructured texts.

10.3 Sentence Classification

In Chapter 8, we use machine learning techniques to classify sentences from citing papers into the IMRaD's method category. Using a self-annotated and machine-validated corpus, we performed three classification experiments. In the first classification task, we have used the self-annotating dataset to build a model and we test it with instances from the corpus found in [1]. Only three categories (Method, Result, Conclusion) are used. In the second classification task, a classifier is trained with the machine validated dataset using 10 fold cross-validation. The model achieves an F-measure score of 97%. In the third classification task, a classifier is trained with the machine validated dataset together with the "Background/Introduction" category instances taken from a previously annotated dataset found in [1]. We applied 10 fold cross-validation and achieved an overall F-score of 93.6% with Multinomial Naïve Bayes and 95% with SVM. The contributions are the dataset that is used to build the learning models as well as the improvement that we have over existing models built with more limited human annotated datasets. We have then shown that sentence classification into the four rhetorical categories, Introduction, Methods, Results, and Discussion (IMRaD), can benefit from a combination of human annotation and computer validation.

10.4 Citation Linkage as a Machine Learning Task

In Chapter 8, we presented the citation linkage problem as a machine learning task and showed that moderate correlations (0.25) could be achieved when we apply regression models with the datasets. When we reduced the dataset to sentences belonging to the "Method" rhetorical category, the correlation scores have improved by 15 percentage points. However, a high correlation (greater than 0.50) was not achieved due to the unbalanced nature of the dataset. Also the similarity measures failed most of the time to discriminate between the data instances belonging to different annotator rating levels. When we use the learned models to rank the predicted scores for sentences in a target article, 18 papers out of 22 (80%) have at least one sentence ranked in the top k positions (k being the number of relevant sentences per paper) and 10 papers (45%) have their Normalized Discounted Cumulative Gain at rank k (*NDCG@k*) scores greater than 43% and *Precision@k* greater than 44%. The Average *NDCG@k* is 30% and the Average *Precision@k* is 29% over all the papers for experiments using full paper sentences as candidate referenced sentences. The Average *NDCG@k* is 41% and the Average *Precision@k* is 42% over all the papers for experiments Method sentences as candidate referenced sentences. These results are very promising and can be improved with better ranking algorithms.

10.5 Citation Linkage as an Information Retrieval Task

In Chapter 9, we tasked the linkage problem to liken it to the way text retrieval techniques tend to resolve a user information request by matching query input with a relevant set of text documents. Based on well established document ranking methods and information retrieval evaluation metrics, we showed that the matching operation between citation sentences and equivalent cited sentences can be performed for most scientific research papers. For each citation-paper linkage task, we computed *Precision@k* and *NDCG@k* (k is fixed to the number of candidate sentences as given by annotators), and found that 18 out of 22 individual linkage operations have at least one sentence in the top k positions. When the dataset is reduced to Method sentences, the result is 20 out of 22. Furthermore, 13 linkage tasks have *Precision@k* and *NDCG@k* above 36% for experiments using full paper sentences as candidate referenced sentences and above 47% for experiments using Method sentences as candidate referenced sentences. Fewer sentences rated by the annotators as 5s and 4s have been ranked as a 0 compared to those sentences rated as 3s, 2s and 1s, as 60% and 44% of the 5s and 4s are ranked in the top positions, respectively. This may mean that the 1s, 2s, and 3s may also require biochemical knowledge and more background information in the ranking model for the linkage to be effective.

Based on these results, we can conclude that the linkage task can be performed using information retrieval techniques. We are aware that further investigation of the reasons why some of the papers yield unsatisfactory results needs to be undertaken in future studies.

A comparison of the machine learning technique (Linear Regression) developed in Chapter 8 and the Information Retrieval techniques developed in Chapter 9 is provided in Section 9.7.5. The information retrieval technique Probabilistic BM25 performs more poorly than Linear Regression in all experimental settings when considering *Precision@k* and *NDCG@k* scores. In all other experimental settings with one exception, all of the information retrieval models (Vector Space Model, Information Based System, Divergence From Randomness, Language Model with Jelinek-Mercer smoothing, and Language Model with Dirichlet priors smoothing (LMD)) perform better than the Linear Regression model when considering *Precision@k* and *NDCG@k* scores. None of the differences in these scores are statistically significant, however.

10.6 Future Work

From this study, we can identify other research areas presented in the following sections.

10.6.1 Build a Citation Sentence Framework for other IMRaD Categories

The focus of this study was to link citation sentences which belong to the "Method" category with the sentences in the cited paper. We would like to extend the study to citation sentences that refer to other IMRaD categories such as "Result" and "Discussion". While we show the feasibility of the citation linkage operation at a more manageable scope, we acknowledge that a citation context encompasses more than methodology terms, and a more complete linkage system should take into account various citation categories.

10.6.2 Build Lexical Resources from Method Mention Context

The techniques used to create the "Method mention" corpora in Chapter 3 enable us to collect method keywords and the context of their usage. It would be useful to devise a methodology to extract relations between method terms as presented in their surrounding text. This can be useful in dictionary building, ontology population and glossary creation.

10.6.3 Build Linkage Corpora in Other Domains

Our linkage corpus was built with papers from the Biomedical domain. We would like future research to focus on designing annotation schemes for papers in other domains as well, such as Computer Science, Chemistry, etc. We believe that the linkage task can be applied to articles from many research domains as long as there is a citation network between articles. But this will certainly imply new experimental settings that will bring new difficulties pertaining to the domain of study. However the study will be worth considering when time and resources permit. For instance we can target the ACL anthology network corpus [149], that is a comprehensive manually curated networked database of citations, collaborations, and summaries in the field of Computational Linguistics. Citation linkage features can be added to the database. This will involve further study of the argumentation structure of the computational linguistics articles and the multi-level annotation of the linkage corpus.

10.6.4 Experiments with Other Machine Learning Techniques

The machine learning experiments studied in Chapter 8 used only linear regression techniques. Investigating the use of non-linear kernels for the Support Vector Machine technique would be a rather straightforward step. Other machine learning techniques, such as induced decision trees, random forests, and artificial neural networks, could also be studied.

10.6.5 Experiments with Combined Techniques

The machine learning methods in Chapter 8 have used a hybrid approach utilizing text similarity measures to compute a likely match between the citation sentence and the target sentences. The information retrieval techniques that have been investigated in Chapter 9 analyze the target text vis a vis the query (the citation sentence or the noun phrases contained in the citation sentence). This analysis takes into account the differences among the target sentences.

Since the machine learning methods view the hybrid approach as a set of performance scores from a set of experts that rate the likelihood of the citation sentence referring to a target sentence, adding performance scores given by the informatin retriveal techniques to the scores given by the text similarity techniques is a straightforward modification to the scores provided to the machine learning methods.

10.6.6 Experiments with Other Text Granularity

At this stage, we have learned that we could not rely only on currently available text similarity measures to build learning models to detect the degree of similarity between a citation

sentence and the equivalent sentences in the cited paper. It will be interesting to investigate whether dividing the articles into a broader granularity can alleviate the problem. This study has focused on sentence-to-sentence matching, because most text applications are primarily concerned with individual sentence units. However a citation context can span many sentences and consequently, the target text can comprise more than one sentence. For this purpose, we will need to implement different annotation guidelines and the multi-level similarity degree might not be achievable. However the information needs that we try to resolve through the linkage task might be realized for some users who might require a broader context to grasp the meaning of the retrieved text.

10.6.7 Interactive Application For Citation Linkage

With the current investigation, we have shown the feasibility of the citation linkage operation as a means to provide domain specific information to an end user. We would like to develop a Web application for automatic citation linkage of scientific research papers. An interactive user interface can be used to present different sentences and the confidence values for their selection. This will involve the use of Asynchronous JavaScript and XML (AJAX) technologies in combination with other visualization tools. We will need to expand our knowledge of such technologies to be able to provide a web service that will be useful to the research community.

Bibliography

[1] Shashank Agarwal and Hong Yu. Automatically classifying sentences in full-text biomedical articles into Introduction, Methods, Results and Discussion. *Bioinformatics*, 25(23):3174–3180, 2009.

[2] Charu C Aggarwal and ChengXiang Zhai. A survey of text classification algorithms. In *Mining Text Data*, pages 163–222. Springer, 2012.

[3] Nitish Aggarwal, Kartik Asooja, and Paul Buitelaar. DERI&UPM: Pushing corpus based relatedness to similarity: Shared task system description. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation)*, pages 643–647. Association for Computational Linguistics, 2012.

[4] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. Sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In *The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics (SEM 2013)*, 2013.

[5] Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 385–393, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[6] Ethem Alpaydin. *Introduction to Machine Learning*. MIT Press, 2004.

[7] Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)*, 20(4):357–389, 2002.

[8] Ion Androutsopoulos and Prodromos Malakasiotis. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187, 2010.

[9] MA Angrosh, Stephen Cranefield, and Nigel Stanger. Context identification of sentences in research articles: Towards developing intelligent tools for the research community. *Natural Language Engineering*, 19(04):481–515, 2013.

[10] Satanjeev Banerjee and Ted Pedersen. An adapted Lesk algorithm for word sense disambiguation using WordNet. In *Computational Linguistics and Intelligent Text Processing*, pages 136–145. Springer, 2002.

[11] Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. UKP: Computing semantic textual similarity by combining multiple content similarity measures. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation)*, pages 435–440. Association for Computational Linguistics, 2012.

[12] Alberto Barrón-Cedeno, Paolo Rosso, Eneko Agirre, and Gorka Labaka. Plagiarism detection across distant language pairs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 37–45. Association for Computational Linguistics, 2010.

[13] Oliver Bender, Franz Josef Och, and Hermann Ney. Maximum entropy models for named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003-Volume 4*, pages 148–151. Association for Computational Linguistics, 2003.

[14] Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[15] Delphine Bernhard and Iryna Gurevych. Answering learners' questions by retrieving question paraphrases from social Q&A sites. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 44–52. Association for Computational Linguistics, 2008.

[16] Andrzej Białecki, Robert Muir, and Grant Ingersoll. Apache Lucene 4. In *SIGIR 2012 Workshop on Open Source Information Retrieval*, page 17, 2012.

[17] Olivier Bodenreider. Lexical, terminological and ontological resources for biological text mining. *Text Mining for Biology and Biomedicine*, pages 43–66, 2006.

[18] Andrew Borthwick. *A maximum entropy approach to named entity recognition*. PhD thesis, New York University, 1999.

[19] Johan Bos and Katja Markert. Recognising textual entailment with logical inference. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 628–635. Association for Computational Linguistics, 2005.

[20] Chris Brockett and William B Dolan. Support vector machines for paraphrase identification and corpus construction. In *Proceedings of the 3rd International Workshop on Paraphrasing*, pages 1–8, 2005.

[21] Terrence A Brooks. Private acts and public objects: An investigation of citer motives. *J. Am. Soc. Inf. Sci.*, 36(4):223–229, July 1985.

[22] Davide Buscaldi, Ronan Tournier, Nathalie Aussenac-Gilles, and Josiane Mothe. IRIT: Textual similarity combining conceptual similarity with an n-gram comparison method. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation)*, pages 552–556. Association for Computational Linguistics, 2012.

[23] Addeane S Caelleigh. Pubmed central and the new publishing landscape: shifts and tradeoffs. *Academic Medicine*, 75(1):4–10, 2000.

[24] Chris Callison-Burch, Philipp Koehn, and Miles Osborne. Improved statistical machine translation using paraphrases. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 17–24. Association for Computational Linguistics, 2006.

[25] David Campos, José Luís Oliveira, and Sérgio Matos. *Biomedical Named Entity Recognition: A Survey of Machine-Learning Tools*. INTECH Open Access Publisher, 2012.

[26] Jingnian Chen, Houkuan Huang, Shengfeng Tian, and Youli Qu. Feature selection for text classification with naïve bayes. *Expert Systems with Applications*, 36(3):5432–5435, 2009.

[27] Boris Chidlovskii and Loic Lecerf. Scalable feature selection for multi-class problems, September 6 2011. US Patent 8,015,126.

[28] Hinrich Schütze Christopher D. Manning, Prabhakar Raghavan. *Introduction to Information Retrieval*. Cambridge University Press, 2009.

[29] Grace Y Chung. Sentence retrieval for abstracts of randomized controlled trials. *BMC Medical Informatics and Decision Making*, 9(1):10, 2009.

[30] Rudi L Cilibrasi and Paul MB Vitanyi. The google similarity distance. *Knowledge and Data Engineering, IEEE Transactions on*, 19(3):370–383, 2007.

[31] Daoud Clarke. Context-theoretic semantics for natural language: an overview. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 112–119. Association for Computational Linguistics, 2009.

[32] Stéphane Clinchant and Eric Gaussier. Information-based models for ad hoc IR. In *Proceedings of the 33rd international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 234–241. ACM, 2010.

[33] Arman Cohan and Luca Soldaini. Towards citation-based summarization of biomedical literature. In *Proceedings of the Text Analysis Conference (TAC 2014)*, 2014.

[34] Arman Cohan, Luca Soldaini, and Nazli Goharian. Matching citation text and cited spans in biomedical literature: A search-oriented approach. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL (NAACL HLT 2015)*, 2015.

[35] Allan M Collins and Elizabeth F Loftus. A spreading-activation theory of semantic processing. *Psychological Review*, 82(6):407, 1975.

[36] James R Curran and Stephen Clark. Language independent NER using a maximum entropy tagger. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003-Volume 4*, pages 164–167. Association for Computational Linguistics, 2003.

[37] James R Curran, Stephen Clark, and Johan Bos. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 33–36. Association for Computational Linguistics, 2007.

[38] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL Recognising Textual Entailment Challenge. In Joaquin Quiñonero Candela, Ido Dagan, Bernardo Magnini,

and Florence d'Alché Buc, editors, *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer Berlin Heidelberg, 2006.

[39] Anita de Waard, Paul Buitelaar, and Thomas Eigner. Identifying the epistemic value of discourse segments in biology texts. In *Proceedings of the Eighth International Conference on Computational Semantics*, pages 351–354. Association for Computational Linguistics, 2009.

[40] Reseivei December. A note on uniform distribution and experimental design. *KeXue TongBao*, 1981.

[41] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[42] Bill Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356. Association for Computational Linguistics, 2004.

[43] William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of Third International Workshop on Paraphrasing (IWP2005)*, pages 9–116, 2005.

[44] Myroslava O Dzikovska, Johanna D Moore, Natalie Steinhauser, Gwendolyn Campbell, Elaine Farrow, and Charles B Callaway. Beetle II: A system for tutoring and computational linguistics experimentation. In *Proceedings of the ACL 2010 System Demonstrations*, pages 13–18. Association for Computational Linguistics, 2010.

[45] Myroslava O Dzikovska, Rodney D Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa T Dang. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. Technical report, DTIC Document, 2013.

[46] Sean R Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996.

[47] Katrin Erk and Sebastian Padó. Paraphrase assessment in structured vector space: Exploring parameters and datasets. In *Proceedings of the Workshop on Geometrical Models*

of Natural Language Semantics, pages 57–65. Association for Computational Linguistics, 2009.

[48] Hui Fang, Tao Tao, and ChengXiang Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–56, 2004.

[49] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.

[50] Katherine Forbes, Eleni Miltsakaki, Rashmi Prasad, Anoop Sarkar, Aravind Joshi, and Bonnie Webber. D-LTAG system: Discourse parsing with a lexicalized tree-adjoining grammar. *Journal of Logic, Language and Information*, 12(3):261–279, 2003.

[51] W Nelson Francis and Henry Kucera. Brown corpus manual. *Brown University*, 15, 1979.

[52] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using Wiikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.

[53] Eugene Garfield. Science Citation Index—a new dimension in indexing. *Science*, 144(3619):649–654, 1964.

[54] Eugene Garfield. Citation analysis as a tool in journal evaluation. *Science*, 178(4060):471–479, 1972.

[55] Eugene Garfield. Is citation analysis a legitimate evaluation tool? *Scientometrics*, 1(4):359–375, 1979.

[56] Mark Garzone and Robert E. Mercer. Towards an automated citation classifier. In Howard J. Hamilton, editor, *Advances in Artificial Intelligence*, volume 1822 of *Lecture Notes in Computer Science*, pages 337–346. Springer Berlin Heidelberg, 2000.

[57] Mark Arthur Garzone. Automated classification of citations using linguistic semantic grammars. Master's thesis, The University of Western Ontario, London, Ontario, 1997.

[58] Andrew Gelman and Jennifer Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press, 2006.

[59] Eva Geuens, David Hoogewijs, Marco Nardini, Evi Vinck, Alessandra Pesce, Laurent Kiger, Angela Fago, Lesley Tilleman, Sasha De Henau, Michael C Marden, et al.

Globin-like proteins in caenorhabditis elegans: in vivo localization, ligand binding and structural properties. *BMC Biochemistry*, 11(1):17, 2010.

[60] Corrado Gini. Measurement of inequality of incomes. *The Economic Journal*, 31(121):124–126, 1921.

[61] Wael H. Gomaa and Aly A. Fahmy. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18, April 2013.

[62] Yufan Guo, Anna Korhonen, Maria Liakata, Ilona Silins Karolinska, Lin Sun, and Ulla Stenius. Identifying the information structure of scientific abstracts: An investigation of three different schemes. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 99–107. Association for Computational Linguistics, 2010.

[63] Dan Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University press, 1997.

[64] Aria D Haghighi, Andrew Y Ng, and Christopher D Manning. Robust textual inference via graph matching. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 387–394. Association for Computational Linguistics, 2005.

[65] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[66] Patrick A. V. Hall and Geoff R. Dowling. Approximate string matching. *ACM Comput. Surv.*, 12(4):381–402, December 1980.

[67] Kazuo Hara and Yuji Matsumoto. Information extraction and sentence classification applied to clinical trial medline abstracts. In *Proceedings of the 2005 International Joint Conference of InCoB, AASBi and KSBI*, pages 85–90, 2005.

[68] Stephen P Harter. A probabilistic approach to automatic keyword indexing. part ii. an algorithm for probabilistic indexing. *Journal of the American Society for Information Science*, 26(5):280–289, 1975.

[69] W. E. Hick. On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, 4(1):11–26, 1952.

[70] Graeme Hirst and David St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 305–332. MIT Press, 1998.

[71] David Hoogewijs, Koen Houthoofd, Filip Matthijssens, Jo Vandesompele, and Jacques R Vanfleteren. Selection and validation of a set of reliable reference genes for quantitative sod gene expression analysis in c. elegans. *BMC Molecular Biology*, 9(1):1, 2008.

[72] Hospice Houngbo and Robert E. Mercer. Method mention extraction from scientific research papers. In *Proceedings of COLING 2012*, pages 1211–1222, Mumbai, India, December 2012. The COLING 2012 Organizing Committee.

[73] Hospice Houngbo and Robert E. Mercer. An automated method to build a corpus of rhetorically-classified sentences in biomedical texts. In *Proceedings of the First Workshop on Argumentation Mining*, pages 19–23, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[74] Susan Hunston. Starting with the small words patterns, lexis and semantic sequences. *International Journal of Corpus Linguistics*, 13(3):271–295, 2008.

[75] Fidelia Ibekwe-SanJuan, Chaomei Chen, Pinho Roberto, et al. Identifying strategic information from scientific articles through sentence classification. In *6th International Conference on Language Resources and Evaluation Conference (LREC 2008)*, pages 1518–1522, 2008.

[76] Aminul Islam and Diana Inkpen. Second order co-occurrence pmi for determining the semantic similarity of words. In *Proceedings of the International Conference on Language Resources and Evaluation, Genoa, Italy*, pages 1033–1038, 2006.

[77] Aminul Islam and Diana Inkpen. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data*, 2(2):Article No. 10, 2008.

[78] Paul Jaccard. *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz, 1901.

[79] Matthew A Jaro. Probabilistic linkage of large public health data files. *Statistics in Medicine*, 14(5-7):491–498, 1995.

[80] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[81] Jay J Jiang and David W Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*, 1997.

[82] Hans Kamp, Josef Van Genabith, and Uwe Reyle. Discourse representation theory. In *Handbook of Philosophical Logic*, pages 125–394. Springer, 2011.

[83] S Sathiya Keerthi, Chong Jin Ong, Keng Boon Siah, David BL Lim, Wei Chu, Min Shi, David S Edwin, Rakesh Menon, Lixiang Shen, Jonathan YK Lim, et al. A machine learning approach for the curation of biomedical literature: Kdd cup 2002 (task 1). *ACM SIGKDD Explorations Newsletter*, 4(2):93–94, 2002.

[84] Halil Kilicoglu and Sabine Bergler. Recognizing speculative language in biomedical research articles: A linguistically motivated perspective. *BMC Bioinformatics*, 9(11):1, 2008.

[85] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, pages 70–75. Association for Computational Linguistics, 2004.

[86] Su N Kim, David Martinez, Lawrence Cavedon, and Lars Yencken. Automatic classification of sentences to support evidence based medicine. *BMC Bioinformatics*, 12(Suppl 2):S5, 2011.

[87] Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003.

[88] David G Kleinbaum and Mitchel Klein. Ordinal logistic regression. In *Logistic Regression*, pages 463–488. Springer, 2010.

[89] Eugene F Krause. Taxicab geometry: an adventure in non-euclidean geometry, volume viii, 1987.

[90] Sandra Kübler, Ryan McDonald, and Joakim Nivre. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 1(1):1–127, 2009.

[91] Miron B Kursa, Witold R Rudnicki, et al. Feature selection with the boruta package, 2010.

[92] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[93] Thomas K Landauer and Susan T Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211, 1997.

[94] Hagen Langer, Harald Lüngen, and Petra Saskia Bayerl. Text type structure and logical document structure. In *Proceedings of the 2004 ACL Workshop on Discourse Annotation*, DiscAnnotation '04, pages 49–56, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[95] Alex Lascarides and Nicholas Asher. Segmented discourse representation theory: Dynamic semantics with discourse structure. In *Computing Meaning*, pages 87–124. Springer, 2007.

[96] Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1536–1545. Association for Computational Linguistics, 2011.

[97] Claudia Leacock and Martin Chodorow. Combining local context and WordNet similarity for word sense identification. *WordNet: An Electronic Lexical Database*, 49(2):265–283, 1998.

[98] Geoffrey Leech, Paul Rayson, and Andrew Wilson. *Word frequencies in written and spoken English: Based on the British National Corpus*. Routledge, 2014.

[99] Yuhua Li, David McLean, Zuhair A Bandar, James D O'Shea, and Keeley Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138–1150, 2006.

[100] Maria Liakata. Zones of conceptualisation in scientific papers: A window to negative and speculative statements. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pages 1–4. Association for Computational Linguistics, 2010.

[101] Maria Liakata, Shyamasree Saha, Simon Dobnik, Colin R. Batchelor, and Dietrich Rebholz-Schuhmann. Automatic recognition of conceptualization zones in scientific articles and two life science applications. *Bioinformatics*, 28(7):991–1000, 2012.

[102] Maria Liakata, Simone Teufel, Advaith Siddharthan, and Colin R Batchelor. Corpora for the conceptualisation and zoning of scientific papers. In *Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, pages 2054–2016, 2010.

[103] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[104] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98)*, pages 296–304, 1998.

[105] Mengxiong Liu. Progress in documentation the complexities of citation practice: A review of citation studies. *Journal of Documentation*, 49(4):370–408, 1993.

[106] Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208, 1996.

[107] Kevin Lund, Curt Burgess, and Ruth Ann Atchley. Semantic and associative priming in high-dimensional semantic space. In *Proceedings of the 17th Annual Conference of the Cognitive Science Society*, volume 17, pages 660–665, 1995.

[108] Christopher D Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[109] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

[110] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[111] Lluís Màrquez, Xavier Carreras, Kenneth C Litkowski, and Suzanne Stevenson. Semantic role labeling: An introduction to the special issue. *Computational Linguistics*, 34(2):145–159, 2008.

[112] Irina Matveeva, Gina-Anne Levow, Ayman Farahat, and Christian Royer. Generalized latent semantic analysis for term representation. In *Proc. of RANLP*, 2005.

[113] Diana Maynard and Sophia Ananiadou. TRUCKS: A model for automatic multi-word term recognition. *Journal of Natural Language Processing*, 8(1):101–125, 2000.

[114] Diana Maynard, Valentin Tablan, Cristian Ursu, Hamish Cunningham, and Yorick Wilks. Named entity recognition from diverse text types. In *Recent Advances in Natural Language Processing 2001 Conference*, pages 257–274, Tzigov Chark, 2001.

[115] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for Naïve Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48, 1998.

[116] Philip M McCarthy, Rebekah H Guess, and Danielle S McNamara. The components of paraphrase evaluations. *Behavior Research Methods*, 41(3):682–690, 2009.

[117] Larry McKnight and Padmini Srinivasan. Categorization of sentence types in medical abstracts. In *AMIA Annual Symposium Proceedings*, page 440. American Medical Informatics Association, 2003.

[118] Ben Medlock and Ted Briscoe. Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, volume 2007, pages 992–999, 2007.

[119] Igor Mel'cuk. *Dependency Syntax: Theory and Practice*. SUNY series in Linguistics. State University of New York Press, Albany, New York, USA, 1988.

[120] Robert Mercer. Locating and extracting key components of argumentation from scholarly scientific writing. In *Natural Language Argumentation: Mining, Processing, and Reasoning over Textual Arguments (Dagstuhl Seminar 16161)*, volume 6(4) #3.15, page 93, 2016.

[121] Robert E. Mercer and Chrysanne Di Marco. The importance of fine-grained cue phrases in scientific citations. In Yang Xiang and Brahim Chaib-draa, editors, *Advances in Artificial Intelligence*, volume 2671 of *Lecture Notes in Computer Science*, pages 550–556. Springer Berlin Heidelberg, 2003.

[122] Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1 (AAAI'06)*, pages 775–780, 2006.

[123] George A Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[124] George A Miller and Edwin B Newman. Tests of a statistical explanation of the rank-frequency relation for words in written English. *The American Journal of Psychology*, 71(1):209–218, 1958.

[125] Jeff Mitchell and Mirella Lapata. Vector-based models of semantic composition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 236–244, 2008.

[126] Tom M Mitchell. *Machine Learning*. McGraw Hill, 1997.

[127] Yoko Mizuta and Nigel Collier. An annotation scheme for a rhetorical analysis of biology articles. In *Language Resources Evaluation Conference (LREC)*, pages 1737–1740, 2004.

[128] Yoko Mizuta and Nigel Collier. Zone identification in biology articles as a basis for information extraction. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, JNLPBA '04, pages 29–35, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[129] Yoko Mizuta, Anna Korhonen, Tony Mullen, and Nigel Collier. Zone analysis in biology articles as a basis for information extraction. *International Journal of Medical Informatics*, 75(6):468–487, 2006.

[130] Dan I Moldovan and Vasile Rus. Logic form transformation of Wordnet and its applicability to question answering. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 402–409. Association for Computational Linguistics, 2001.

[131] Alvaro Monge and Charles Elkan. An efficient domain-independent algorithm for detecting approximately duplicate database records. In *Workshop on Research Issues on Data Mining and Knowledge Discovery*, 1997.

[132] M. J. Moravcsik and P. Murugesan. Some results on the function and quality of citations. *Social Studies of Science*, 5:86–92, 1975.

[133] Tony Mullen, Yoko Mizuta, and Nigel Collier. A baseline feature set for learning rhetorical zones using full articles in the biomedical domain. *ACM SIGKDD Explorations Newsletter*, 7(1):52–58, 2005.

[134] Preslav I. Nakov, Ariel S. Schwartz, and Marti A. Hearst. Citances: Citation sentences for semantic analysis of bioscience text. In *Proceedings of the SIGIR'04 Workshop on Search and Discovery in Bioinformatics*, pages 81–88, 2004.

[135] Hidetsugu Nanba and Manabu Okumura. Towards multi-paper summarization reference information. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'99, pages 926–931, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[136] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[137] Rodney D Nielsen, Wayne Ward, James H Martin, and Martha Palmer. Annotating students' understanding of science concepts. In *Language Resources Evaluation Conference (LREC)*, 2008.

[138] Sebastian Padó and Mirella Lapata. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199, 2007.

[139] Raquel Mochales Palau and Marie-Francine Moens. Argumentation mining: The detection, classification and structure of arguments in text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 98–107. ACM, 2009.

[140] David D. Palmer and David S. Day. A statistical profile of the Named Entity task. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, ANLC '97, pages 190–193. Association for Computational Linguistics, 1997.

[141] R Panico, WH Powell, and Jean-Claude Richer. *A guide to IUPAC Nomenclature of Organic Compounds*. Blackwell Science, 1995.

[142] Siddharth Patwardhan. *Incorporating dictionary and corpus information into a context vector measure of semantic relatedness*. PhD thesis, University of Minnesota, Duluth, 2003.

[143] Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Computational Linguistics and Intelligent Text Processing*, pages 241–257. Springer, 2003.

[144] Ted Pedersen. Information content measures of semantic similarity perform better without sense-tagged text. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 329–332. Association for Computational Linguistics, 2010.

[145] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet::similarity - measuring the relatedness of concepts. In *Demonstration papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics, 2004.

[146] James L. Peterson. Computer programs for detecting and correcting spelling errors. *Communications of the ACM*, 23(12):676–687, 1980.

[147] Martin Potthast, Benno Stein, and Maik Anderka. A Wikipedia-based multilingual retrieval model. In *Advances in Information Retrieval*, pages 522–530. Springer, 2008.

[148] Dragomir R Radev, Hongyan Jing, and Malgorzata Budzikowska. Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization*, pages 21–30. Association for Computational Linguistics, 2000.

[149] Dragomir R Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. The ACL anthology network corpus. *Language Resources and Evaluation*, 47(4):919–944, 2013.

[150] Dietrich Rebholz-Schuhmann, Antonio José Jimeno Yepes, Erik M Van Mulligen, Ning Kang, Jan Kors, David Milward, Peter Corbett, Ekaterina Buyko, Elena Beisswanger, and Udo Hahn. CALBC silver standard corpus. *Journal of Bioinformatics and Computational Biology*, 8(1):163–179, 2010.

[151] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.

[152] Anna Ritchie. *Citation context analysis for information retrieval*. PhD thesis, University of Cambridge, 2009.

[153] Stephen Robertson and Hugo Zaragoza. *The Probabilistic Relevance Framework: BM25 and Beyond*. Now Publishers Inc., 2009.

[154] V Rus, M Lintean, C Moldovan, W Baggett, N Niraula, and B Morgan. The SIMILAR corpus: A resource to foster the qualitative understanding of semantic similarity of texts.

In *Semantic Relations II: Enhancing Resources and Applications, The 8th Language Resources and Evaluation Conference (LREC 2012)*, pages 50–59, 2012.

[155] Stephan R Sain. The nature of statistical learning theory. *Technometrics*, 38(4):409–409, 1996.

[156] Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. Takelab: Systems for measuring semantic text similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation)*, pages 441–448. Association for Computational Linguistics, 2012.

[157] Stanley M Selkow. The tree-to-tree editing problem. *Information Processing Letters*, 6(6):184–186, 1977.

[158] Claude E Shannon and Warren Weaver. *The Mathematical Theory of Information*. University of Illinois Press, 1949.

[159] Hagit Shatkay, Fengxia Pan, Andrey Rzhetsky, and W John Wilbur. Multi-dimensional classification of biomedical text: Toward automated, practical provision of high-utility text to diverse users. *Bioinformatics*, 24(18):2086–2093, 2008.

[160] Jorge Silva, Ana Aguiar, and Fernando Silva. A parallel computing hybrid approach for feature selection. In *IEEE 18th International Conference on Computational Science and Engineering (CSE 2015)*, pages 97–104. IEEE, 2015.

[161] H Small. Citation context analysis. *Progress in Communication Sciences*, 3:287–310, 1982.

[162] Henry Small. Cited documents as concept symbols. *Social Studies of Science*, 7:113–22, 1978.

[163] Larry Smith, Lorraine K Tanabe, Rie J Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, et al. Overview of BioCreative II gene mention recognition. *Genome biology*, 9(Suppl 2):S2, 2008.

[164] Temple F Smith and Michael S Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[165] Robert R Sokal. A statistical method for evaluating systematic relationships. *Univ Kans Sci Bull*, 38:1409–1438, 1958.

[166] Larisa Soldatova and Maria Liakata. An ontology methodology and CISP - the proposed Core Information about Scientific Papers. In *JISC Project Report*. JISC collections, 2007.

[167] Dan Stefuanescu, Rajendra Banjade, and Vasile Rus. Latent semantic analysis models on Wikipedia and TASA. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation Conference (LREC-2014)*, pages 1417–1422, 2014.

[168] John Swales. Citation analysis and discourse analysis. *Applied Linguistics*, 7(1):39–56, 1986.

[169] Kuo-Chung Tai. The tree-to-tree correction problem. *Journal of the ACM (JACM)*, 26(3):422–433, 1979.

[170] Howard M Taylor and Samuel Karlin. *An Introduction to Stochastic Modeling*. Academic press, 2014.

[171] Simone Teufel. *Argumentative zoning: Information extraction from scientific texts*. PhD thesis, School of Cognitive Science, University of Edinburgh, UK, 1999.

[172] Simone Teufel and Marc Moens. Argumentative classification of extracted sentences as a first step towards flexible abstracting. In *Advances in Automatic Text Summarization*, pages 155–171. MIT Press, 1999.

[173] Simone Teufel, Advaith Siddharthan, and Colin Batchelor. 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*, EMNLP '09, pages 1493–1502. Association for Computational Linguistics, 2009.

[174] Simone Teufel, Advaith Siddharthan, and Dan Tidhar. Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP '06)*, pages 103–110. Association for Computational Linguistics, 2006.

[175] Katrin Tomanek, Joachim Wermter, and Udo Hahn. A reappraisal of sentence and token splitting for life sciences documents. In *MEDINFO 2007*, volume 129 of *Studies in Health Technology and Informatics*, pages 524–528. IOS Press, 2007.

[176] M. Torii and K. Vijay-Shanker. Anaphora resolution of demonstrative noun phrases in Medline abstracts. In *Procedings of PACLING 2005*, pages 332–339, 2005.

[177] Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. Developing a robust part-of-speech tagger for biomedical text. In Panayiotis Bozanis and Elias N. Houstis, editors, *Advances in Informatics: 10th Panhellenic Conference on Informatics (PCI 2005)*, volume 3476 of *Lecture Notes in Computer Science*, pages 382–392. Springer, 2005.

[178] Peter Turney. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning (ECML '01)*, pages 491–502, 2001.

[179] Vladimir Naumovich Vapnik. *Statistical Learning Theory*. Wiley New York, 1998.

[180] Alexei Vinokourov. *The organisation and retrieval of document collections: A machine learning approach*. PhD thesis, University of Paisley, UK, 2003.

[181] W John Wilbur, Andrey Rzhetsky, and Hagit Shatkay. New directions in biomedical text annotation: Definitions, guidelines and corpus construction. *BMC Bioinformatics*, 7:356–365, 2006.

[182] Shatkay H. Wilbur WJ, Rzhetsky A. New directions in biomedical text annotation: definitions, guidelines and corpus construction. *Text*, 7(356), 2006.

[183] Mann William and Sandra Thompson. Rhetorical structure theory: Towards a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[184] William E Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research*, 1990.

[185] William E Winkler. The state of record linkage and current research problems. In *Statistical Research Division, US Census Bureau*, 1999.

[186] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

[187] Yasunori Yamamoto and Toshihisa Takagi. A sentence classification system for multi biomedical literature summarization. In *21st International Conference on Data Engineering Workshops*, pages 1163–1163. IEEE, 2005.

[188] Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 129–136. Association for Computational Linguistics, 2003.

[189] ChengXiang Zhai. Statistical language models for information retrieval. *Synthesis Lectures on Human Language Technologies*, 1(1):1–141, 2008.

[190] ChengXiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 334–342. ACM, 2001.

[191] Chengzhi Zhang, Huilin Wang, Yao Liu, Dan Wu, Yi Liao, and Bo Wang. Automatic keyword extraction from documents using Conditional Random Fields. *Journal of Computational Information Systems*, 4(3):1169–1180, 2008.

[192] Chengzhi Zhang and Dan Wu. Bilingual terminology extraction using multi-level termhood. *The Electronic Library*, 30(2):295–309, 2012.

[193] Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6):1245–1262, 1989.

[194] Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 825–832. Association for Computational Linguistics, 2006.

[195] Yitao Zhang and Jon Patrick. Paraphrase identification by text canonicalization. In *Proceedings of the Australasian Language Technology Workshop*, pages 160–166, 2005.

[196] Guodong Zhou, Jie Zhang, Jian Su, Dan Shen, and Chewlim Tan. Recognizing names in biomedical texts: A machine learning approach. *Bioinformatics*, 20:1178–1190, 2004.

[197] Liang Zhou, Chin-Yew Lin, and Eduard Hovy. Re-evaluating machine translation results with paraphrase support. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 77–84. Association for Computational Linguistics, 2006.

Appendix A

List of Datasets

A.1 List of Articles

The following is a list of the cited papers used in this study.

Article #	Article Title	Article Link
1	Selection and validation of a set of reliable reference genes for quantitative sod gene expression analysis in C. elegans	http://bmcmolbiol.biomedcentral.com/articles/10.1186/1471-2199-9-9
2	SUMO-1 possesses DNA binding activity	http://bmcresnotes.biomedcentral.com/articles/10.1186/1756-0500-3-146
3	Mitochondrial activities in human cultured skin fibroblasts contaminated by Mycoplasma hyorhinis	http://bmcbiochem.biomedcentral.com/articles/10.1186/1471-2091-4-15
4	Tying the loose ends together in DNA double strand break repair with 53BP1	https://celldiv.biomedcentral.com/articles/10.1186/1747-1028-1-19
5	Pro-protein convertases control the maturation and processing of the iron-regulatory protein, RGMc/hemojuvelin	http://bmcbiochem.biomedcentral.com/articles/10.1186/1471-2091-9-9

6	Targeted silencing of Jab1/Csn5 in human cells downregulates SCF activity through reduction of F-box protein levels	http://bmcbiochem.biomedcentral.com/articles/10.1186/1471-2091-7-1
7	Tracking the connection between evolutionary and functional shifts using the fungal lipase/feruloyl esterase A family	http://bmcevolbiol.biomedcentral.com/articles/10.1186/1471-2148-6-92
8	Analysis of DNA relaxation and cleavage activities of recombinant Mycobacterium tuberculosis DNA topoisomerase I from a new expression and purification protocol	http://bmcbiochem.biomedcentral.com/articles/10.1186/1471-2091-10-18
9	Generating rate equations for complex enzyme systems by a computer-assisted systematic method	http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-10-238
10	Identification of ATP binding residues of a protein from its primary sequence	http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-10-434
11	The RIN: an RNA integrity number for assigning integrity values to RNA measurements	http://bmcmolbiol.biomedcentral.com/articles/10.1186/1471-2199-7-3
12	Characterization of a synthetic human LINE-1 retrotransposon ORFeus-Hs	https://mobilednajournal.biomedcentral.com/articles/10.1186/1759-8753-2-2
13	Accelerated exchange of exon segments in Viperid three-finger toxin genes (Sistrurus catenatus edwardsii; Desert Massasauga)	http://bmcevolbiol.biomedcentral.com/articles/10.1186/1471-2148-8-196
14	Generation of medaka gene knockout models by target-selected mutagenesis	https://genomebiology.biomedcentral.com/articles/10.1186/gb-2006-7-12-r116

15	Implementation of two high through-put techniques in a novel application: detecting point mutations in large EMS mutated plant populations	https://plantmethods.biomedcentral.com/articles/10.1186/1746-4811-5-13
16	Quantification of mRNA in single cells and modelling of RT-qPCR induced noise	http://bmcmolbiol.biomedcentral.com/articles/10.1186/1471-2199-9-63
17	Prediction and identification of Arabidopsis thaliana microRNAs and their mRNA targets	https://genomebiology.biomedcentral.com/articles/10.1186/gb-2004-5-9-r65
18	Comparative analysis of the Saccharomyces cerevisiae and Caenorhabditis elegans protein interaction networks	http://bmcevolbiol.biomedcentral.com/articles/10.1186/1471-2148-5-23
19	Bringing metabolic networks to life: convenience rate law and thermodynamic constraints	https://tbiomed.biomedcentral.com/articles/10.1186/1742-4682-3-41
20	Prediction of mono- and di-nucleotide-specific DNA-binding sites in proteins using neural networks	http://bmcstructbiol.biomedcentral.com/articles/10.1186/1472-6807-9-30
21	Identifying biological themes within lists of genes with EASE	https://genomebiology.biomedcentral.com/articles/10.1186/gb-2003-4-10-r70
22	High content image analysis for human H4 neuroglioma cells exposed to CuO nanoparticles	http://bmcbiotechnol.biomedcentral.com/articles/10.1186/1472-6750-7-66

Table A.1: Articles Used in the Study

A.2 List of Datasets

The following are the datasets used in this study. The first corpus consists of the sentences tagged with its rhetorical category. The second dataset comprises the 22 annotated papers used in this study.

Self-Annotating Rhetorical Corpora
`https://github.com/hospice/rethorical_self_annotating_corpora`

Citation Linkage Dataset
`https://github.com/hospice/linkagefiles`

Appendix B

Example of a Full Paper Annotation

The following is an example of the annotation of a complete cited paper (Paper #8) by one of our annotators. Each sentence is presented with a rating: The rating 0 means that the annotator does not believe that this sentence is being cited by the citation. The ratings 1–5 are the ratings of confidence (see the instructions given to the annotators in Appendix C) given by the annotator for those sentences thought to be cited. The annotations for all of the papers in our study are available at https://github.com/hospice/linkagefiles.

Title of Cited Paper #8: Analysis of DNA relaxation and cleavage activities of recombinant Mycobacterium tuberculosis DNA topoisomerase I from a new expression and purification protocol
Link: http://bmcbiochem.biomedcentral.com/ articles/10.1186/1471-2091-10-18

Citation Sentence: Expression and purification procedures utilizing plasmid pLIC-MTOP have been developed to produce milligram quantities of soluble and active MtTOP1 protein.

Title of Citing Paper: The DNA relaxation activity and covalent complex accumulation of Mycobacterium tuberculosis topoisomerase I can be assayed in Escherichia coli: application for identification of potential FRET-dye labeling sites

Annotation of Paper #8:

Number	Sentence	Rating
1	Background	0
2	DNA topoisomerases are ubiquitous enzymes involved in the regulation of DNA supercoiling and overcoming topological barriers during replication transcription recombination and repair.	0

3	In bacteria the major classes of topoisomerases type IA and type IIA modify DNA topology by transiently cleaving and rejoining one or two strands of DNA respectively .	0
4	Both of these classes form a 5'-phosphotyrosyl enzyme-DNA linkage during the catalytic cycle of DNA cleavage and religation .	0
5	Topoisomerases are attractive targets for development of new anti-infectives .	0
6	Bacterial DNA gyrase and topoisomerase IV from the type IIA class are targets of antibiotics such as quinolones and fluoroquinolones.	0
7	These antibiotics exhibit their bactericidal properties by trapping the covalent protein-DNA complexes formed by DNA gyrase or topoisomerase IV .	0
8	Although fluoroquinolones are effective against a broad spectrum of bacteria alarming increase in fluoroquinolone-resistant pathogens warrants the need to develop novel drugs against new cellular targets.	0
9	Bacterial topoisomerase I responsible for relaxing negatively supercoiled DNA is the most common type IA topoisomerase present in almost all bacteria .	0
10	Escherichia coli topoisomerase I (EcTOP) is the well studied prototype for type IA topoisomerase .	0
11	EcTOP relaxes negatively supercoiled DNA through a magnesium-dependent ATP-independent catalytic mechanism.	0
12	No specific inhibitor for bacterial topoisomerase I effective at a relevant clinical and physiological concentration has been identified.	0
13	Bacterial topoisomerase I by virtue of its presence in nearly all bacterial genomes and in view of its association with DNA during the vulnerable stage of cleavage-religation could be utilized as a target for novel antimicrobials .	0
14	This strategy could be useful in developing drugs to treat highly fatal bacterial diseases like tuberculosis .	0
15	The fact that approximately one-third of the world's population is affected by tuberculosis indicates the need to develop effective drugs against this disease .	0

16	Also since multiple drug resistance is common in Mycobacterium tuberculosis it would be significant if a novel antibiotic targeting M. tuberculosis DNA topoisomerase I can be developed .	0
17	A logical first step towards finding inhibitors selective to M. tuberculosis topoisomerase I is to characterize the DNA modification ability of this enzyme.	0
18	In this study we describe a new expression and purification protocol for recombinant M. tuberculosis topoisomerase I capable of producing milligrams of pure protein.	5
19	We also report the first detailed characterization of this enzyme with respect to its DNA cleavage sites and relaxation activity under different assay conditions.	0
20	Results	0
21	Expression and purification of M. tuberculosis topoisomerase I	0
22	Genome sequencing of M. tuberculosis H37Rv strain has revealed the presence of topA gene Rv3646c which encodes a DNA topoisomerase I (MtTOP) comprising of 934 amino acids with an estimated molecular weight of 102.3 kDa .	0
23	Previously Yang et al have cloned and purified DNA topoisomerase I from M. tuberculosis Erdman strain in E. coli BL21 (DE3).	0
24	Our efforts to express and purify recombinant MtTOP in E. coli BL21 (DE3) similarly by induction of the T7 promoter were frustrated by the insolubility of the expressed protein.	0
25	Difficulties have also been encountered by other researchers while trying to use recombinant DNA topoisomerase I gene present in genomic libraries of M. tuberculosis and Mycobacterium smegmatis to complement the temperature dependent deficiency of topoisomerase I (topA) function in E. coli strain AS17 .	0
26	Difference in codon usage was surmised to be one of the possible reasons behind this result .	0
27	We overcame these difficulties by expressing MtTOP from a recombinant plasmid pLIC-MTOP in an E. coli Arctic express (DE3)RP strain (Stratgene) at low temperatures (12°C).	4

28	The Arctic express (DE3)RP strain contained a chromosomally integrated T7 RNA polymerase which was expressed from the lacUV5 promoter.	0
29	Induction of T7 RNA polymerase protein synthesis with IPTG resulted in the expression of the T7 promoter-driven recombinant protein.	3
30	In addition the Arctic express (DE3)RP strain expressed cold chaperonin proteins (Cpn10 and CPn60) and extra copies of tRNAs (recognizing arginine and proline codons) that facilitated the expression of recombinant proteins by overcoming issues of protein solubility and codon bias respectively.	1
31	Recombinant MtTOP was soluble and initially expressed as a hexa-histidine fusion protein only in the presence of IPTG (Figure 1).	3
32	Purification of the fusion protein was achieved using nickel affinity chromatography.	3
33	Subsequent SDS-PAGE analysis (Figure 2A) showed the predominant presence of only the fusion protein with the expected molecular weight.	0
34	The hexa-histidine fusion tag was cleaved off by TEV protease treatment and MtTOP of high purity was eluted by increasing the potassium chloride gradient from a single-stranded DNA cellulose column (Figure 2B) .	2
35	The eluted fractions were pooled and dialyzed into storage buffer.	1
36	Approximately 12 milligrams of purified protein was obtained from 7 L of bacterial culture in LB medium.	4
37	Characterization of DNA relaxation activity of M. tuberculosis topoisomerase I	0
38	DNA relaxation assay was used to characterize the purified MtTOP.	0
39	We compared the ability of MtTOP with that of similarly purified E.coli topoisomerase I (EcTOP) in relaxing negatively supercoiled DNA by agarose gel electrophoresis.	0
40	Initial assays evaluated the minimum amount of enzyme (MtTOP or EcTOP) required to bring about complete relaxation of negatively supercoiled DNA under standard conditions (Figure 3).	0
41	One unit of enzyme was defined as the amount of enzyme required to relax 0.5 μg of negatively supercoiled plasmid DNA in 30 min at 37° C.	0

42	Results indicated that 100 ng of EcTOP and 500 ng of MtTOP (Figure 3A) constitute one unit of enzyme activity.	1
43	However at lower concentrations of enzyme = 12.5 ng there is no difference between the ability of MtTOP and EcTOP in removing the negative supercoils from the plasmid DNA substrate (Figure 3).	0
44	The percent relaxation values reported are averages of at least three independent experiments.	0
45	Error bars denote the standard error of mean.	0
46	For a more detailed analysis of the relaxation activity of the purified enzymes a time course assay with 50 ng each of MtTOP and EcTOP was performed (Figure 4).	0
47	At the early time points the rate of removal of the negative supercoils by the two enzymes was similar.	0
48	However as the plasmid DNA substrate became partially relaxed the relaxation activity of MtTOP was less efficient than EcTOP in removing the residual negative supercoils.	0
49	It has been a well known fact that Mg2+ ions are required for the relaxation activity of bacterial type IA topoisomerases including E. coli topoisomerase I .	0
50	We compared the Mg2+dependence of the relaxation activity of EcTOP and MtTOP using two different enzyme concentrations (50 ng or 1 unit in a 20-?l assay) and a range of Mg2+ levels (Figure 5).	0
51	At a lower enzyme concentration (50 ng) relaxation by EcTOP had a optimal range of Mg2+ concentrations between 2.5 to 7.5 mM while the optimal range of Mg2+ concentrations for MtTOP was slightly higher (5?12.5 mM) (Figure 5B).	0
52	Similar optimal levels of Mg2+ were found for the relaxation activities of both the EcTOP and MtTOP at higher enzyme concentrations equivalent to one unit of enzyme activity with no relaxation observed in the absence of Mg2+ (Figure 5A).	0
53	The optimal Mg2+ concentrations found here for MtTOP are higher than the 1 mM concentration determined in previous work .	0
54	In other studies involving the characterization of the topoisomerase I from M. smegmatis the optimal Mg2+ concentration for relaxation activity was found to be about 5 mM .	0

55	Mapping of DNA cleavage sites using single-stranded DNA substrates	0
56	Although the majority of topoisomerases do not have specific sequence requirements for cleavage sites many of them show at least a certain degree of non -randomness in cleavage site recognition .	0
57	For example EcTOP and Micrococcal luteus topoisomerase I cleave the sequence CXXX? (? represents the cleavage site) more preferentially than others .	0
58	Archeal and bacterial reverse gyrases which are type IA topoisomerases also have limited sequence requirements with only the preference of a cytosine or requirement of at least a pyrimidine at the -4 position of the cleavage site .	0
59	Previous studies elucidating the sequence specificity of topoisomerase I from M. smegmatis reported a strong topoisomerase I site (STS) wherein the enzyme recognizes and cleaves the sequence CG/TCT?T .	0
60	We utilized different single-stranded 5'-32P labeled DNA substrates ranging from 200?550 bases in length generated from either an E. coli plasmid or M. tuberculosis genomic DNA to characterize the MtTOP preferred cleavage sites.	0
61	Results indicate that the DNA cleavage selectivity of MtTOP is very similar to that of EcTOP (Figure 6A Table 1).	0
62	The two enzymes share many cleavage sites on DNA derived either from E. coli or M. tuberculosis but some cleavage sites were preferred by only one of these two enzymes (Figure 6B Table 1).	0
63	All of the cleavage sites for both enzymes were found to have a cytosine at the -4 position (CXXX?) as previously shown for many bacterial topoisomerase I enzymes .	0
64	There was no specific cleavage sequence recognition for MtTOP as reported for M. smegmatis topoisomerase I.	0
65	Discussion	0
66	Tuberculosis (TB) is the second leading cause of adult deaths due to infectious diseases world-wide second only to HIV.	0
67	The surge in multi-drug resistant M. tuberculosis makes it crucial to identify novel targets for development of new TB treatment.	0
68	M. tuberculosis topoisomerase I could be one such novel target since there is only one type IA topoisomerase found in M. tuberculosis.	0

69	A recent genome wide transposon mutagenesis experiment has postulated and categorised M. tuberculosis topA gene as essential .	0
70	It is also likely to be essential because every bacterium has at least one type IA topoisomerase activity.	0
71	MtTOP is therefore an attractive target for drugs which would interfere with its relaxation activity (catalytic inhibitors).	0
72	Moreover besides inhibiting the overall relaxation activity of MtTOP a more potent bactericidal effect could be achieved by drugs (catalytic poisons) that enhance the accumulation of covalent complexes on DNA similar to the bactericidal mechanism of fluoroquinolones on type IIA bacterial topoisomerases.	0
73	To aide such drug development efforts it is important to have MtTOP protein in high purity and quantity.	0
74	Here we report that by utilizing the E. coli Arctic express RP(DE3) strain we took advantage of the higher GC rich codon usage efficiency and low temperature chaperone in this strain to obtain soluble MtTOP in high yield (12 mg from 7 L of bacterial culture).	4
75	This enables future development of high through-put assays for inhibitors targeting MtTOP.	0
76	The DNA cleavage activity of MtTOP has not been characterized previously and there is a also a need for more detailed analysis of its DNA relaxation activity than in the early study of the enzyme .	0
77	Careful comparison with E. coli topoisomerase I (EcTOP) showed that the two enzymes had similar efficiency initially in relaxing the negatively supercoiled plasmid DNA isolated from E. coli.	0
78	However as the substrate plasmid DNA became partially relaxed MtTOP was slower than EcTOP in removing the residual negative supercoils.	0
79	This could be due to the different C-terminal domain found in the enzymes.	0
80	The C-terminal domain found in EcTOP has been proposed to be important for substrate binding and coordination of strand passage during the relaxation cycle .	0

81	The C-terminal domain of MtTOP has no homology to the C-terminal domain in EcTOP so it may function differently during the catalytic cycle.	0
82	The N-terminal two-thirds the transesterification domains of EcTOP and MtTOP have high degree of homology (41.9% identical).	0
83	Analysis of cleavage sites on both E. coli and M. tuberculosis derived DNA substrate showed that the cleavage site preferences are quite similar with a C in the -4 position as have been observed for several bacterial topoisomerase I as well as archeal and bacterial reverse gyrase enzymes.	0
84	It is somewhat surprising that the cleavage site preference of MtTOP is not the same as that reported for M. smegmatis topoisomerase I (CG/TCT?T).	0
85	It is possible that this is due to the different experimental protocols used in analysis of the cleavage sites .	0
86	Besides M. smegmatis topoisomerase I there are other examples of type IA topoisomerases that have cleavage site preferences different from that of EcTOP.	0
87	These include CTT? for E. coli topoisomerase III CANNN? for human topoisomerase III ANN? for yeast topoisomerase III .	0
88	It remains unclear which part of the type IA enzyme structure determines the cleavage site selectivity.	0
89	The specific sequence information for DNA cleavage by MtTOP should be useful in design of oligonucleotide substrates for DNA cleavage assays.	0
90	Conclusion	0
91	A new procedure for expression and purification of recombinant MtTOP protein in high yield has been described.	5
92	The enzyme is as efficient as EcTOP in initial removal of negative supercoils from plasmid DNA but is less efficient than EcTOP in removing the remaining negative supercoils.	0
93	The preferred DNA cleavage sites of MtTOP have limited sequence specificity but contain a C nucleotide in the -4 position similar to most bacterial topoisomerase I and archeal reverse gyrase cleavage sites characterized previously.	0

94	Methods	0
95	MtTOP expression and purification	0
96	MtTOP was expressed from a recombinant plasmid pLIC-MTOP in E.coli Arctic express (DE3)RP strain (Stratagene).	3
97	MtTOP coding sequence was amplified from the genomic DNA of M. tuberculosis H37RV strain with suitable primers (LIC-Mtop5'-TACTTCCAATCCAATGCAGCTGACCCGAAAACG and LIC-Mtop3'-TTATCCACTTCCAATGTTATTAGTCGCGCTTGGCTGC) using PfuUltra II Fusion HS DNA polymerase (Stratagene) and cloned into a vector pLIC-HK through a ligation independent cloning procedure .	1
98	Cloning of MtTOP coding sequence into this vector containing a T7 promoter allowed T7 RNA polymerase dependent expression of MtTOP along with a tobacco etch virus (TEV) protease-cleavable N-terminal hexahistidine tag .	0
99	The resulting pLIC-MTOP plasmid capable of expressing recombinant MtTOP was first isolated in E. coli NEB Turbo competent cells (New England Biolabs) and then transformed into Arctic express (DE3)RP cells after sequence confirmation.	2
100	Expression of MtTOP in transformed Arctic express (DE3)RP cells was induced by 1 mM IPTG at 12° C according to the manufacturer's (Stratagene) protocol.	4
101	After 24 h of induction the cells were collected and subjected to freeze-thaw lysis in lysis buffer (50 mM NaH2PO4 300 mM NaCl 10 mM imidazole 1 mg/ml Lysozyme pH 8.0).	0
102	The recombinant protein in the soluble lysate was allowed to bind to Ni-NTA agarose (Qiagen) and packed into a column.	3
103	After washing the column overnight with wash buffer (50 mM NaH2PO4 300 mM NaCl 20 mM imidazole pH 8.0) the topoisomerase protein was eluted with an elution buffer (50 mM NaH2PO4 300 mM NaCl 250 mM Imidazole pH 8.0) containing higher concentrations of imidazole.	3
104	Eluted MtTOP was cleaved with TEV protease to remove the N-terminal hexa-histidine tag and purified by passing through a single-stranded DNA cellulose column as described .	2

105	DNA Relaxation Activity assays	0
106	To assay for one unit of relaxation activity EcTOP and MtTOP enzymes of the same concentrations were diluted serially ranging from 500?1 ng and assayed for DNA relaxation activity in a standard reaction volume of 20 ?l with 10 mM Tris-HCl (pH 8.0) 50 mM NaCl 0.1 mg/ml gelatin 6 mM MgCl2 and 0.5 ?g of supercoiled pBAD/thio plasmid DNA (purified by CsCl gradient centrifugation).	0
107	After incubation at 37° C for 30 min the reactions were stopped by adding 5 ul of 50 mM EDTA 50% glycerol and 0.5%(v/v) bromophenol blue.	0
108	The DNA was electrophoresed in a 1.0% (w/v) agarose gel with TAE buffer (40 mM Tris-acetate pH 8.1 2 mM EDTA).	0
109	The gel was stained with ethidium bromide and photographed over UV light.	0
110	One unit of enzyme was defined as the least quantity of the enzyme required for complete relaxation of negatively supercoiled DNA under the given reaction conditions.	0
111	Mg2+ dependence of EcTOP and MtTOP to relax negatively supercoiled DNA was compared with either 50 ng or one unit of enzyme (100 ng of EcTOP 500 ng of MtTOP) under varying concentrations of MgCl2 ranging from 0?20 mM over a time period of 30 min at 37° C with similar reaction conditions as described above.	0
112	Also a low concentration (50 ng) of EcTOP and MtTOP under standard conditions (6 mM MgCl2) as described earlier was used to compare the ability of the respective enzymes to relax negatively supercoiled DNA at various time points of 0 10 20 30 45 60 75 90 120 and 180 sec at 37° C.	0
113	Cleavage of Single-stranded DNA	0
114	To compare and map the cleavage sites of EcTOP and MtTOP single stranded DNA substrates were generated first by PCR (Table 2) followed by strand denaturation.	0
115	Each of these substrates were radio-labeled at the 5' end by having one of the corresponding forward or reverse primers labeled with [?-32P]ATP in the presence of T4 polynucleotide kinase prior to the PCR.	0

116	The PCR products were purified using the DNA Clean and Concentrator Kit (Zymos) and eluted in TE buffer (10 mM Tris-HCl pH 8.0 1 mM EDTA).	0
117	Prior to the addition of topoisomerase in the cleavage assay the DNA substrate was denatured to single strands by heating at 95° C for 5 min and rapidly cooled on ice.	0
118	After incubation with the topoisomerase at 37° C for 10 min trapping of the covalent enzyme-DNA complex and cleaved DNA was achieved by the addition of 0.1 M NaOH.	0
119	After neutralization the DNA was electrophoresed in a 6% polyacrylamide sequencing gel followed by autoradiography of the dried gel to visualize the 5'-end-labeled DNA cleavage products.	0
120	DNA sequencing reaction products were generated with the same 5' end labeled primer corresponding to that of the substrate used in the cleavage assay and by following the cycle sequencing procedures according to the manufacturer's instructions (SequiTherm DNA sequencing Kit Epicentre).	0
121	The sequencing reaction products were electrophoresed next to lanes containing cleavage products to identify the cleavage sites.	0
122	Authors' contributions	0
123	AA carried out the relaxation and DNA cleavage assays and drafted the manuscript.	0
124	ND and BC developed and carried out the expression and purification protocol.	0
125	YT conceived the study and participated in the design and coordination and helped to draft the manuscript.	0
126	All authors read and approved the final manuscript.	0

Table B.1: Complete Annotation of Paper #8

Appendix C

Annotation Instructions

C.1 Annotation Instructions

1. After reading the sentence in the leftmost column, you are to read each sentence in the rightmost column and mark the box in the second column if the content of the sentence in the third column has some similarity to the content of the sentence in the leftmost column. Mark as many boxes as you deem appropriate.

2. Only one sentence appears in the leftmost column (it is repeated for easy reference). The sentences in the third column are all of the sentences that appear in a scientific article and are presented chronologically as they appear in the article.

3. Once you have finished the first part of the annotation task, you will be presented with the sentence pairs that you have marked and you will be asked to perform the second part: for each sentence pair, give the strength of the similarity, ranking them from a weak match to a strong match (1-5)

Curriculum Vitae

Name:	**Kokou Hospice Houngbo**
Post-Secondary Education and Degrees:	University of Western Ontario London, ON, Canada Ph.D., Computer Science, April 2017 Rochester Institute and Technology Rochester, New-York, USA M.Sc. Information Technology, August 2008. University of Abomey-Calavi Abomey-Calavi, Benin Bachelor's Degree in English, June 2001. Polytechnic University College Abomey-Calavi, Benin Electrical and Electronics Engineering Diploma, September 2000.
Honours and Awards:	Fulbright Scholarship July 2006—April 2008.
Related Work Experience:	Graduate Research and Teaching Assistant The University of Western Ontario, London ON Canada September 2008—August 2013. Computer System Professional and Teacher Professional and Commercial Training Center, Cotonou, Benin September 2001—June 2006.

Publications:

1. **Houngbo, Hospice.**, Mercer, R.E.: Investigating Citation Linkage with Machine Learning. *In Canadian Conference on Artificial Intelligence*, pp. 78-83. Springer, Cham,

2017.

2. **Houngbo, Hospice.**, Mercer, R.E.: Investigating Citation Linkage as an Information Retrieval Task. In: to appear. *(2017), CIcLing 2017*

3. **Houngbo, Hospice.**, Mercer, R.E.: An automated method to build a corpus of rhetorically classified sentences in biomedical texts. *In: Proceedings of the First Workshop on Argumentation Mining*, Baltimore, Maryland, Association for Computational Linguistics (2014) 19-23.

4. **Hospice Houngbo** and Robert E. Mercer. Method mention extraction from scientific research papers. *In Proceedings of COLING 2012, pages 1211-1222*, Mumbai, India, December 2012.