Contextual Citation Recommendation using Scientific Discourse Annotation Schemes

Daniel Duma

Doctor of Philosophy

Institute for Language, Cognition and Computation

School of Informatics

University of Edinburgh

2018

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Daniel Duma

Acknowledgements

The previous page establishes that all the work contained herein is my own work. This page is the giant asterisk to it. There are too many people to list here without whose help, input and encouragement this document would have never existed, and without whom I would not be who I am now.

I am eternally indebted to my supervisor Ewan Klein, for taking me under his wing, for patiently working with me through my crazy ideas and for his guidance and gentle help in walking this path. Ewan has been to me much more than a supervisor: a mentor, a friend, and an inexhaustible source of inspiration.

Similarly, I would like to extend my deepest gratitude to Maria Liakata and her group for the most exciting and fruitful collaboration and the many opportunities it has brought me, including my brief stint at the Alan Turing Institute. A particularly large bucket of thanks goes to James Ravenscroft, for his priceless help in annotating a million files with CoreSC, but also for his friendship and his continuous support and encouragement through the hardest times.

The untimely departure of Jon Oberlander was a very difficult moment, not just for me, but for all who knew him. Jon was present in one way or another at every step of my academic career in Edinburgh: teacher, mentor, examiner, internal reviewer. Every interaction I ever had with him was enriching and uplifting, and while juggling twenty official positions he still always managed to find time to help and share insightful advice. After talking to Jon I always walked away with a bigger smile and a lighter heart. He held the helm steady as head of ILCC, he was an inspiring lecturer, he was my internal reviewer for my three yearly reviews and he had previously been my second marker for my MSc dissertation. His cheerful and encouraging scribbled comments on it remain to this day a motivation to me, and in spirit this thesis owes a lot to him.

From the day I arrived, Clare Llewellyn took on the role of big sister, sharing with me her hard-earned wisdom, and pushing me to publish, publish, publish. For all of this, and for her tips and guidance, I am deeply indebted to her.

My thanks also to Victor Lavrenko, for his invaluable input during the early stages of this research. I consider myself lucky to have had the benefit of his insights before he left to pursue the Silicon Valley dream. Similarly, I am grateful to Charles Sutton for accepting me as yet another student in times of need, and whose input (before he also left to pursue the Silicon Valley dream) was always very insightful, kind and helpful.

Beyond the purely academic, I am very thankful for the chance to overlap in space-time with the most astoundingly talented and kind people. I am very fortunate to have enjoyed the company and friendship of Mihai Dobre, Valentin Radu, Stefanie Speichert, Emilian Radoi, Ivan Cano, Sam Ribeiro, Gustav Henter, Aleks Stolicyn, and Ruslan Habalov, among many others. I am also thankful for the great times I shared with Federico Fancellu, Manny Kahembwe, Ben Krause and my other Alexa Prize teammates. Too many names must go unlisted here, but I want to extend a thank you to the many, many Forum dwellers that I have got to meet in my years here, as well as to my fellow interns and to my friends in the startup bubble, from whom I have learned more than books could ever teach me.

I would also like to thank Larry and Sergey for all the free food and for two astonishing summers. How the two internships at Google changed my life can hardly be summarised here.

And finally, I lovingly and deeply thank my parents, Eugenia and Calin, for not only giving me life (which turns out to be massively useful), but also an education, a loving home and their endless and unwavering support.

Abstract

All researchers have experienced the problem of fishing out the most relevant scientific papers from an ocean of publications, and some may have wished that their text editor suggested these papers automatically. This thesis is vertebrated by this task: recommending contextually relevant citations to the author of a scientific paper, which we call *Contextual Citation Recommendation* (CCR). Like others before, we frame CCR as an Information Retrieval task and we evaluate our approach using existing publications. That is, an existing in-text citation to one or more documents in a corpus is replaced with a placeholder and the task is to retrieve the cited documents automatically. We carry out a cross-domain study and evaluate our approaches using two separate document collections in two different domains: computational linguistics and biomedical science.

This thesis is comprised of three parts, which build cumulatively. Part I establishes a framework for the task using a standard Information Retrieval setup and explores different parameters for indexing documents and for extracting the evaluation queries in order to establish solid baselines for our two corpora in our two domains. We experiment with symmetric windows of words and sentences for both query extraction and for integrating the anchor text, that is, the text surrounding a citation, which is an important source of data for building document representations. We show for the first time that the contribution of anchor text is very domain dependent.

Part II investigates a number of scientific discourse annotation schemes for academic articles. It has often been suggested that annotating discourse structure could support Information Retrieval scenarios such as this one, and this is a key hypothesis of this thesis. We focus on two of these: Argumentative Zoning (AZ, for the domain of computational linguistics) and Core Scientific Concepts (for the domain of biomedical sciences); both of these sentence-based, scientific discourse annotation schemes which define classes such as Hypothesis, Method and Result for CoreSC and Background/Own/Contrast for AZ. By annotating each sentence in every document with

AZ/CoreSC and indexing them separately by sentence class, we discover that consistent citing patterns exist in each domain, such as that sentences of type Conclusion in cited papers are consistently cited by other sentences of type Conclusion or Background in citing biomedical articles.

Finally, Part III moves away from simple windows over terms or over sentences for extracting the query from a citation's context, and investigates methods for supervised query extraction using linguistic information. As part of this, we first explore how to automatically generate training data in the form of citation contexts paired with an optimal query to generate. Second, we train supervised machine learning models for automatically extracting these queries with limited prior knowledge of the document collection and show important improvements over our baselines in the domain of computational linguistics. We also investigate the contribution of stopwords to each corpus and we explore the performance of human annotators at this task.

Lay Summary

We could think of a scientific article as a quantum of science: the minimal unit of progress. Different areas of science have different conventions as to how a scientific paper should be structured, what style of writing is expected and acceptable, etc. However, one thing they all have in common: citing other published papers.

The rate of scientific publishing has been growing exponentially, which makes the job of an academic author more difficult. In spite of the availability of search engines, discovering, reviewing and understanding the relevant publications to cite takes time and effort. We aim to ease the discovery of this relevant literature.

This thesis is vertebrated by this task: automatically recommending contextually relevant citations to the author of a scientific paper, which we call *Contextual Citation Recommendation* (CCR). Like others before, we treat CCR as Information Retrieval (web search), and so each academic paper as a "page" and the citations from one paper to another as "links" between them. We evaluate how relevant our recommendations are by taking previously published papers, removing their existing citations, substituting them with placeholders (e.g. "<insert citation here>") and then trying to recover these prior citations from a large collection of papers. The task is to find the most relevant paper for every place in the text where a citation originally existed.

We evaluate our approaches using two separate document collections in two different scientific domains: computational linguistics and biomedical science. This thesis is comprised of three parts, which build cumulatively. Part I establishes a framework for the task using a standard Information Retrieval setup. A search engine works in two major stages: *indexing* and *retrieval* (search). During indexing, the documents are read and processed to create a representation that will make searching possible and efficient. We explore what parts of a document we should index to obtain the best results: for example, the title, the abstract, or the whole text and their relative contributions. At the same time we investigate indexing the "anchor text", that is, the text that appears around a web link, or in our case, a few words or sentences that appear around

a citation to a given paper from another paper. We also explore different parameters for extracting the evaluation queries, which are equivalent to the sort of query a user would type into a search engine. We show for the first time that the contribution of anchor text is very domain dependent.

In Part II we investigate how scientific discourse annotation schemes for academic articles could help for this task. Every document has a purpose, a rhetorical aim, such as to inform or to persuade, and scientific articles are meant to have a formal structure based on argumentation that facilitates this. We can thus label each sentence in an article based on its contribution to the rhetorical aim. We use existing trained machine learning classifiers to annotate each sentence in every document with its type (e.g. Hypothesis, Method or Result) and we investigate what types of sentences (in citing papers) tend to cite what other types of sentences (in cited papers). We discover that consistent citing patterns exist in each domain, such as that sentences of type Conclusion in cited papers are consistently cited by other sentences of type Conclusion or Background in citing biomedical articles.

Finally, in Part III we investigate query extraction. In Parts I and II we used baseline methods for this, such as taking the words in the text as they appeared, filtering out grammar words, and treating this as the query. Here we investigate supervised machine learning methods for extracting better performing queries by using linguistically inspired features. As part of this, we first explore how to automatically generate training data in the form of citation contexts paired with an optimal query to generate. Second, we train different machine learning models for automatically extracting these queries and we compare their performance in the two domains. We also evaluate how human annotators fare at extracting these queries.

Contents

List of Figures

Chapter 1

Motivation and task definition

> “Research is what I’m doing when I don’t know what I’m doing."
>
> Wernher von Braun

1.1 Introduction

For many researchers, keeping up with the research in their academic field and finding the related work they seek is an unsolved problem. Scientific publishing is nowadays increasingly digital, and yet, with a growing rate of publication, finding the published research one seeks is still a fragmented and time-consuming task.

In practice, authors have different information needs at different stages of their research (Teufel, 2010). When approaching a new field, one may seek a broad overview of the common approaches employed, while as the research progresses, one may need more finely-grained relevance to specific passages in a draft document. As examples, one may wish to:

- Find out how a particular problem has been solved before and whether those techniques are applicable to one’s current research.
- Find papers (that is, reports of other research) that have the same research goal or that employ similar methods.
- Find support for a particular statement that is a common observation.

Search engines for this task do exist, but in spite of their widespread availability and increasing sophistication, they have serious limitations. Still much unnecessary effort is left to the author: trying out different queries, scrolling down a long list of results, and, in the better cases, following links to related literature and filtering by date, citation count or author. This lengthy process does not guarantee that the information sought will be found.

Existing software solutions for this task based on keyword and graph search do not fully solve these problems of scientific discovery, as they cannot specifically suggest appropriate citations based on a specific information need, and the search cannot be performed with attention to specific types of statements in the documents to recommend.

This thesis is motivated by one idea: to help researchers carry out their research. We want to augment human capabilities: help authors discover relevant scientific literature and more efficiently write a scientific paper.

1.2 Academic literature: searching and publishing

At the time of writing, there exist private academic search engines, citation indexing services and paywalled collections such as ScienceDirect[1] and Web of Science[2]. However, open and free multidisciplinary options also exist, perhaps best exemplified, in order of popularity, by Google Scholar[3] and Microsoft Academic Search[4].

In addition, other services are worthy of note here. First, CiteSeerX[5] has been run as a not-for-profit service at UPenn since 2007, doing essentially the same task as Google Scholar: crawling the web, retrieving academic papers, converting them to machine readable formats, indexing this information and building a large graph of papers and authors, and providing keyword search, faceted browsing and a recommender system. Second, SemanticScholar[6] seems to have picked up the torch from CiteSeerX in being the most innovative open academic search engine by adding several important features. At present, beyond the standard filters for publication date and author it allows the user to filter results by type of publication (journal article, review and conference paper) as well as the specific venue name (e.g. LREC, ACL, etc.). At the same time, when navigating the links to citing papers and cited papers, these links are

[1] https://www.sciencedirect.com/

[2] https://login.webofknowledge.com/

[3] https://scholar.google.co.uk/

[4] https://academic.microsoft.com/

[5] http://citeseerx.ist.psu.edu/

[6] https://www.semanticscholar.org

ranked by the degree to which the citing publication was influenced by the cited one, and the citing sentences are shown. CiteSeerX is an example of research in this field becoming a useful feature of an academic tool (Valenzuela et al., 2015).

In the years since the beginning of this research project, there have been significant shifts in academic publishing in some fields, with some of them championing Open Access publishing. Specifically in Artificial Intelligence and related disciplines (e.g. Natural Language Processing), which are of relevance to the subject matter of this thesis, the overall rate of publishing has continued to increase. This is consistent with the overall trend across disciplines (Larsen and Von Ins, 2010). Further, the scientific community has been especially committed to Open Access, in some cases boycotting the established publishers to further this cause.[7] More and more powerful public initiatives are pushing for Open Access publishing of all publicly-funded research, such as Coalition S[8], a commitment by the national research funding organisations of 11 European countries (including the UK) that "by 2020 scientific publications that result from research funded by public grants provided by participating national and European research councils and funding bodies, must be published in compliant Open Access Journals or on compliant Open Access Platforms". Along these lines, a recent government report by Hall and Pesenti (2017) outlines to the UK government the need to "increase ease of access to data", including "development of data trusts, to improve trust and ease around sharing data", "making more research data machine readable" and "supporting text and data mining as a standard and essential tool for research".

The number and size of conferences in Artificial Intelligence and related fields has been growing exponentially[9], which increases the number of venues to publish in in step with the growth of the community of researchers, but beyond this, publishing has moved online. ArXiv.org[10] is a pre-print server that has become the go-to repository for the most recent research in computer science and artificial intelligence, and before that for physics, astronomy, mathematics and statistics. ArXiv is a fully open access and free repository, which has sparked the community-led development of some integrations and tools. Perhaps the most worthy mention of this is ArXiv Sanity Preserver[11], which integrates a simple recommender system with an innovative user interface (UI) that presents not just the title and abstract of the paper as is common but also thumb-

[7] https://retractionwatch.com/2018/05/01/thousands-boycott-new-nature-journal-about-machine-learning/

[8] https://www.scienceeurope.org/coalition-s/

[9] https://nips.cc/Conferences/2017/Press#demo-1853

[10] https://arxiv.org/

[11] http://www.arxiv-sanity.com/

nails of each page. Similar repositories to ArXiv now exist for other fields, such as bioRxiv[12] for the biomedical sciences.

An online repository of all publicly accessible scholarly articles in this latter domain of biomedical and life sciences, PubMed Central,[13] has existed since the year 2000. This repository is particularly important to our research, as it is the source of one of our corpora.

Open Access (OA) publishing has grown significantly over the last few years, partly driven by public initiatives (Björk et al., 2014), and perhaps encouraged by findings that OA articles are more likely to be cited (Antelman, 2004) – according to some studies twice as likely on average (Eysenbach, 2006).

While the adoption of OA publishing is far from universal, the current trend suggests that at a point in the not too distant future the majority of scientific publishing will be available to anyone at no cost to the user. Such a situation is a requirement for the approach we suggest to be applicable.

So the options for literature discovery are many for many fields of research, and we can see a trend towards open access publishing, but actually finding the relevant literature continues to be a task that breaks the research workflow. There seems to be a large gap to be filled here. User modelling, from the field of Human Computer Interaction (Fischer, 2001) aims to make use of a user's current task and backgrounds to reduce their information overload. One can envision a type of academic literature discovery system that will integrate with the research process at different stages, proactively providing relevant information about previous work. The research we present here aims towards this goal.

1.3 Contextual Citation Recommendation

Imagine that as a researcher in the field of computational linguistics you are working on a draft paper which contains a sentence like the following[14]:

> A variety of coherence theories have been developed over the years ... and their principles have found application in many symbolic text generation systems (e.g. **[CITATION HERE]**)

[12] https://www.biorxiv.org/

[13] https://www.ncbi.nlm.nih.gov/pmc/about/intro/

[14] Adapted from the introduction to Regina Barzilay and Mirella Lapata. 2008. Modelling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Wouldn't it be helpful if your editor automatically suggested some references that you could cite here? This is what a *citation recommendation* system ought to do. If the system is able to take into account the context in which the citation occurs — for example, that papers relevant to our example above are not only about text generation systems, but specifically mention applying coherence theories — then this would be much more informative. So a *contextual citation recommendation* (CCR) system is one that assists the author of a draft document by suggesting other documents with content that is relevant to a particular context in the draft. Our research explores novel ways in which these recommendations might be improved by attending to the type of citation one may be seeking, yet abstracting away from the engineering task of integrating CCR into any specific editing tool.

It is necessary to define some terminology that will be employed throughout this thesis. In the following passage, the strings 'Scott and de Souza, 1990' and 'Kibble and Power, 2004' are both *citation tokens*:

> A variety of coherence theories have been developed over the years... and their principles have found application in many symbolic text generation systems (e.g. Scott and de Souza, 1990; Kibble and Power, 2004)

Note that a citation token can use any standard format. Furthermore

- a *citation context* is the context in which a citation token occurs, with no limit as to how this context is represented, length or processing involved;

- a *collection-internal reference* is a reference in the bibliography of the source document that matches a document in a given corpus;

- a *resolvable citation* is an in-text citation token (e.g. "Doe et al. (2010)" or [1]) which resolves to a collection-internal reference. This is important, as we evaluate our system by trying to recover citations that have been already used in published work, and for this we need access to the contents of the original document.

- a *target document* is the document in the document collection that a collection-internal reference points to;

- a *citation placeholder* is a token that substitutes a missing citation token. Continuing the example above, this would be the span of text "[CITATION HERE]".

The ultimate aim of this research is to advance the technology required to build a CCR System, one that can assist the author of a draft document by suggesting other documents with content that is relevant to a particular context in the draft, paying attention to the function of the sought citation in the draft text and attempting to provide support to claims, comparing methodologies and findings.

1.4 Aims and scope of this thesis

1.4.1 Aim

We want to recommend relevant papers to the author of a draft academic paper. The relevance should be as fine-grained as possible: to the paragraph, sentence, and ideally to a sub-span of a sentence. In pursuit of this aim, in this thesis we investigate applying existing scientific discourse annotation schemes to this task, looking for patterns in citation between different types of citing and cited sentences.

Some key hypotheses of this research are:

- H1: The rhetorical or argumentative function of sentences in an academic document can be classified into a number of discrete classes using shallow semantics.
- H2: The classification of a citation context via H1 helps to identify relevant content in candidate reference papers in a corpus.
- H3: The identification of fine-grained relevant content via H2 improves the ranking of papers that are relevant with respect to a given citation context.
- H4: The application of linguistically motivated natural language processing methods to the selection and weighting of terms for a query improves the query's performance.

1.4.2 The task

Most literature on the task (He et al., 2010; Huang et al., 2014) so far has framed this task as an *information retrieval (IR)* task and we continue this approach in this work. There are strong reasons for this choice. The main one is practical: the list of potentially relevant papers is large enough that it demands special strategies to deal with their acquisition, storage, indexing, and with the assessment of relevance of each paper, if we are aiming for a robust, practical system. Another main reason is conceptual:

the task fits the standard formulation rather well: the user has an *information need* to which they seek an answer by writing a *query*, which in its simplest form is a sequence of keywords. A *similarity function* or metric is then applied between the query and each *document* in the *collection* in turn and the list of *most relevant* papers are then *ranked* in descending order according to their score.

Our task does bring some distinct features to the general IR approach. On one hand, scientific papers have a formal structure that is reasonably well understood and has been extensively studied (Teufel, 2010), which suggests that it could be employed to better inform the often somewhat naïve IR approach of treating the document as a single bag-of-words. On the other hand, IR systems expect a rather short query which can be ambiguous, can contain misspellings, etc. In our case however, we do not start from a formed query but from a long span of text that is rich in structure, and from which we can automatically extract this query.

We evaluate against corpora of existing publications, where the objective is to retrieve the citations that were originally employed in an existing publication. Our evaluation methodology is explained in detail in Section 2.6.

One important concept that much previous work has employed and so we need to define is that of *anchor text*, which is the text surrounding a citation to document d_2 found in $d_1 - d_n$. This is a key source of information for indexing web pages by search engines like Google (Brin and Page, 1998), and has a parallel in CCR. The key difference is that anchor text in web pages is often explicitly marked, while in scientific publications it needs to be extracted from the context through some processing (Ritchie et al., 2008).

1.4.3 Requirements

We necessarily need to explore scientific domains for which there are plenty of open access and machine-readable academic papers available. At the same time, it would be desirable to explore more than one domain so that we get some indication as to how well the approaches transfer between different bodies of scientific literature. The domains explored should ideally be distinct in regards to their object of study, the conventions adopted for document structure, the methods employed and the terminology employed.

Given the fact that this research is motivated by eminently practical goals, it is preferable to use existing, industrial grade IR systems rather than experimental, *ad-*

hoc or custom built ones. Robust, scalable, well tested and well documented retrieval engines that are freely available and open source software are keyword-based, so it is desirable to focus on keyword-based search.

1.5 Roadmap of this thesis

This thesis is comprised of three main parts, separated into seven chapters. These parts are complementary and aim to progressively build on top of each other. However, as they span a range of research subtopics, each of them contains its own review of previous work and description of methodology.

Part I (Chapters 1, 2 and 3) motivates our research, presents the framing of the task, provides a review of the most relevant related work and establishes our initial baselines using an information retrieval approach. In essence, it constitutes a first approach to the problem we are tackling by applying IR techniques that have been commonly used in previous work, while at the same time examining our two corpora in some detail.

- **Chapter 1** presents the motivation for this research and a necessary introduction to the field of Information Retrieval, as well as the architectures and approaches we employ. We also discuss potential evaluations and present our chosen evaluation method and metrics.

- **Chapter 2** provides an overview of the relevant previous work, which employs the same conceptual approach to evaluation, based on recovering the citations that we find in papers that have already been published. For this task, we need a large amount of machine-readable scientific articles in the same domain that are densely interlinked by citations, and here we also present the two corpora used in the experiments: the ACL Anthology Corpus and the PubMed Central Open Access Subset. These corpora belong to the two markedly different domains of computational linguistics and biomedical science, so we also offer metrics that describe their similarities and differences.

- **Chapter 3** presents a number of baseline approaches to the task using a standard Information Retrieval formulation. Specifically, we explore what we should consider to be the *context* of a citation. From this context we extract an evaluation query. We also investigate what text we should use to represent a document in the collection, with an interest on what produces the best results. Our experiments combining several query extraction methods with several document representation methods provide us with strong baselines for the work that follows.

Part II (Chapters 4 and 5) moves beyond standard IR approaches that pay little attention to the structure of a document. Instead, it explores applying Scientific Discourse Annotation schemes to this task. Under this formulation, each sentence in a scientific paper is automatically assigned a label that assigns it one of several mutually exclusive classes. Each class represents the rhetorical or argumentative function of that sentence with respect to the aim of the document. Our hypothesis is that a relation exists between types of citing sentences and types of cited sentences and that, if so, we could exploit these relations to increase the relevance of recommendations.

- **Chapter 4** introduces Scientific Discourse Annotation: annotation schemes for scientific articles that try to classify each sentence according to its function in the document. We review a number of the schemes defined over the years, and we motivate our final choice of using Core Scientific Concepts for PubMed Central and Argumentative Zoning for the ACL Anthology Corpus.

- **Chapter 5** presents experiments applying Scientific Discourse Annotation to the CCR task. We find consistent patterns between types of citing and cited sentences (e.g. sentences of type Results tend to cite sentences of type Results in biomedical papers). Interestingly, we also find these patterns in the anchor text of citations to a document.

Up until here, we have treated a simple window of tokens or sentences around the citation placeholder as the context of the citation, from which we extracted the query by simply taking the resulting bag-of-words. **Part III** (Chapter 6) moves to investigate if we can extract a better query by applying a measure of linguistically informed analysis.

- **Chapter 6** presents a new approach to extracting the query. It moves beyond simple windows of words or sentences to determining which keywords to select from an insertion context using linguistically motivated NLP methods. We show that we can generate the training data for supervised machine learning models directly from our document collection.

Finally, **Chapter 7** summarises the main contributions of this thesis and outlines some potential avenues for future research.

1.6 Contributions

The main contributions of this thesis are:

1. We carry out the most exhaustive cross-corpus examination of parameters for extracting text from a target document and the anchor text of its incoming citations, as well as for generating the query that will retrieve it. These results were partly reported in Duma and Klein (2014). We show that different domains cite differently: what is true of computational linguistics (a document is better described by the summaries made of it in other papers) does not hold for biomedical science (a document is better described by its own textual content).

2. For the very first time, we apply scientific discourse annotation to the task of CCR. We apply two separate scientific discourse schemes, Argumentative Zoning and Core Scientific Concepts, that were tailored to different domains, to two large corpora in these two domains. Our results show there are consistent patterns in citation between types of citing sentences and cited sentences, in both domains (Duma et al., 2016a).

3. We also applied this new approach to the anchor text of a document's citations and again found consistent citing patterns, as reported in Duma et al. (2016b).

4. We show that a human annotator can match or defeat a baseline keyword extraction system when documents are indexed by their full textual content (Duma et al., 2016c). However, when we add the anchor text of other documents to the document representation, the human is outperformed by the baseline. This discourages using sentence selection approaches for generating a CCR query.

5. We employ standard approaches to keyphrase extraction and demonstrate that adding phrases to be matched to a query is of little benefit to the CCR task: what really counts in our baseline IR formulation is the weighting of the terms.

6. We show that linguistic, morphological, structural and statistical features of the words in a text can be used as predictors of the weight to assign to a given term in an extracted query in our computational linguistics corpus. We also show that this does not hold in our corpus of biomedical science.

7. We contribute a novel approach to the annotation of this ideal per-term boost, based on finding the weights that will maximise the evaluation score for that context from the results of the evaluation query.

8. We publish all the code of our evaluation framework under an open source license[15]. At the same time, we make our corpus of manually annotated keywords in citation contexts freely available [16].

[15] https://github.com/danduma/minerva

[16] https://github.com/danduma/keywordcorpus

Chapter 2

Previous work and corpora

> “Quotation is a serviceable substitute for wit."
>
> Somerset Maugham

2.1 Introduction

As we proposed in the previous chapter, existing software solutions for scientific discovery may fall short of solving the information need of authors of scientific papers. In this section, we start by introducing the field of Information Retrieval and standard techniques in it. We then present common approaches in the related field of Recommender Systems, which overlap with Contextual Citation Recommendation in their practical aim, and to varying degrees in the techniques employed. We then review what we consider the most relevant approaches to CCR to date, paying attention to different aspects of the research that have been tackled. We explain common methods for document representation (that is, which text from the document or other documents in the collection to index) and query extraction and finally we present and describe the corpora for our experiments: the ACL Anthology Corpus and the PubMed Central Open Access Subset.

2.2 Information Retrieval

As mentioned in the previous chapter, previous attempts to develop CCR systems have typically framed the task as one of Information Retrieval (IR). Consequently, a short introduction to the field is in order (mostly based on Manning et al. (2008)).

First of all, IR operates over *document collections*, which tend to contain quite a large number of documents, from which text can be retrieved in some machine-readable format. Every document in the collection is usually treated as a "*bag of words*", that is, a collection of tokens (known in IR as *terms*) with no linguistic structure, although the *proximity* of one term to another is often used as a feature for indexing and retrieval. The total set of unique terms in the document collection is known as the *vocabulary*. Each term in the vocabulary is represented by a unique integer (i.e. a scalar), which represents the term's unique string.

The aim of the system is to retrieve and present to the user the most relevant documents found in the collection to satisfy the user's *information need*, i.e. "the topic about which the user desires to know more" (Manning et al., 2008). The information need is expressed through a *query*, which is typically a relatively short list of terms, which does not need to manifest a linguistic structure.

Some systems allow the user to perform *boolean retrieval*, in which the terms may be connected by logical operators (e.g. "(term1 AND term2) OR term3 AND NOT term4", where only documents containing terms that satisfy this logical formula will be retrieved) and often by distance operators (e.g. "term1 NEAR term2", which will retrieve documents where *term1* is within a specified number of terms from *term2*).

Modern IR systems pay attention not just to the presence or absence of terms but also to their frequency. This gives rise to *term weighting schemes*, whereby every word receives a weight that depends on its frequency across the collection of documents. One of the most widely used schemes is Term Frequency-Inverse Document Frequency or *tf-idf*, that essentially combines the relative term frequency in the document (*tf*) with the inverse frequency of the term in the document collection, i.e the number of documents the term occurs in (*idf*). The actual implementation of this formula varies depending on the task, but the basic formulation is as follows:

$$tfidf(t) = tf(t) * idf(t) \tag{2.2.1}$$

This simple formula is often very useful for many Natural Language Processing tasks (Jurafsky, 2000), as it has the effect of offsetting the weight of a term based on its frequency in a document and in the query against the relative popularity of the term in the document collection. A term that appears in few documents in the collection is deemed more relevant in every document where it does appear than a term that appears in many documents.

A number of approaches exist to creating *document representations* and determining their relevance to a given query, such as the *Vector Space Model (VSM)*. In this introduction we are going to focus on the VSM, in which each document is converted into a vector representation, that is, a vector of terms. These vector representations are generated by situating each document in an n-dimensional space where n is the total number of terms in the vocabulary. The position of this vector in each dimension is based on the number of times that a given term (word) occurs in the document that is being represented, typically weighted by a function such as *tf-idf* that we presented above.

A similarity or ranking function is finally required to determine the relevance of each document to the query. One example is *cosine similarity*, which simply measures the distance between vectors in that multidimensional space (see Formula 2.2.2).

$$sim(q,d) = \frac{V(q) \cdot V(d)}{|V(q)||V(d)|} \tag{2.2.2}$$

While cosine similarity is at the core of many widely used similarity functions, these are typically more complex, particularly with respect to the term weighting. While our experiments do not aim to compare different similarity functions, we would be remiss not to mention other families of similarity functions, such as those based on language models and probabilistic similarity functions, of which Okapi BM25 (Robertson et al., 2009) is a reference point for many others.

The evaluation metrics of IR are also relevant to the evaluation of a CCR system. Two very widely used metrics are:

- *Precision*: the fraction of retrieved documents that are relevant to the query over the total number of retrieved items
- *Recall*: the fraction of relevant documents that are retrieved over the total number of relevant items that should have been retrieved.

There is typically a tradeoff between these metrics: increasing one tends to decrease the other. For this reason, they are typically reported together with their harmonic mean, known as *F-score* or *F1-measure*.

Information Retrieval is a very large field with plenty of research literature available, which goes much beyond the scope of this introduction. Fortunately, a variety of free, sophisticated, robust and open-source IR systems exist which can be used directly as the indexing, storage and retrieval modules in a pipeline, thereby eliminating or significantly reducing the need for specific development in this area. Two main systems are the Apache projects Lucene (Białecki et al., 2012) and Indri (Strohman et al., 2005), the performance of which has been widely studied and established (Turtle et al., 2012). Of these two, Lucene is by far the most widely used in commercial applications[1].

2.3 Lucene and Elasticsearch

Following the requirements outlined in 1.4.3, we have chosen Apache Lucene[2] as the backend retrieval engine for the experiments reported in this thesis. Further, we use Elasticsearch (Gormley and Tong, 2015) as a high-level interface to Lucene. One reason for this is that Elasticsearch provides a REST API over HTTP, allowing us to decouple storage and retrieval from processing, while at the same making it easy to parallelise the code.

Another reason is that, while the Lucene retrieval engine provides an expressive and complex query language, this is made much more accessible and easier to use by the elasticsearch query DSL[3], which represents the query as a nested JSON structure. This structure aims to be human readable and of theoretically unlimited complexity, although in practice it is bounded by the available processing capacity, both in terms of CPU and memory.

Lucene is a field-based retrieval system, which means that each *document* in an index can in fact be thought of as several separate documents. For example, let us consider an index where every document d contains the fields *title*, *abstract* and *body*. Although the API wil present a single index with multiple documents, each of them with multiple fields, in practice any global index metrics (e.g. inverse document frequency, *idf*) will be per-field: the *idf* factor of a given term t will depend on the field f (*title*, *abstract*, or *body*) not the index, so this component is better represented as idf_{tf}.

2.3.1 Query types

Many types of query are provided by the API, which vary in functionality and scope. Listing all the available options would be beyond the scope of this introduction, but some examples of query types that are relevant to the approaches presented in this thesis include:

[1] https://wiki.apache.org/lucene-java/PoweredBy

[2] http://lucene.apache.org/

[3] https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html

Match/term queries: this is the basic matching query. It matches one or more terms in a document, and it is aware of the analyser applied to the document. That is, if the document was lemmatised at indexing time, the query will lemmatise the terms too.

***Match_phrase*:** same as *match* and *term* above, but this query matches only when all the terms are present in the document, in the specified field, and within a given distance. By default, this distance is zero, i.e. they are adjacent, but it can be increased to an arbitrarily large distance.

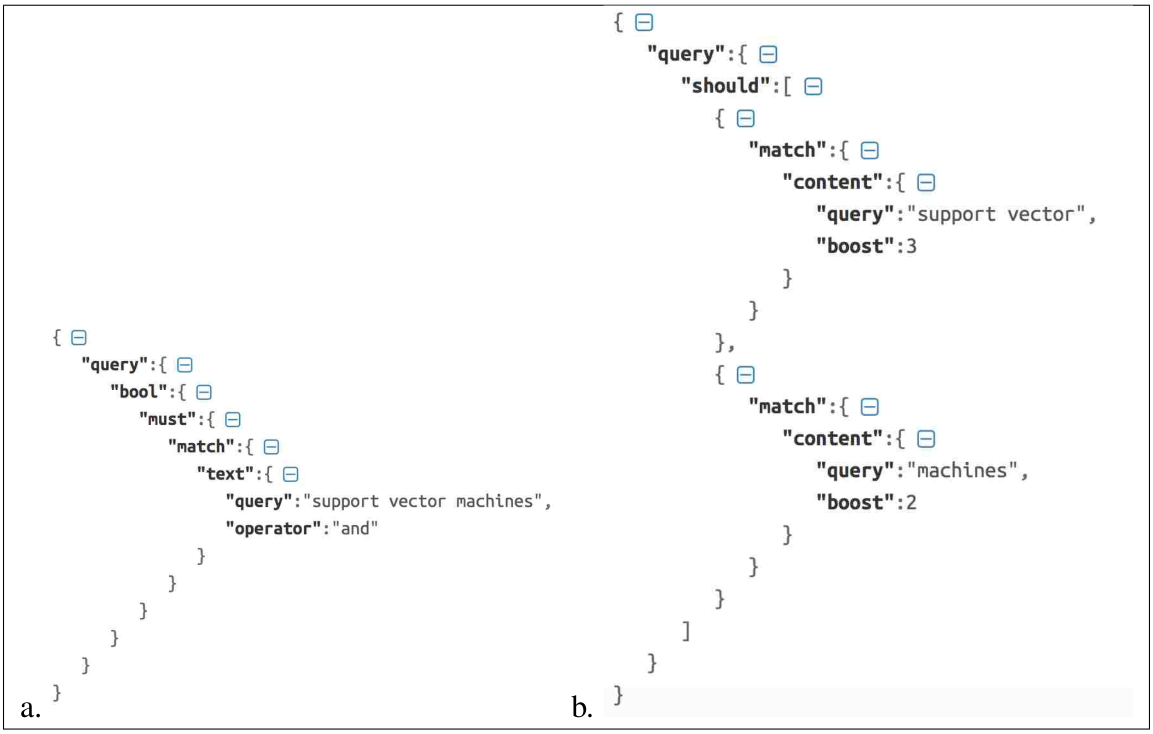
***Bool*:** a boolean operator of the type AND, OR or NOT. By default, all terms or phrases in the query are treated as optional, and when present, their score gets added together. This is the approach we take here.

DisjunctionMax (dis_max): a query that generates the union of documents produced by its subqueries, and that scores each document with the maximum score for that document as produced by any subquery, plus a tie breaking increment for any additional matching subqueries. We use this type of query in Chapter 5, where it becomes key in our experiments.

Figure 2.3.1 includes two simple queries for exemplification. Query *a* is the DSL version of the Lucene query "AND support AND vector AND machines", which means the document must match the three terms or it will not appear in the results. The document's score will be computed by adding together the individual scores of each term. Query *b* is one possible DSL rendition of "support^3 OR vector^3 OR machines^2" which means that if the document matches any of the terms it will appear in the results. Again its score will be computed by adding together the individual term scores, but in this case we are adding manual weights to the terms, thus multiplying their individual scores. The scores of both “support” and “vector” are multiplied by 3, and the score of “machines” by 2. The query containing “support” and “vector” is joined by the default operator, which is OR.

2.3.2 Similarity function

Many standard similarity functions are based on the cosine distance between the query and the document vectors, typically referred to as *cosine similarity*, which we have defined in Formula 2.2.2. Some of these formulas have parameters that can be tuned, but throughout this work we employ the ClassicSimilarity function as implemented in



```
{ ⊟
  "query":{ ⊟
    "bool":{ ⊟
      "must":{ ⊟
        "match":{ ⊟
          "text":{ ⊟
            "query":"support vector machines",
            "operator":"and"
          }
        }
      }
    }
  }
}
```

```
{ ⊟
  "query":{ ⊟
    "should":[ ⊟
      { ⊟
        "match":{ ⊟
          "content":{ ⊟
            "query":"support vector",
            "boost":3
          }
        }
      },
      { ⊟
        "match":{ ⊟
          "content":{ ⊟
            "query":"machines",
            "boost":2
          }
        }
      }
    ]
  }
}
```

Figure 2.3.1: Example Elasticsearch queries.

the Apache Lucene retrieval engine, [4], which allows for no fine tuning. In what follows we introduce this formula and explain its components.

$$score(q,d) = coord(q,d) \cdot \sum_{t \in q} tf(t \in d) \cdot idf(t)^2 \cdot norm(t,d)$$

Figure 2.3.2: The classic Lucene similarity formula (ClassicSimilarity).

In this formula:

- the *coord* term is an absolute multiplier of the number of terms in the query q found in the document d. The presence of a term in the query and the document is enough to add 1 to the multiplier of the overall score.

- tf is the absolute term frequency score of term t in document d. In the practical implementation, this term is in fact the square root of the absolute term frequency score, which means that the presence of a term is more important than how many times it appears.

- $idf(t)$ is the inverse document score. In the practical implementation, $idf(t) =$

[4] http://lucene.apache.org/core/7_0_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html

$1 + log(\frac{numDocs}{docFreq+1})$, where *numDocs* is the length of the set of total documents indexed and *docFreq* is the number of documents in which the term appears.

- *norm* is a normalisation factor that divides the overall score by the length of document d, so that shorter fields contribute more to the score.

As mentioned above, Lucene is a field-based retrieval system, which means that for each document in the index we can store many fields and each field is indexed separately. An important thing to note, therefore, is that all these components in the formula, including *norm* and idf are per-field, not per-document, so a term's score is not normalised over how many documents contain it, but how many contain it in a given field.

2.4 Evaluation method

A very important first problem to tackle for the CCR task is that of evaluation. This system being ultimately intended to help human authors to carry out the specific task of writing a scientific paper, we believe that its ultimate measure of success must be task-based. That is, it should be measured by how much time it saves authors, to what degree it increases productivity or quality of output, or how it impacts user satisfaction. This would involve *user studies* (measuring user satisfaction through explicit ratings) and *online evaluation* (measuring system performance through implicit metrics, such as Click-Through-Rates).

However, in order to carry out task-based evaluation, the system must first be built to a significant level of functionality, which makes this type of evaluation not possible to implement during its initial development. We should then differentiate between two main stages of work on the system: development and production-ready. This thesis focuses on the early development stage, during which it is very useful to have the ability to automatically assess the impact in efficiency of different algorithms.

For this reason, most previous work in this area has employed *offline evaluation*, where the performance of the system is measured as its *accuracy* based on some ground truth annotation of the relevance of the recommended papers to the given input.

The ultimate measure of relevance of the recommended papers must still come from human judgement. Asking humans to directly rate the output or performance of the system is a common measure of evaluation, which however has a particularly low utility/cost ratio, as every time evaluation needs to be carried out, the same costly process needs to be carried out. Further, it could introduce potential statistical problems, when testing with different populations, and lead to unreplicable results.

An approach that is similar in capturing these judgments but more desirable from a utility-cost perspective is capturing human relevance judgments as annotations to a corpus. This has the advantage of allowing for repeated automatic testing against this set of annotations and is the standard way of evaluating IR systems, known as building a *test collection* (Sanderson, 2010). This consists in building by hand a corpus of a set of queries and a set of relevance judgements for each query against which to measure the performance of the system.

However, annotation is an arduous effort that requires considerable manual input and very careful preparation, so we wish to avoid it altogether if possible. Mining existing resources for these human judgements is then highly desirable. Fortunately there is no lack of data that meets the requirements for this task: every scientific paper contains human ”judgements” in the form of citations to other papers which are contextually appropriate: that is, relevant to specific passages of the document and aligned with its argumentative structure.

There are many advantages to this type of evaluation: as it only considers papers for which the full textual content is available in the document collection, we can directly connect the citation token’s context to the contents of the cited paper. Therefore this approach provides a set of readily available “gold standard” existing human judgments of the relevance of a paper to a context.

There are also disadvantages: it is highly likely that other documents exist in the collection that the original author was unaware of and may have been better matches than the existing citations.

Also, it is likely the language used in the context around a citation is primed by the text in the cited document, often copied literally. As we proposed before, the ideal way to evaluate this task is extrinsically and task-based, therefore directly in a production system, with user interaction data, which is not an option at this stage.

2.5 The task: Citation resolution

With the aim of removing the need for purpose-specific annotation, we follow a common approach in the literature and evaluate the performance of our recommendation against existing scientific publications. We substitute all citations in the text with *citation placeholders* and make it our task to match each placeholder with the correct reference that was originally cited. We call these originally cited documents *tar-*

get documents. We only consider *resolvable citations*, that is, citations to references that point to a document that is in our collection, which means we have access to its full machine-readable contents (*collection-internal references*). We call this evaluation method *citation resolution*.

The core criterion of this task is to use existing human judgements that we have clear evidence for, in the form of contextually relevant citations. We use these judgements for evaluation: our task is to rank all documents in our document collection D according to their relevance score to the query q which is extracted from the context C.

The algorithm for the task is presented in Figure 2.5.1. First we split our corpus into a document collection D and a test set T (1) and we create an index of the documents in D (2). The document collection is the set of potential documents to recommend: we index these and extract our evaluation queries only from contexts around citations in the documents in the test set.

For any given *test document* (3), we first extract all the citation tokens found in the text that correspond to a collection-internal reference (a). We then create a *document representation* of the referenced document, a vector in a high-dimensional space where each dimension represents the presence of a unique term in the document. This representation can be based on any information found in the document collection, excluding the document d itself, such as the text of the referenced document and the text of documents that cite it.

For each citation token we then extract its context (see 3.b.i in Figure 2.5.1), from which we generate the *query* in IR terms. We then attempt to *resolve* the citation by computing a score for the match between each reference representation and the citation context (3.b.ii). We rank all collection-internal references by this score in decreasing order, aiming for the target documents to appear at as low a rank as possible (3.b.iii).

In the case where multiple citations share the same context, that is, they are made in direct succession (e.g. *"...compared with previous approaches (Author (2005), Author and Author (2007))"*), the first n elements of the list of suggested documents all count as the first element. That is, if any of the references in a multiple citation of n elements appears in the first n positions of the list of suggestions, it counts as a successful resolution and receives a score of 1. The final score is averaged over all citation contexts processed.

```
1. Split corpus into document collection $D$ and test set $T$
2. Create index of all documents in $D$
3. For test document $d \in T$
    (a) For every reference $r$ in its bibliography $R_d$
        i. If $r$ is in document collection $D$
        ii. Add all inline citations $C_r$ in $d$ to list $C$
    (b) For each citation $c$ in $C$
        i. Extract context $ctx_c$ of $c$, and from it generate the query $q \in Q$
        ii. Rank all documents in $D$ that match the query $q$ by the similarity
            between query and document $sim(q,d)$
        iii. Measure score based on the rank of the original reference $r_c$ in the
             list of results $L$
```

Figure 2.5.1: Algorithm for citation resolution. We measure how well we did at our task using the rank at which the target document appears in the list of ranked retrieval results.

2.6 Evaluation metrics

Most evaluation metrics employed for this task are based on the rank at which the originally cited collection-internal document (henceforth *cited document*) appears (Konstan and Riedl, 2012). Different scores are often computed based on this. Some relevant and widely used ones include:

- *Top-1 Accuracy*: this is a binary measure, that only evaluates whether the cited document appears at rank 1 (i.e. the highest scoring result). If it does appear at rank 1, the score for that query is 1, or 0 otherwise.

- *Mean Reciprocal Rank* (MRR) : This is statistical measure, which is evaluated as the multiplicative inverse of the rank of the first correct answer. As in our experiments there is normally one correct answer, this evaluates to one over that rank. MRR is averaged over all evaluated queries. In Formula 2.6.1, Q is a set of queries and $rank_i$ is the rank of the first relevant document to the query Q_i.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{2.6.1}$$

- *Discounted Cumulative Gain* (DCG): Similar to MRR, a measure based on the rank of the target document in the list of suggested references, where the query score decreases logarithmically rather than linearly (Wang et al., 2013). In this formula (2.6.2), p is a particular position in the list of results and rel_i is the graded relevance of the result at position i.

$$DCG = \sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)} \tag{2.6.2}$$

- *Normalised Discounted Cumulative Gain* (NDCG): This measure is computed by normalising the DCG score by the Ideal Discounted Cumulative Gain (IDCG), the maximum DCG score possible for that query, where all relevant documents are retrieved at their ideal ranks. When dealing with a single relevant document that should always be in first position, the IDCG is $1/log(2)$. This is the normalising factor for DCG in our setup. As is the case with other IR metrics, given that a cut-off is applied to the list of retrieved results and only the top-K are considered, NDCG is most correctly reported at a rank, i.e. NDCG@K. In our case, $K = 200$ so all reported results are NDCG@200[5]. The IDCG at position p is the maximum possible DCG score at that position. The NDCG at position p is the actual DCG score normalised by the ideal score (IDCG)

$$IDCG_p = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{log(i+1)} \tag{2.6.3}$$

$$NDCG_p = \frac{DCG_p}{IDCG_p} \tag{2.6.4}$$

- *Recall*: While standard to IR, recall is harder to define when we have a single document as the ground truth for a given query. In related work it is computed as the presence of the original reference in the list of suggestions generated by the system, which only returns the top-K scoring documents, and therefore it is Recall@K.

For our task, our ground truth is a very small set of human-authored citations, typically a single one. This means that measuring Recall as the one citation appearing within a rank cut-off adds very little information over the other metrics. Indeed, it has been shown that metrics used in previous work such as NDCG, Precision, Recall and F-measure are highly correlated (Beel et al., 2016b). NDCG is a popular IR metric given

[5] This value is in the range of values previously reported in the liturature (e.g. He et al. (2010)) and it serves the purpose of making it feasible to implement the experiments we report in Chapter 5).

its smooth discounting over ranks and so its satisfactory scaling to retrieving large result sets from large collections. Much of the most relevant previous work employs it, so we therefore choose NDCG as our main metric too.

2.7 Caveats of our evaluation method

The evaluation method and dataset crucially determines the architecture of the system. Here we are employing the same evaluation method as most of the literature on this topic (e.g. He et al. (2010); Huang et al. (2014)), which is based on retrieving the citations that were originally employed in papers that have already been published. This offers the advantage of a large corpus with explicit pre-existing human annotation that we can treat as a gold standard.

However, an important caveat is due here: this task is not truly representative of the task that the potential future system would be applied to. When authors cite other papers, it is frequent that they have read them to some depth (even if, say, just the abstract) and this primes their lexical choice when summarising the merits and relevance of the work being cited. Often full expressions or even sentences are copied from the work, which makes it rather easy to obtain high scores through ranking by an IR system.

It is likely that an evaluation carried out on a productised CCR System with live users would highlight different areas that require improvement. One essential capability of a search engine such as Google Search[6] is to recognise synonyms and retrieve pages containing them (Manning et al., 2008). For example, the queries “edinburgh jobs developer”, “edinburgh jobs software engineer” and “edinburgh jobs programmer” yield similar top level results and the synonyms are highlighted in the snippets for each result.

While the limitations of this evaluation method seem to have been largely ignored in the CCR literature so far, we attempt to mitigate the effect of having a corpus of already published papers and not of draft papers with relevance judgments by on the one hand, clearly separating the document collection from test set and on the other, excluding self citation from resolvable citations. We do not consider finding one’s own work is a realistic use case to evaluate against.

In all of the experiments reported here we apply a cut-off of $k = 200$ and retrieve the top-k results. Therefore, if the originally cited collection-internal document does not

[6] https://www.google.com/

figure among the top 200 results, the score for that query will be 0. Applying this cut-off significantly speeds up evaluation, which will become absolutely necessary in the experiments reported in Chapter 5. At the same time, this allows us to better compare the results between sets of experiments.

One more thing to note is that multi-citations (citation sites where several citations are adjacent in the same sentence) are evaluated in the following way. If a correct paper is retrieved at any rank that is lower than the total number of citation tokens in the citation site, it is counted as having rank 1, that is, having been retrieved in first position and therefore getting maximum NDCG score. This is because the list of citation tokens in a citation site is typically arbitrary (e.g. alphabetical or chronological) and no citation token is necessarily more relevant than another one to the given context.

2.8 Recommender systems

There are many avenues for enquiry in this research project, as it is driven by its application to recommend contextually relevant literature to academic authors. Plentiful previous work related to this application exists, particularly in the domain of *recommender systems*, which significantly overlaps in aims and scope and often in approaches with *academic search engines*.

In a thorough survey of the field until 2013, Beel et al. (2016b) recognise seven different classes of approaches to academic recommender systems, of which we find these six relevant to our own research:

- *Collaborative filtering (CF)*: This approach is based on the assumption that like-minded users like the same items. A model is built of each user, and the similarity between users is established based on the items they have each rated. The missing entries for items that a user has not rated in the large user-item matrix are inferred from other users that are deemed similar. The advantage of CF is that it is independent of content: the similarity between items is established through the similarity between users and their preferences and interests. As we will see later, CF can be used as part of a hybrid system, but in essence this approach is very far from our application, since we deal directly with the contents of a document.

- *Content-based filtering (CBF)*. Of all of these approaches, CBF is the most similar one to the task we undertake here. In CBF, the interests of users are inferred from the items they interact with. These items are represented as vectors of features and weights. When the item is textual in nature, it is common for the features to be based on the presence and counts of words and n-grams in the document. It is common for CBF to employ the Vector Space Model to assess the similarity of items and users, and it is also common to use tf-idf as a weighting scheme. This makes it clear that CBF and Information Retrieval overlap hugely in techniques and application. The main difference is that CBF deals with users and their histories of interactions with the items in the collections to recommend, whereas IR in its vanilla form just takes keywords directly from the users.

- *Co-occurence*: Conceptually and algorithmically this is quite similar to CF, the differences being that no specific user model is built and similarity between items is established based on their co-occurences. The common way to establish co-ocurrence in academic papers is through co-citation. Small (1973) proposed that the more frequently two papers are cited together, the more *related* they are to each other. Relatedness is markedly different from *similarity*, which again connects to CF in that no intrinsic document features are required for this comparison. More fine-grained approaches may look beyond the metadata of the paper, and actually measure the proximity of co-citations in running text (Gipp and Beel, 2009).

- *Graph-based*: This approach is based on the connections inherent in academic publishing, particularly the explicit links between papers in the shape of citations. However, other nodes on a graph can be authors, venues, institutions, years, and in specific domains they could be objects of study (e.g. genes and proteins), methods, etc. The links between these nodes can be built from citations, "published in" relations, authorship or relatedness between objects of study.

- *Global Relevance*: A simple approach to a recommender system is to avoid user modelling and recommend to every user the items with the highest global relevance. In the case of academic papers, the most immediate and naïve translation of this is to recommend the papers with the highest citation count. However, more sophisticated metrics of global relevance exist, including PageRank (Page et al., 1999) and HITS (Kleinberg, 1999), specific citation counts of venues and authors, authors' affiliations, h-index and citation count, as well as recency of articles or venue type.

- *Hybrid recommendation*: Ultimately, it is indisputable that in a practical setting all of these approaches can contribute towards better recommendations. Hybrid approaches combine two or more of those presented above.

Our research is firmly based on Information Retrieval techniques, but conceptually, and because of its application, there is significant overlap with CBF recommender systems. While our approach most closely aligns with CBF, it is quite likely that a hybrid system would be desirable. Integrating other approaches such as graph-based metrics (e.g. "other papers by this author") or co-occurence (i.e. "people who cited this also cited...") would likely improve the relevance of recommendations, or at least save the author time and increase user satisfaction. However, given time and resource constraints, we choose to focus on researching the least explored areas where we can make the most significant contribution and assume that when the time came to implement and productise a system, a number of other well-studied approaches would be of use.

2.9 Research aspects

In order to give some structure to the research landscape of our narrow field (IR-based CCR) and its nearest neighbours, we are presenting previous approaches according to several *aspects* of the research. These could be chosen in many ways, but quite a natural division emerges if we follow the different steps of an information retrieval application: collecting the text from documents, indexing these documents and using queries to retrieve documents from this index, for which we use a similarity metric between query and document.

Documents in the document collection must be indexed for retrieval: an index must be built according to the presence of terms in documents in the collection. However, before we do this, we need to choose what parts of the text inside the document or inside the collection we should index and how. We refer to all the processing done to the raw document, to the selection of relevant text, and to the processing of this text, as well as choosing in what fields to index it and how, as *document representation*.

Once the index has been built from this text, queries can be run against it, where each document will be measured against the query using a *similarity metric*, which can conceptually be framed as a variety of functions. This can include employing learned models, such as in a learn-to-rank approach (Liu et al., 2009) or using any function that can be used for re-ranking, but is commonly a version of tf-idf-weighted cosine similarity as defined in Section 2.3.2. Very importantly, in our task these queries do not come pre-formed by the user, but instead we aim to extract the relevant context automatically from the context of a citation placeholder and from it to generate a query, which we refer to as *query extraction* here.

2.9.1 Document representation

Within the *document representation* step, there are several aspects to consider: what text from a document should be indexed, what fields this text should be indexed in, what weighting scheme should be applied, what type of analysis should be applied to this text, etc.

While all documents are indexed using a bag-of-words approach, deciding what information from the document to include and how to weight it is a key aspect. As further detailed in Section 2.10, we distinguish between:

- *internal representations* (any subset of the text that a document contains, e.g. *full text* vs *title and abstract*),

- *external representations* (a bag-of-words formed of all contexts of citations to this document found in other documents in the collection that are not part of the test set) and

- *mixed representations* (a concatenation of both internal and external ones).

Some previous work (He et al., 2010) makes use of the text around existing citations inside a source document to represent this document, so we need to make a distinction here between:

- The *incoming link contexts*, which are the collection of contexts around citations to a document d contained in a set of documents D. Collectively, we refer to all text extracted from these contexts and concatenated as the *inlink context* of document d.

- The *outgoing link contexts*: the collection of contexts of citations inside a document d to a set of documents D. We refer to all of the text extracted from these contexts, once concatenated, as the *outlink context* of document d.

We explore the usefulness of internal, external and mixed representations and their parameters in Chapter 3. Beyond this, we also carry out experiments with document representations where different sentences in a document would be indexed in different fields based on an automatic classification of those sentences into a number of rhetorically-motivated classes. This work is detailed in Chapter 4 and Chapter 5.

2.9.2 Similarity metric

Much related work seems to concern itself primarily with the way the similarity between queries and documents is measured. Several approaches and methodologies for computing the similarities between query and document have been applied and evaluated. We examine some of these in more detail in section 2.10, but some examples are: Restricted Boltzmann Machines for topic discovery (Tang and Zhang, 2009), topic models (Kataria et al., 2010), translation models (He et al., 2012), neural methods (Huang et al., 2015a, 2014), plus of course variants of cosine similarity applied to different types of vector representations.

In this work we do not aim to define a new similarity metric, but to use existing algorithms and metrics that are already implemented, and that are already provided by the Lucene retrieval engine. What we investigate is how to leverage these well-understood similarity functions to address these objectives:

1. Increase the relevance of recommendations by choosing how to build the bag-of-words document representation (Chapter 3).

2. Select the optimal context window from which to extract the query (Chapter 3).

3. Improve relevance by applying scientific discourse annotation schemes (see Chapter 4 and Chapter 5).

4. Extract a better performing query (i.e. set of keywords and their weights) from the insertion context (Chapter 6).

2.9.3 Query extraction

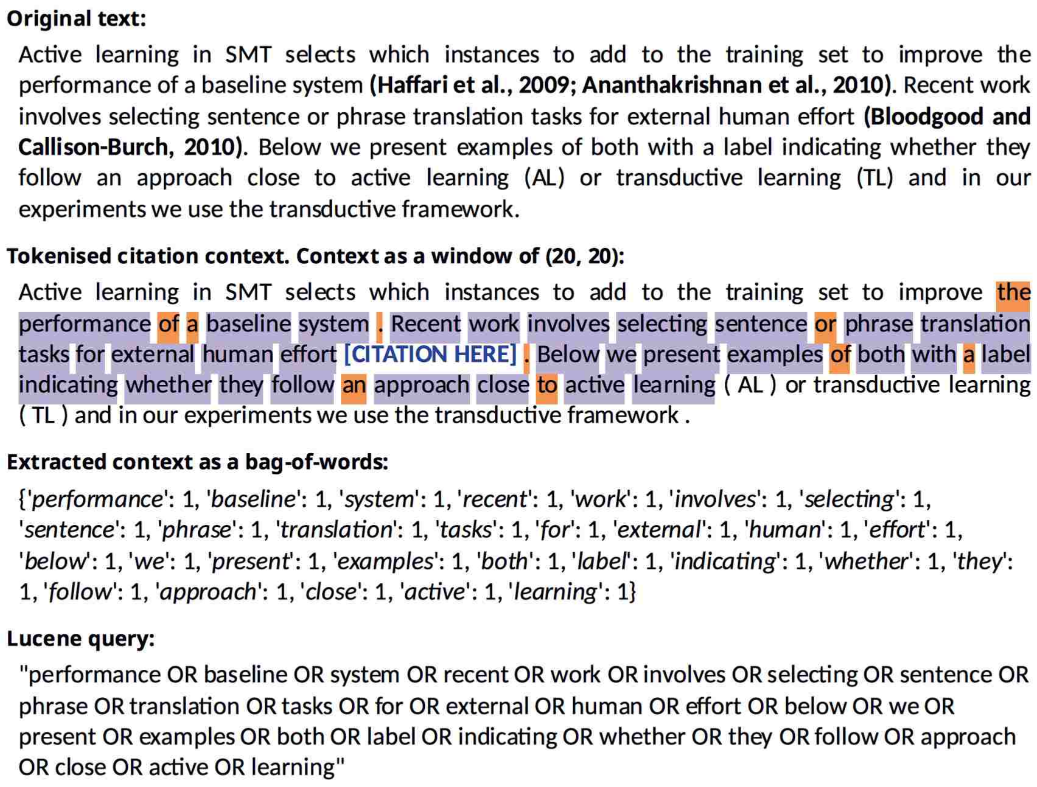
Using already published papers to evaluate the performance of a CCR System during development requires treating the textual context of an existing citation as the input query, and the extraction of this query offers many options for experimentation. Generating a query from the context of a citation placeholder is in fact one of the most important tasks. Conceptually we can distinguish two sub-tasks: *context extraction* and *query generation*.

First we need to define what the *context* of a citation is in order to extract it. In all related work and in the research presented herein, this context is exclusively textual in nature. Within this, what spans of text in a document are relevant to a given citation site has been explored by previous work in the literature.

One option is to use a window of words around the citation: this means up to *w* words (tokens) left and right of the place in the text where the citation would appear, which produces a list of word tokens that is used as the query. This is a frequently employed technique (He et al., 2010; Bradshaw, 2003; Kataria et al., 2010), although Ritchie (2009) proposes it may be too simplistic a method. Other methods have been tested, e.g. full sentence extraction (He et al., 2012), or windows over sentences instead of words, such as 1 sentence before, the citing sentence and 1 after (Huang et al., 2015a). Ritchie et al. (2008) compare these window methods for context extraction but for the purpose of generating external document representations, which we discuss below in Section 2.10.1.2.

Query generation seems to have been generally overlooked in the literature and seldom if ever treated as a separate step. Quite often the context is treated as a bag of words, and converting this into a query is a simple transformation of removing stopwords and punctuation and removing infrequent words in the corpus (e.g. Kataria et al. (2010)).

Figure 2.9.1 provides an example of extracting a simple evaluation query from the context of a citation placeholder using a symmetric window over tokens of 20 before and 20 after, and filtering stopwords from a basic list of most common 10 stopwords.



Original text:

Active learning in SMT selects which instances to add to the training set to improve the performance of a baseline system **(Haffari et al., 2009; Ananthakrishnan et al., 2010)**. Recent work involves selecting sentence or phrase translation tasks for external human effort **(Bloodgood and Callison-Burch, 2010)**. Below we present examples of both with a label indicating whether they follow an approach close to active learning (AL) or transductive learning (TL) and in our experiments we use the transductive framework.

Tokenised citation context. Context as a window of (20, 20):

Active learning in SMT selects which instances to add to the training set to improve the performance of a baseline system . Recent work involves selecting sentence or phrase translation tasks for external human effort [CITATION HERE] . Below we present examples of both with a label indicating whether they follow an approach close to active learning (AL) or transductive learning (TL) and in our experiments we use the transductive framework .

Extracted context as a bag-of-words:

{'*performance*': 1, '*baseline*': 1, '*system*': 1, '*recent*': 1, '*work*': 1, '*involves*': 1, '*selecting*': 1, '*sentence*': 1, '*phrase*': 1, '*translation*': 1, '*tasks*': 1, '*for*': 1, '*external*': 1, '*human*': 1, '*effort*': 1, '*below*': 1, '*we*': 1, '*present*': 1, '*examples*': 1, '*both*': 1, '*label*': 1, '*indicating*': 1, '*whether*': 1, '*they*': 1, '*follow*': 1, '*approach*': 1, '*close*': 1, '*active*': 1, '*learning*': 1}

Lucene query:

"performance OR baseline OR system OR recent OR work OR involves OR selecting OR sentence OR phrase OR translation OR tasks OR for OR external OR human OR effort OR below OR we OR present OR examples OR both OR label OR indicating OR whether OR they OR follow OR approach OR close OR active OR learning"

Figure 2.9.1: Simple example of context extraction and query generation. Shaded in light purple, the tokens within a context window of 20 tokens before and 20 tokens after. Shaded in orange, the stopwords and punctuation removed from this context.

2.10 Previous work on CCR

A prime example of an IR-based CCR system, which has influenced much other research, and certainly our own, is the line of work best exemplified by He et al. (2010). In the paper, the authors in the research group behind the academic search engine CiteSeerX[7] detail the design and evaluation of a CCR system that uses the CiteSeerX dataset (in its state at the time). This system makes use of the information retrieval techniques outlined above and uses the service's corpus, which consisted of 456,787 documents. These documents were automatically converted from a print-ready format (mostly PDF) to a plain-text representation by the production CiteSeerX pipeline at the time.

In this particular article they do not make use of the standard user-facing CiteSeer setup, which employs Lucene as a retrieval backend through the Apache Solr

[7] http://citeseerx.ist.psu.edu/

interface[8]. Instead, they define their own Context-Based Relevance Model (CRM for short), which is based on comparisons between density matrices of concepts (terms) and documents. They also implement their own backend for computing the term-document density matrices in the corpus, also employing tf-idf as a weighting scheme, but computing their own probability-based similarity metric. The system they present is hybrid, employing several measures of similarity.

Due to the size of the collection, and so with scalability in mind, the system of He et al. (2010) proceeds in two stages. The first stage collects a set of candidates from which to choose the specific suggestions for document *d*, and the second chooses the best fit for each local context. They propose and test a number of methods for retrieving the candidate set, which includes Global Recommendation (finding the top N documents with Title and Abstract most similar to the query manuscript), Author (documents that share authors with the query manuscript), CitHop (papers cited by the papers already in the candidate set), Local Recommendation (top N papers whose incoming link contexts are most similar to the outgoing link context), papers containing the top-N most similar outlink contexts to the current context, and hybrid methods which combine these approaches.

We do not follow this two-step approach and instead aim to directly recommend the most relevant papers from the full collection, which is enabled by the speed of execution of a productised and mature retrieval backend. At the same time, we do not use the context of citation inside a document to describe that document itself, i.e. the outgoing link contexts.

Other relevant work to ours is reported in Ritchie et al. (2008), although the task is not framed as CCR but as a standard search task, where queries are provided by the user, and uses a different evaluation method. While also using a conversion of the ACL anthology for their experiments, their approach to evaluation was to manually build a *test collection*, a small set of 82 queries together with manually annotated relevance judgements for the documents to be returned (11.4 on average per query). Much of their work deals with how to build representations of in-collection documents, with particular attention on extracting anchor text from citing documents: what we label *external methods*. They focus on two aspects for comparison and exploration: 1. different window sizes for extracting the text from incoming link contexts and 2. using their text collection for evaluating several standard similarity metrics provided by the Indri retrieval framework (Okapi, Cosine, Indri, KL-divergence and KL-divergence

[8] http://lucene.apache.org/solr/

with relevance feedback). The former (1) significantly overlaps with our experiments in Chapter 3 and Chapter 5, where we explore these different parameters among others for our own task and for two different corpora. The latter (2) falls outside of the scope of our own work: while there are differences in the absolute performance of different standard similarity metrics in different scenarios, this is a parameter we keep fixed during our experiments.

Going beyond widely used IR similarity functions, the main focus of many examples of previous work in this area has in fact been on exploring alternative ways of measuring the similarity between queries and papers in the collection. One example of a family of approaches employs *topic models*, which are unsupervised algorithms for discovering the main themes (topics) that pervade large collections of documents (Blei, 2012). Topic models are generative models that aim to retrieve some latent structure in an otherwise unorganised collection by modelling the co-occurrence patterns of words in a collection of documents. Some aim to model the similarity of words based on their co-occurrence, such as Latent Semantic Analysis (LSA, Landauer (2006)). Others aim to learn distribution of words per topic and topics per document, such as Latent Dirichlet Allocation (LDA, Blei et al. (2003)), and so are often employed for clustering documents in collections. Kataria et al. (2010), and before them, Nallapati et al. (2008), use the topic-word and topic-citation multinomial distributions for what they call *link prediction*, which differs little if at all from the contextual recommendation task we undertake. They employ variants of PLSA (Hofmann, 1999) and LDA.

An earlier approach to recommending research papers based on common topics (Tang and Zhang, 2009) employs a 2-layer Restricted Boltzmann Machine (RBM) instead of LDA for discovering these latent topics. They set the number of topics to T=200 and compute a topic representation of each citation context, and then compute KL divergence to measure the relevance between the in-collection paper and the query. The query extraction they employ is simply to take the citing sentence, removing stopwords, numbers and infrequent words.

Approaches to document clustering based on topic models have been successfully applied in several tasks. They are unsupervised or lightly supervised approaches, with the advantage of a low need for annotation. However, clustering similar documents is not at the immediate heart of our task and further, the number of topics is a fixed parameter, which remains a research question and a limitation. While topic modelling may prove useful as part of a hybrid recommender system, our research is focused on the IR formulation of this task.

Another set of related approaches employ translation models. Huang et al. (2012) consider the textual context of a citation token as the “descriptive language”, which they want to learn to *translate* into the “reference language”, which is made of tokens that represent the IDs of papers in the collection. Before them, Lu et al. (2011) used a more intuitive approach, where they used a simple phrase-based translation model trained with expectation-maximisation (EM) to translate between a citation’s context and the contents of the cited document. They then used this model to estimate the likelihood of a document given a context and so to rank the documents in the collection.

	Corpus	**Document representation**	**Context extraction**	**Similarity**
Ritchie et al. (2008)	~9k (ACL)	Full text + Inlink contexts (Window (50, 75, 100), Sentences (1, 3))	Manual annotation (using a test collection)	Indri: Okapi, Cosine, KL divergence, KL + relevance feedback
Tang and Zhang (2009)	1.6k NIPS papers 3.3k (CiteSeer)	Full text	Citing sentence	2-layer RBM (200 topics) KL divergence
He et al. (2010)	450k (CiteSeer)	Title + abstract Outlink contexts Inlink contexts	Window (50)	Links in graph (authors, citations) + co-cited probability + cosine distance
Kataria et al. (2010)	~3k (CiteSeer) ~3k (Webkb)	Full text Inlink contexts (Window (30))	Window (30)	Topic models (cite-PLSA-LDA)
He et al. (2012)	30k	Passage (500, 1000, 2000, 3000), 4/5 overlapping	Citing sentence	Translation model (contexts to reference IDs)
Huang et al. (2015a)	> 1 million (CiteSeer)	In-link contexts (Sentences (3))	Sentences (3; 1 up, 1 down)	Cosine distance + neural word embeddings

Table 2.10.1: A grid comparison of previous work: the corpora and the number of documents employed, and the methods used for document representation, context extraction and similarity metrics. The implementation details of each published work are too complex to summarise here: this table only aims to present an overview and a guide for comparison with relevant previous work.

In recent years there has been a significant shift in the types of similarity function employed for many NLP applicatons, with neural approaches taking front stage.

A major reason for this shift is the emergence of successful algorithms for learning *distributional representations* of words, of which *word2vec* (Mikolov et al., 2013) is perhaps the most famous instance. This family of approaches aims to represent the meaning of every word as a vector in a high-dimensional latent space, where the vectors of related words will have a small cosine distance between them. The relation between words is very loosely defined, as these vectors are learned simply from the co-occurrence of words in text, using an efficient algorithm that makes training possible on very large amounts of data. This means that a word stops being treated as a single number (a scalar) and the "similarity" between words becomes computable directly from data.

Using these approaches, Huang et al. (2015a) present a neural probabilistic model for CCR that jointly learns semantic representations for citation contexts and cited papers, and train it by maximising the likelihood of a paper given a context. They compare their results to several baselines based on previous approaches, including Tang and Zhang (2009), Kataria et al. (2010) and Huang et al. (2012) that we have mentioned above. They report small but statistically significant improvements in scores over Huang et al. (2012), which also by far outperforms the others. While this is a very strong line of work, and certainly representative of the types of approaches favoured in the recent literature, here we aim to investigate more traditional NLP approaches applied to information retrieval.

Table 2.10.1 presents the previous work we have reviewed above, according to the corpora used and the research aspects we considered in Section 2.9. As we have seen in this section, we find the largest variety of research focusing on the similarity metrics employed, while the methods for extracting the context of a citation placeholder, for generating the query and for creating a document representation are normally fixed for each published article, although they are clearly key to the functioning of the system.

As we have seen, previous work has used both the contents of the target document and the incoming link contexts, which we defined in Section 2.9.1 as internal representations and external representations. It is now time to have a deeper look at how these are created.

2.10.1 Building a document representation

In order to judge how similar each document is to the input query, a vector-space document representation must be created for each document in the collection. There are typically two main sources from which to obtain the text from which to create the document representation: *internal*, that is, the text of the document itself, and *external*, which is the text of other documents that link to it. CCR efforts to date have heavily focused on external representations of documents, exploiting the context in which an in-text citation occurs as the equivalence of *anchor text* in HTML documents.

2.10.1.1 Internal document representations

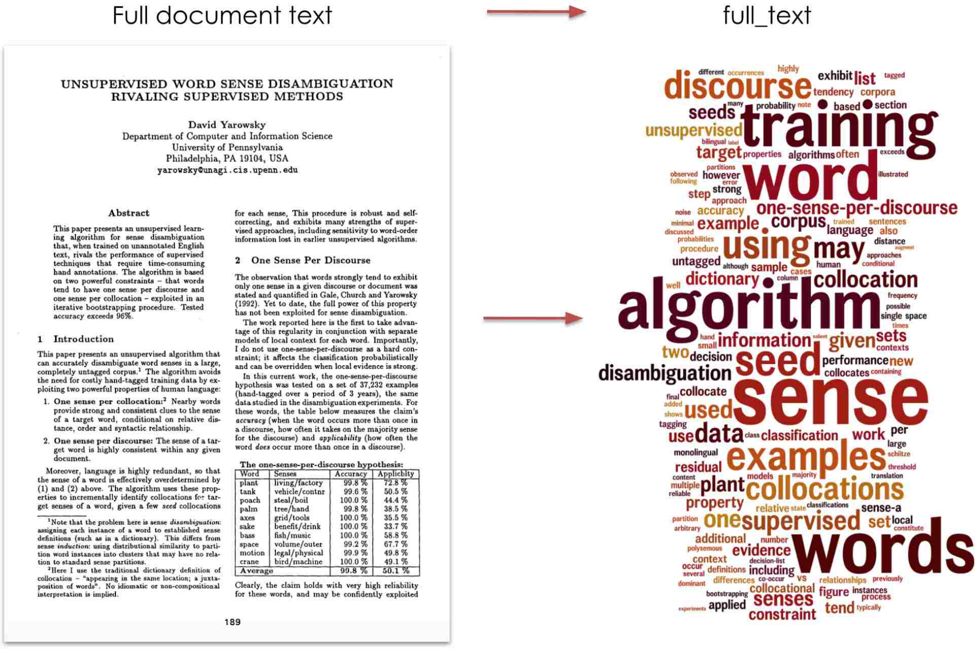
In previous work it has been reported that when building a document representation for an IR system for scientific documents, using select chunks or sections of a document leads to improvements in several tasks. For instance, the experiments of Jimeno-Yepes et al. (2013) suggest that automatically generated summaries lead to similar recall and better indexing precision than full-text articles for a keyword-based indexing task.

While those summaries can be automatically generated, a first intuition is that a scientific paper is already expected to contain what is arguably a summary of its contents in the Abstract. In fact, the Abstract is itself often directly generated as a summary of the document by the human author.

However, the purpose of an Abstract is manifold; it is also a piece intended to gain the reader's attention and persuade them to read on (Hyland, 2009). Also, it is immediately intuitive that this chunk of the document is unlikely to contain all the information that a CCR system would require. For instance, Gay et al. (2005) report that using specific parts of the article text to build a Vector Space Model document representation improves the results of their indexing task by 7.4% over just using the title and the abstract. They obtain the best results using a combination of "the sections Results, Results and Discussion, and Conclusions together with the article's title and abstract, the captions of tables and figures, and sections that have no titles" (sic.).

Going beyond purely positional and structural features, the focused summarisation of a scientific paper could benefit from the classification of different parts of it according to their rhetorical function within the article. Argumentative Zoning has already been applied to content selection for generating user-readable short summaries of scientific articles (Teufel and Moens, 2002), but never before to building a document representation for indexing for a CCR task. We explore this in Chapters 4 and 5.

One of the representations we use in our experiments is full_text, which is represented visually in Figure 2.10.1.



UNSUPERVISED WORD SENSE DISAMBIGUATION
RIVALING SUPERVISED METHODS

David Yarowsky
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA
yarowsky@unagi.cis.upenn.edu

Abstract

This paper presents an unsupervised learning algorithm for sense disambiguation that, when trained on unannotated English text, rivals the performance of supervised techniques that require time-consuming hand annotations. The algorithm is based on two powerful constraints – that words tend to have one sense per discourse and one sense per collocation – exploited in an iterative bootstrapping procedure. Tested accuracy exceeds 96%.

1 Introduction

This paper presents an unsupervised algorithm that can accurately disambiguate word senses in a large, completely untagged corpus.[1] The algorithm avoids the need for costly hand-tagged training data by exploiting two powerful properties of human language:

1. **One sense per collocation:**[2] Nearby words provide strong and consistent clues to the sense of a target word, conditional on relative distance, order and syntactic relationship.
2. **One sense per discourse:** The sense of a target word is highly consistent within any given document.

Moreover, language is highly redundant, so that the sense of a word is effectively overdetermined by (1) and (2) above. The algorithm uses these properties to incrementally identify collocations fo: target senses of a word, given a few *seed* collocations for each sense, This procedure is robust and self-correcting, and exhibits many strengths of supervised approaches, including sensitivity to word-order information lost in earlier unsupervised algorithms.

2 One Sense Per Discourse

The observation that words strongly tend to exhibit only one sense in a given discourse or document was stated and quantified in Gale, Church and Yarowsky (1992). Yet to date, the full power of this property has not been exploited for sense disambiguation.

The work reported here is the first to take advantage of this regularity in conjunction with separate models of local context for each word. Importantly, I do not use one-sense-per-discourse as a hard constraint; it affects the classification probabilistically and can be overridden when local evidence is strong.

In this current work, the one-sense-per-discourse hypothesis was tested on a set of 37,232 examples (hand-tagged over a period of 3 years), the same data studied in the disambiguation experiments. For these words, the table below measures the claim's *accuracy* (when the word occurs more than once in a discourse, how often it takes on the majority sense for the discourse) and *applicability* (how often the word *does* occur more than once in a discourse).

The one-sense-per-discourse hypothesis:

Word	Senses	Accuracy	Applicblty
plant	living/factory	99.8 %	72.8 %
tank	vehicle/contnr	99.6 %	50.5 %
poach	steal/boil	100.0 %	44.4 %
palm	tree/hand	99.8 %	38.5 %
axes	grid/tools	100.0 %	35.5 %
sake	benefit/drink	100.0 %	33.7 %
bass	fish/music	100.0 %	58.8 %
space	volume/outer	99.2 %	67.7 %
motion	legal/physical	99.9 %	49.8 %
crane	bird/machine	100.0 %	49.1 %
Average		99.8 %	50.1 %

Clearly, the claim holds with very high reliability for these words, and may be confidently exploited

[1] Note that the problem here is sense *disambiguation*: assigning each instance of a word to established sense definitions (such as in a dictionary). This differs from sense *induction*: using distributional similarity to partition word instances into clusters that may have no relation to standard sense partitions.

[2] Here I use the traditional dictionary definition of collocation – "appearing in the same location; a juxtaposition of words". No idiomatic or non-compositional interpretation is implied.

189

Figure 2.10.1: Illustration of a bag-of-words internal document representation as a word cloud.

2.10.1.2 External document representations

When citing previous work, it is often the case that an extremely short summary of it (often one sentence or less) is included along with the citation, focusing on the relevance of this work to the argument being made or the overall research reported in the paper.

Previous work in citation recommendation has exploited this as a source of information for creating a representation of a scientific document. This is the approach taken by Ritchie (2009) and He et al. (2010). This is in most ways identical to the equivalent concept of *anchor text* used in web search, where the actual text that is shown for a hyperlink, or surrounds it, is mined as a description of the linked resource or web page.

An external document representation is built from the contexts of citations to this document found in other documents in the collection by simply extracting a span of text around those citations and concatenating all these spans into a single bag-of-words (BOW), which is then indexed in the same way as any other BOW. These BOWs from the incoming link contexts are also generated using the type of window methods that we presented in Section 2.9.3 for query extraction.

A visual representation of this approach can be found in Figure 2.10.2. In this diagram we find some essential ingredients to this approach:

- *Test document*: this is the document from which we extract the evaluation queries. As part of the test collection, this document is not part of the document collection and therefore not indexed as a candidate for recommendation.
- *Target document*: this is the document that is cited in a resolvable citation token that is found in the test document.
- *Citing documents*: these are other in-collection documents that are also part of the document collection and which cite the target document.
- *Incoming citation links*: these are citation tokens found in the document collection that resolve to the target document.

Figure 2.10.3 illustrates how the bag of words representing the documents is generated from the anchor text.

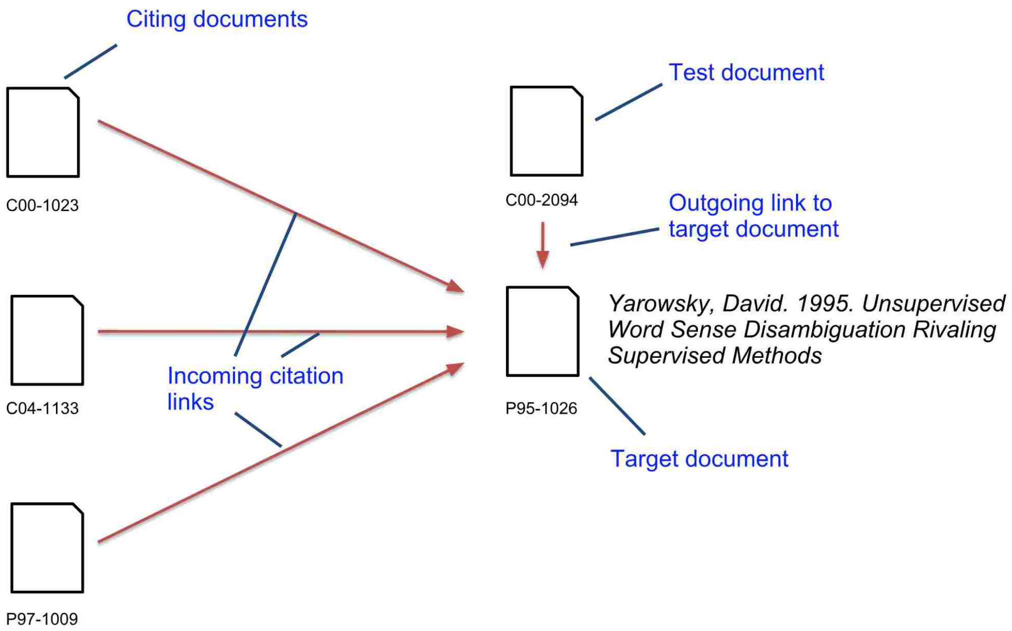
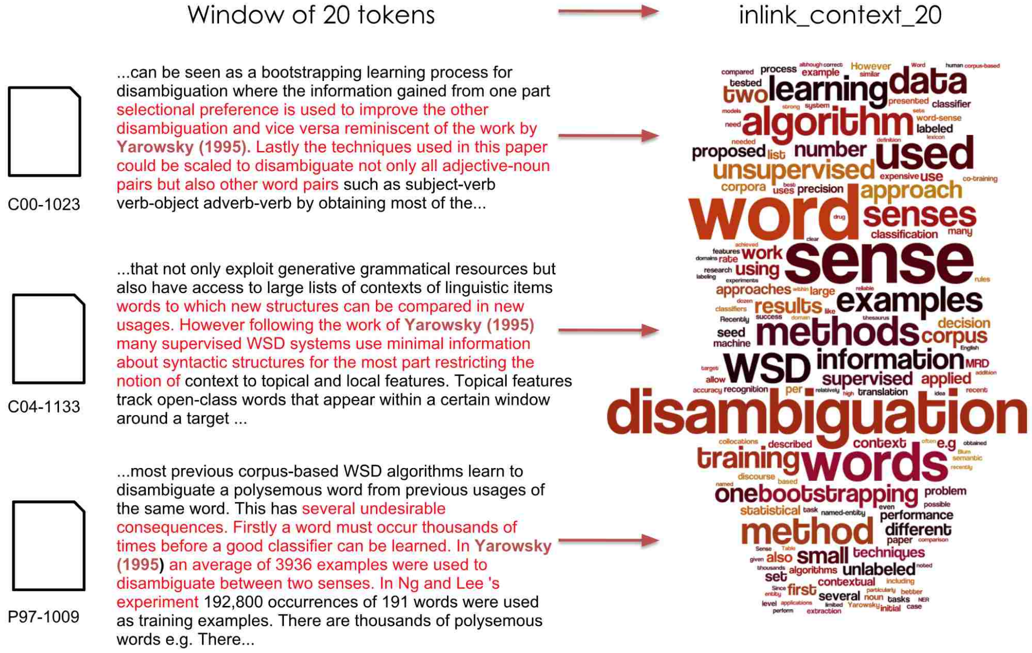

Citing documents
C00-1023
C04-1133
P97-1009
Incoming citation links
Test document
C00-2094
Outgoing link to target document
Yarowsky, David. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods
P95-1026
Target document

Figure 2.10.2: Intuition for external document representations.

2.10.1.3 Mixed document representations

We can combine internal and external document representations to create *mixed* document representations. While there are many ways they could be combined, in our work we simply concatenate internal and external bag-of-words representations of a document. We can see a visual example of this approach in Figure 2.10.4. We explore different parameters for building these representations in Chapter 3.



...can be seen as a bootstrapping learning process for disambiguation where the information gained from one part selectional preference is used to improve the other disambiguation and vice versa reminiscent of the work by Yarowsky (1995). Lastly the techniques used in this paper could be scaled to disambiguate not only all adjective-noun pairs but also other word pairs such as subject-verb verb-object adverb-verb by obtaining most of the...

...that not only exploit generative grammatical resources but also have access to large lists of contexts of linguistic items words to which new structures can be compared in new usages. However following the work of Yarowsky (1995) many supervised WSD systems use minimal information about syntactic structures for the most part restricting the notion of context to topical and local features. Topical features track open-class words that appear within a certain window around a target ...

...most previous corpus-based WSD algorithms learn to disambiguate a polysemous word from previous usages of the same word. This has several undesirable consequences. Firstly a word must occur thousands of times before a good classifier can be learned. In Yarowsky (1995) an average of 3936 examples were used to disambiguate between two senses. In Ng and Lee 's experiment 192,800 occurrences of 191 words were used as training examples. There are thousands of polysemous words e.g. There...

Figure 2.10.3: Illustration of a bag-of-words external document representation (word cloud): it is built by concatenating the bag-of-words contexts (red text) of outgoing citations to the target document.

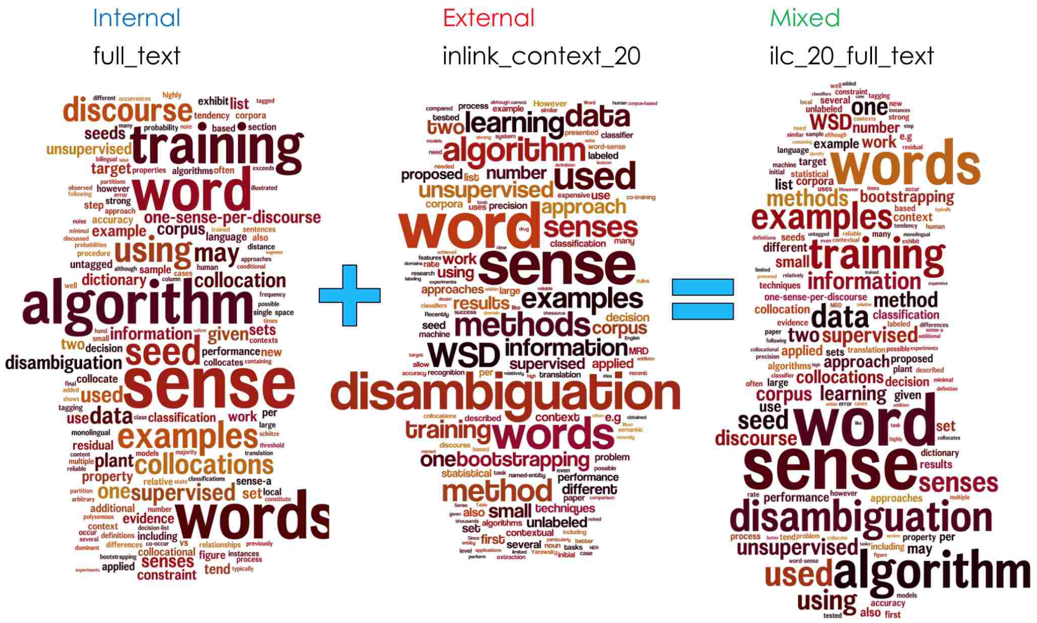


Figure 2.10.4: Illustration of a bag-of-words mixed document representation.

2.11 Corpora

One key issue encountered during the early stages of this project was securing access to suitable corpora of research papers. Scientific documents are now normally made available through digital means in born-digital formats, but these formats are designed for fidelity in print reproduction and for human consumption; not for being machine-readable and easily mined. By far the most common publication format of scientific documents is Adobe Portable Document Format (PDF), standardised as ISO 32000[9]. Unfortunately, extracting machine-readable text, figures, tables, citations and references from PDFs is a very challenging task with a significant error rate in reconstructing the original document structure (Alex and Burns, 2014). Even though version 6 of the PDF standard allows including some structural information in the final file, this is seldom present in PDFs in the wild (Déjean and Meunier, 2006).

In order that a case can be made for the generalisability of the approaches presented herein, these should be applied to more than one corpus in more than one domain. Several collections of documents were required that fit a number of requirements; each collection needs to:

1. Be in a machine-readable format. That is, each file needs to be in a structured text format like e.g. XML or JSON. Converting PDFs would introduce a significant error rate and add an unmanageable overhead to the project. From this format it should be possible to retrieve in text format: separate paragraphs, sections and section titles, inline citations and references for the document.

2. Have a degree of closure: for at least a few hundred documents, a significant percentage of their references should be resolvable to documents in the same collection.

3. Be focused to one domain. We would like to establish to what degree the findings about one discipline transfer to another, so the documents in the corpus must all belong to a reasonably clearly defined area of science, e.g. biochemistry, computational linguistics.

4. Be in a domain for which there exist Functional Annotation Schemes and automated systems for the tagging of the corpus. This will become important in this thesis and is presented in depth in Chapter 4.

[9] https://www.iso.org/standard/51502.html

5. Come with no legal restrictions to its processing and usage in a test CCR system.

6. Be of sufficient size. The exact measure of sufficiency is hard to define a priori, but generally we would like as big a collection as possible that fits all of the above criteria. A survey of research paper recommender systems concluded that minor variations in datasets lead to strong variations in the reported performance of the approaches (Beel et al., 2016b). Given the often relatively small corpora used for these experiments (as we can see in Table 2.10.1), we want to use as large a corpus as is practical given our experimental setup and available processing power.

Two such collections of documents stand out as good candidates for this task: the ACL Anthology Corpus and the Open Access Subset of PubMed Central.

2.11.1 ACL Anthology corpus

The ACL Anthology is a digital archive of conference and journal papers in natural language processing and computational linguistics since the 1970s (Bird et al., 2008), published by the Association of Computational Linguistics. The whole collection of documents is available to access and download in PDF format with practically no restrictions. A number of ongoing efforts exist to convert this collection to machine-readable formats[10].

An authoritative conversion of a large chunk of the ACL Anthology to a machine-readable XML-based format is the ACL Anthology Reference Corpus[11] (AAC) by Bird et al. (2008). The version employed for these experiments is meant to be a comprehensive conversion of the ACL Anthology from its origins up to and including the year 2011.

The AAC collection is based on automatically converted PDF files, many of them scans of two decades of ACL conference and journal issues. Automated methods for estimating of the accuracy of OCR conversion that correlate with human judgements have been proposed (Alex and Burns, 2014), but to our knowledge no such analysis of this corpus has been carried out yet. However, automated correction employing Neural Machine Translation has been attempted (Nastase and Hitschler, 2018). From manual inspection, the text remains full of the typical noise of OCR digitising, so we apply significant extra preprocessing, using a large number of bespoke heuristics and regular

[10] https://acl-arc.comp.nus.edu.sg/, https://web.eecs.umich.edu/~lahiri/acl_arc.html

[11] http://www.delph-in.net/aac/

expressions, targeted at removing inline XML, piecing together broken paragraphs and broken sentences, etc. For more details, please see Appendix A.

The initial experiments carried out as part of this research and reported in Duma and Klein (2014) used a conversion of the ACL Anthology by Ritchie et al. (2006a), where a complex pipeline including OCR processing and the matching to publication-specific templates were applied to generate XML representations of the PDF files. This corpus was significantly smaller (about 9000 documents) and the conversion quality from PDFs, again from manual examination, was much lower.

Metadata harvesting directly from PDF files is notoriously difficult, but fortunately the metadata for this collection can be obtained in a clean and manually-curated form from the ACL Anthology Network (ANN) (Radev et al., 2009b). This is key for matching an in-collection reference to the actual document in the collection and avoiding the manual annotation other studies needed to apply (Radev et al., 2009a).

2.11.2 PubMed Central Open Access Subset corpus

PubMed Central (PMC) is a repository of scientific articles based at the National Center for Biotechnology Information (NCBI). PMC was established with the aim to "provide a comprehensive electronic archive of the peer-reviewed literature relevant to the biological sciences" (Roberts, 2001). While copyrighted content *is* hosted in PMC[12], a significant part of the articles are published under an Open Access license, forming the PubMed Central Open Access Subset (which from now on we refer to as simply PMC), which allows carrying out the kind of indexing and mining that is necessary for the experiments reported herein.

Very importantly for our needs, these papers are provided in a hand-authored, schema-validated XML format (the Journal Article Tag Suite[13]) that preserves structural information such as paragraphs, sections, citations and references. This is perhaps the highest-quality machine-readable corpus of scientific papers that currently exists that includes not just the metadata (e.g. authors, publication date and venue, etc.) but also the fully structured textual document content.

[12] https://www.ncbi.nlm.nih.gov/pmc/about/intro/

[13] https://jats.nlm.nih.gov/

2.12 Corpora statistics

Table 2.12.1 shows the statistics for the two corpora as a whole. The values presented for the test set are the maximal available values – that is, corresponding to the total amount of documents in this set. In practice, a much smaller set of the documents is used to generate the evaluation queries from. The values presented in this table include:

Number of documents: total number of machine-readable documents in this subset of the corpus

Valid documents: these are documents that contain a minimum of two sentences. Documents containing a single sentence are in fact blank apart from the title.

Total citations: how many in-text citations are recoverable in all the documents

Resolvable citations: of these citations, how many can be matched with a reference in the paper's bibliography that can in turn be matched with the relevant paper in the corpus

In-collection references: how many of the references in the paper's bibliography are found in the corpus

Self-references: how many of these references link to papers where the citing paper and the cited paper share the same first author

Self-references ratio: what ratio this is of the total in-collection references

From Table 2.12.1we can see that most papers in the AAC corpus (98%) contain two or more sentences, and we deem these valid documents. This proportion is lower for PMC, at about 86%. We filter out the invalid documents for our test set from which we extract the queries.

	AAC Corpus	PMC Corpus
Total number of documents	22,428	1,029,271
Valid documents (> 1 sentence)	22,040	887,535
Total citations	217,732	35,003,708
Total resolvable citations	54,062	1,941,218
In-collection references	37,369	1,165,310
Self-references	1,749	38,426
Self-references ratio	4.68%	3.30%

Table 2.12.1: Statistics for the two corpora: ACL Anthology Corpus and PubMedCentral-Open Access Subset.

When examining the distribution over the values summarised in Table 2.12.1, we observe both similarities and differences between the two corpora. Yet a single number is a poor descriptor of a distribution. We can see histograms over several metrics relating to the documents in each collection in Figure 2.12.1 and Figure 2.12.2. Both collections have significant outliers, so for generating these plots, we have limited them to the valid papers in the corpus (those that contain more than one sentence) and filtered out outliers by selecting papers with a maximum of 400 sentences, 100 outgoing citations, 20 resolvable citations, 50 incoming citations (times cited), 20 in-collection references and 150 paragraphs.

- **Number of citations:** Even filtering out the blank papers, we still see that a large number of the papers in our corpora have no valid outgoing citation links that we could recover from the running text (just under 6k for AAC and over 250k for PMC). For AAC then we see a smooth distribution, where most papers contain between 1 and 10 valid citation tokens and this then smoothly decreases until fading into the long tail at around 30. PMC on the other hand has a greater diversity and no obvious trend, except the frequency continues until well around the 100 valid citations.

- **Number of in-collection references:** It is immediately clear that over 12k documents in AAC (more than half the corpus) have zero references to be found inside the collection. For PMC, the proportion is similar, at 500k out of just over a million papers.

- **Number of resolvable citations:** This is an important aspect of our collections: as we motivated before, we need a sufficient degree of inter-linking of our collection in order to properly evaluate the different methods. The distributions for this appear similar for the two corpora, with a sufficient number of documents containing resolvable citations.

- **Times cited:** Looking at the linking aspect from the incoming perspective, for AAC 7,286 papers are cited at least once inside the corpus, and this number is 308,571 for PMC.

- **Number of sentences, paragraphs and sections:** The median document has a similar number of sentences in both collections (135 for AAC and 139 for PMC), but from the plot we can see that while AAC approaches a normal distribution on this metric, PMC exhibits a bimodal distribution. This is due to many papers in PMC being abstract-only. This difference is then mirrored in the distribution over number of paragraphs, and, less clearly, sections.

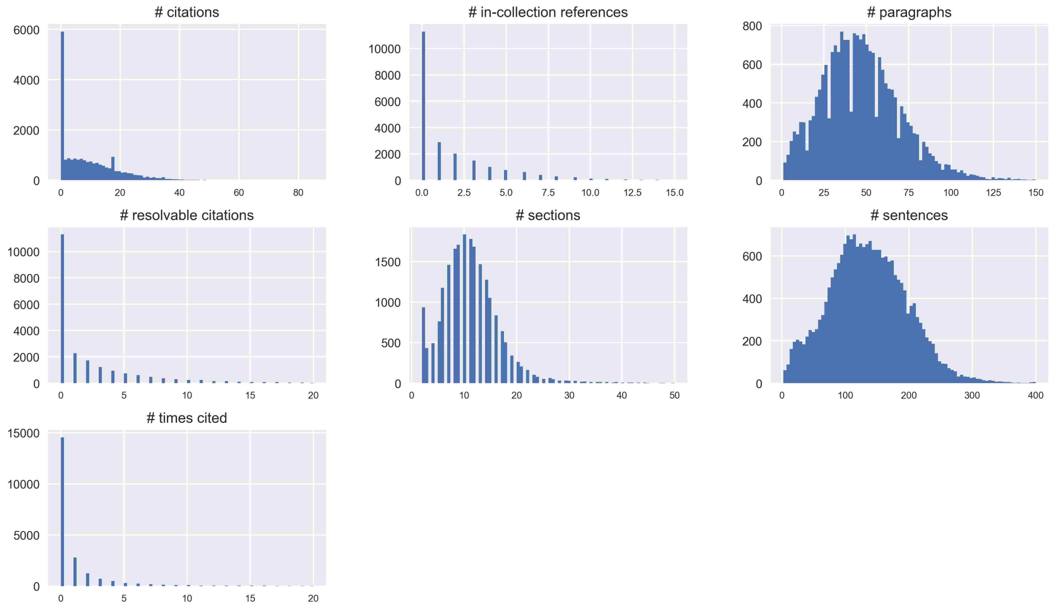


Figure 2.12.1: Histograms of AAC corpus statistics.

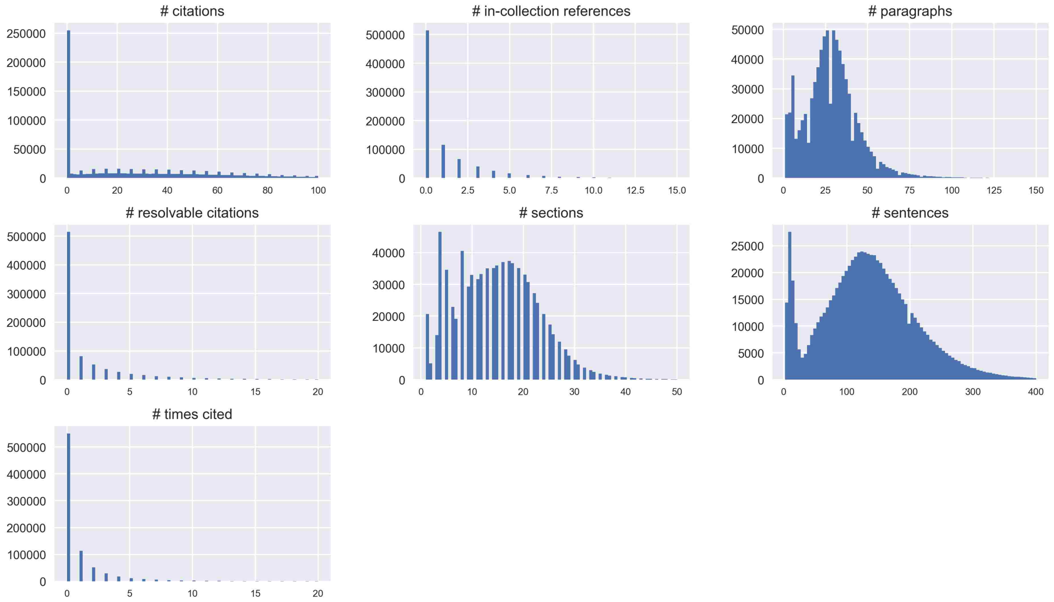


Figure 2.12.2: Histograms of PMC corpus statistics.

2.13 Corpus processing

In order to evaluate this task correctly, we need the structure of documents we are processing to be clearly annotated. Particularly, we need to be able to retrieve:

- in-text citation tokens
- structured data in the paper's references (at a minimum, title, authors, year)
- individual sentences and paragraphs

In the case of the PMC corpus, retrieving this structured representation requires very little effort as a majority of the annotation has been carried out and verified and is immediately retrievable from the XML file. The one notable exception is sentence splitting, which does not come pre-annotated and must be performed on the document.

For the AAC corpus, much of this annotation is unavailable or very noisy, so it must be performed on the corpus. This annotation does not require manual effort, but the automated ways of carrying it out are often non trivial due to OCR conversion errors, inconsistent XML annotations, or anecdotally variable reference styles, sometimes used inconsistently inside a same document. More on this in Appendix A.

Some of the automated tasks that must be carried out include:

- Identify and annotate citation tokens in the body of the document. This is mostly carried out by applying several regular expressions.
- Parse text from the References/Bibliography section, trying to recover the names of the authors, the title of the article and the year of publication. This is important both to match the reference to its in-text citations and to match it to the correct document in the corpus. The AAC corpus requires a larger measure of processing, as the PMC corpus is perfectly structured.
- Match the in-text citation with the reference in the bibliography section.
- Match the reference with the right paper in the corpus.
- Split sentences in the text.
- Parse the XML schema of the corpus.
- Extract position-relevant text from the document, such as:
    - Individual sentences and the paragraphs they belong to.

- The sentence containing a citation token and sentences preceding or following it, within the same paragraph or outside.
- Sections and section headers

All of these are important to the experimental approach presented here, which benefits from clean and clearly structured data. Beyond extracting a citation's context for the purpose of generating a query and selecting different spans of text of a document to index, being able to retrieve individual sentences, paragraphs and sections is key for their use as features for the automatic annotation of sentence function, as explained in Chapter 4.

2.14 Conclusion

We have presented our task: Contextual Citation Recommendation (CCR) and motivated framing it as an *information retrieval* (IR) problem, together with a short introduction to the field of IR and the general formulation of its approaches. The basis of evaluation is to attempt to recover the original document cited in a given context in published paper, using information retrieval methods to retrieve it from the indexed document collection. We have also defined important concepts used throughout this thesis and provided a short introduction to the information retrieval methods and frameworks we employ. In the next chapter, we explore some of the most relevant previous work on CCR, comparing it with our own approaches, and explore the corpora we will be evaluating against. We have also motivated and defined our evaluation method, which we call *citation resolution*.

There is abundant previous literature belonging to the field of recommender systems. Of the several classes of methodologies applied, our task aligns most closely with content-based filtering, yet is not subsumed by the approaches in this field.

Several previous approaches to the task of Contextual Citation Recommendation exist that frame it as an Information Retrieval task. In order to organise them and guide our own research, we propose three research *aspects*: *document representation*, which deals with what spans of text are extracted from which documents in order to then be indexed for retrieval; *query extraction*, which is the methodology for extracting the context around a citation placeholder and generating an IR query from this text; and *similarity metric*, which is the algorithmic way of measuring the similarity between the query and each document in the collection.

For document representation, we have defined the three types of methods we have identified in previous literature: *internal methods* (indexing just the text contained in the document itself), *external methods* (indexing the anchor text of a citation to the document) and *mixed methods* (combining the previous two).

We have selected two corpora of machine readable, open access publications in two different domains. One is the ACL Anthology Corpus (AAC), a conversion of publications from conference and journal papers in computational linguistics of roughly 22,000 documents to a custom XML format. The other is the PubMed Central Open Access Subset (PMC): a collection of over one milllion papers, most of which contain the full document text. This corpus is the highest quality such resource we have identified, as it is manually curated, converted and schema-validated to a well documented, open standard XML format.

In the next chapter, we explore two of the aspects we described: *document representation* and *query extraction* using standard symmetric methods, exploring each corpus individually.

Chapter 3

A baseline Information Retrieval approach

“The journey of a thousand miles begins with a single step."

Lao Tzu

3.1 Introduction

This chapter presents a baseline approach to the task of Contextual Citation Recommendation, based on the standard formulation of it as an Information Retrieval scenario. As hinted at in Chapter 2, it is not feasible to directly compare with the results of other similar studies given the variation in corpora and the lack of standard annotated test collections for this task. The lack of a common evaluation framework and therefore common baselines is a widely recognised problem in the research landscape of academic recommender systems (Beel et al., 2016a). For example, while plenty of published research uses the CiteSeer collection for offline evaluation, this dataset is not fixed: different studies collected CiteSeer data at different times and pruned the datasets differently.

As pointed out in Section 1.4.3, our aim is to test new approaches to CCR that involve deeper analysis of scientific documents. Inevitably, such approaches benefit from the documents being available in a structured format such as XML. This contrasts with previous IR-based approaches (e.g. He et al. (2010), He et al. (2012), Huang et al. (2012)) which have treated the documents as unstructured running text, often mined from converted PDFs. Consequently, this is another factor which makes it infeasible to draw direct comparisons with previous work on CCR.

However, we can apply previously used, de facto standard techniques for query extraction and document representation using our fixed corpora and retrieval backend. We explore in depth a wider range of parameters to the context extraction and document representation than has been previously published, which will establish baselines for all further experiments presented here.

3.2 Methodology

3.2.1 Corpus split

First, the corpus is split into a *document collection,* which is indexed for retrieval, and a *test set,* which we treat as the set of draft papers that we want recommendations for[1]. The main division is by *year*: we want to clearly and unambiguously separate:

1. the documents that we consider the draft documents for which we need to recommend relevant citations, from which we will be extracting the queries

2. the documents that we want to recommend, including the documents from which we build external document representations, as detailed below.

At the same time, we need documents with as many collection-internal references as possible to generate these queries. Therefore we select the test set from the most recent documents in both collections and use the remainder as our document collection.

For the ACL Anthology Corpus (AAC), we select documents published in 2011 and after for our test set and indexed the remaining documents as the document collection. For PubMed Central (PMC), it is 2014 and onwards. Then for each corpus, documents are selected for query extraction, ordering them by number of collection-internal references, in descending order. From these, a query is extracted from each resolvable citation, using each query extraction method, which is documented in Section 3.2.3 below.

[1] This approach was originally presented in Duma and Klein (2014), and while what we present here is conceptually very similar, there are several differences. The main one is that the evaluation now considers all documents in the document collection for retrieval as opposed to only those cited within a single paper. Further to this, experiments are expanded and now cover two larger corpora, none of them used in the original paper.

	AAC Corpus	Document Collection	Test Set
Number of documents	22428	20395	1653
Valid documents (> 1 sentence)	22040	20387	1653
Total citations	217732	190460	27272
Resolvable citations	54062	45696	8366
In-collection references	37369	31531	5838
Self-references	1749	1535	214
Self-references ratio	4.68%	4.87%	3.67%

Table 3.2.1: AAC corpus split statistics for experiments.

	PMC Corpus	Document Collection	Test Set
Number of documents	1029271	706814	268039
Valid documents (> 1 sentence)	887535	634788	252747
Total citations	35003708	25766440	9237268
Resolvable citations	1941218	1229883	711335
In-collection references	1165310	729077	436233
Self-references	38426	25272	13154
Self-references ratio	3.30%	3.47%	3.02%

Table 3.2.2: PMC corpus split statistics for experiments.

3.2.2 Indexing

The document collection is indexed using Elasticsearch. While Apache Lucene, the underlying retrieval engine, is a field-based retrieval engine as detailed in Chapter 2, we index all text into a single field, i.e. a single bag-of-words for a Vector-Space Model (VSM) representation. No distinction is made in the index between text coming from different semantically meaningful parts of a paper, e.g. title, abstract, section headers, etc.

The two document collections (AAC and PMC) are indexed independently and into separate indices: it is impossible for a paper from one collection to appear in the results of a query generated from a paper in the other.

Three different approaches to generating a document's VSM representation are tested here: *internal representations*, which are based on the contents of the document, *external representations*, which are built using a document's incoming link citation contexts (following Ritchie (2009) and He et al. (2010)) and *mixed representations*, which are an attempt to combine the two.

Internal representations: these are generated using two different methods: *title plus abstract* and *full text*. Title plus abstract means only the text contained in the title and the abstract of the paper are used to generate the BOW. Full text is unsurprisingly the full plain text of the document, including section headings, footnotes, etc. but excluding references. Previous work has sometimes employed *passages* for generating document representations (e.g. Duma and Klein (2014)). These consist in splitting the document into half-overlapping passages of a fixed length of k words and choosing for each document the passage with the highest similarity score with the query. While in naïve custom implementations of information retrieval they have been found to sometimes outperform full text as a document representation, we have found in our experiments that when using modern similarity metrics and weighting schemes which normalise tf-idf scores over document length this advantage disappears, so we exclude them from our evaluation.

External representations (*inlink_context*): these are based on extracting the context around citation tokens to the document from other documents in the collection, excluding the set of test papers. This is the same as using the *anchor text* of a hyperlink to improve results in web-based IR (see Davison (2000) for extensive analysis). This context is extracted in the same way as the query: as a window, i.e. a list of w tokens surrounding the citation left and right. We present our results using symmetrical windows of $w = \{30, 40, 50, 100\}$ before and after, and windows of sentences using different combinations of "up" and "down" parameters, e.g. *2up_2down* or *1only* (only the citing sentence).

Mixed representations: these are built by simply concatenating the internal and external bags-of-words that represent the documents, from which we then build the VSM representation. For this, we combine different window sizes for the *inlink_context* with *full_text* and *title_abstract*.

Once generated, these BOWs are each indexed in an index by type (i.e. all BOWs inlink_context_10_10 will be added to the same index, separate from all other types).

3.2.3 Query extraction

We want to use the existing text surrounding a citation in a document to generate a query that will retrieve the paper or papers that were originally cited (the *target documents*). For this, we substitute all citations in the text with *citation token placeholders* and extract the citation context for each. This context is a list of word tokens and we make this list our query in IR terms. A single extra processing step is added here, where stopwords from a very short list[2] are removed in order to speed up the running of the queries, as well as all punctuation. We observe this increases the average scores: we will explore motivated stopword selection and removal in depth in Chapter 6. Other than this, the terms are extracted as they come, including their term frequency in the context. We use two different types of windows for extraction: windows of tokens and windows of sentences.

Window of tokens (methods named *window*): we take a *window* of up to *w1* words left and *w2* words right around the placeholder. For example, *window20_10* means extracting a context window of 20 tokens before the citation token and 10 tokens after.

Window of sentences (methods named *sentence*): we select the sentence containing the citation and others around it. For example, *sentence1up_1down* means extracting the sentence containing the citation, one before it, and one after it, for a total of 3 sentences. On the other hand, *sentence_paragraph* means extracting the whole paragraph where the citation occurs.

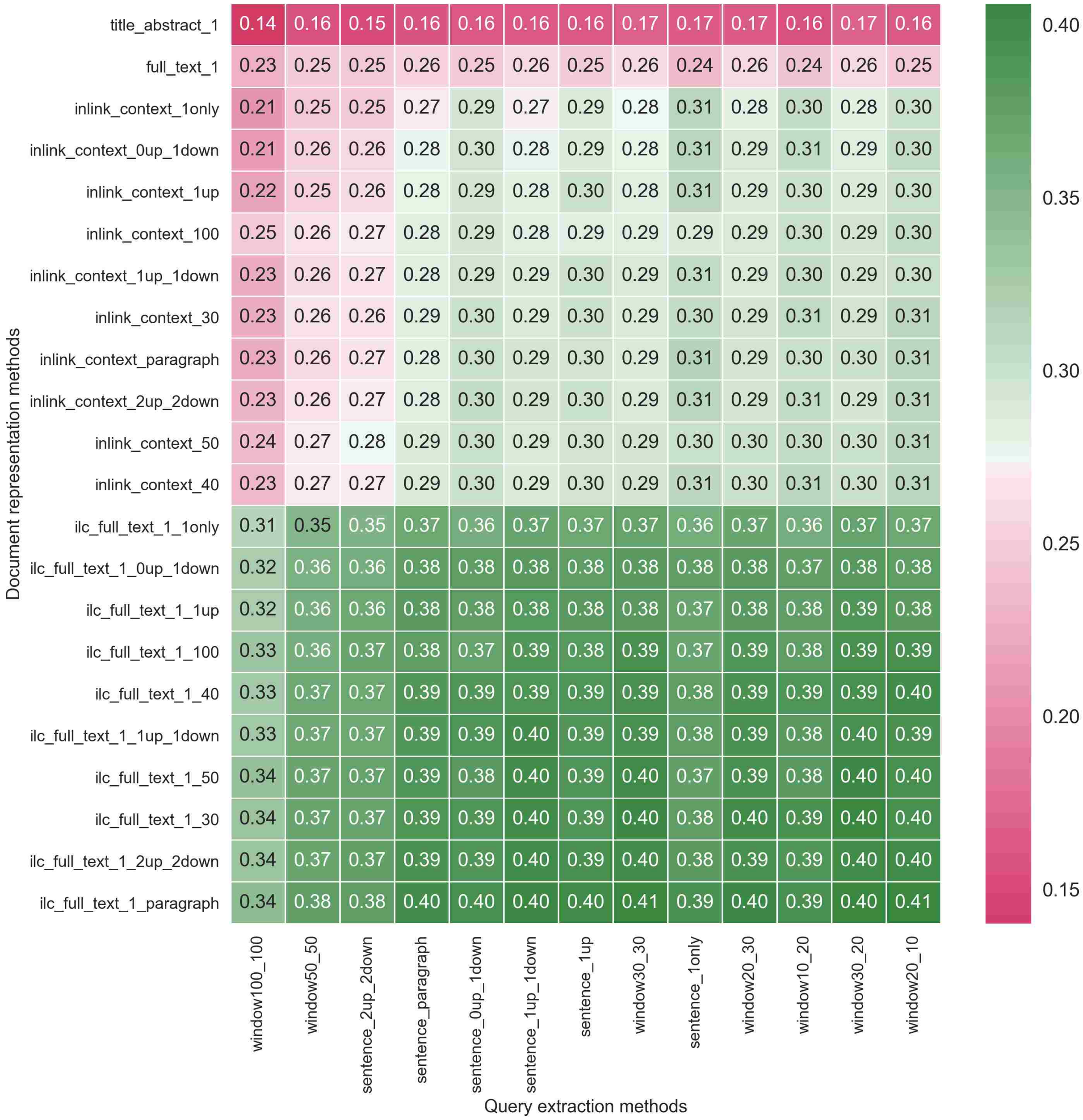
3.2.4 Similarity function

Each query is run with default parameters using the Lucene DefaultSimilarity, where each extracted term is added to the query as an independent term joined with an OR operator. This has the effect of simply adding up the scores of the query terms that match the document.

3.3 Results: AAC

Table 3.3.1 presents a selection of the most relevant results, with the best result and document representation method of each type highlighted. We present results for the most relevant parameter values, producing the highest scores of all those tested.

[2] *{a, an, and, by, from, not, of, or, the, to, with}. We manually selected the top 10 stopwords from a number of stopword lists we obtained from open source projects. We then added "an" to this list as a version of "a".*


0.40
0.35
0.30
0.25
0.20
0.15
Document representation methods
Query extraction methods

	window100_100	window50_50	sentence_2up_2down	sentence_paragraph	sentence_0up_1down	sentence_1up_1down	sentence_1up	window30_30	sentence_1only	window20_30	window10_20	window30_20	window20_10
title_abstract_1	0.14	0.16	0.15	0.16	0.16	0.16	0.16	0.17	0.17	0.17	0.16	0.17	0.16
full_text_1	0.23	0.25	0.25	0.26	0.25	0.26	0.25	0.26	0.24	0.26	0.24	0.26	0.25
inlink_context_1only	0.21	0.25	0.25	0.27	0.29	0.27	0.29	0.28	0.31	0.28	0.30	0.28	0.30
inlink_context_0up_1down	0.21	0.26	0.26	0.28	0.30	0.28	0.29	0.28	0.31	0.29	0.31	0.29	0.30
inlink_context_1up	0.22	0.25	0.26	0.28	0.29	0.28	0.30	0.28	0.31	0.29	0.30	0.29	0.30
inlink_context_100	0.25	0.26	0.27	0.28	0.29	0.28	0.29	0.29	0.29	0.29	0.30	0.29	0.30
inlink_context_1up_1down	0.23	0.26	0.27	0.28	0.29	0.29	0.30	0.29	0.31	0.29	0.30	0.29	0.30
inlink_context_30	0.23	0.26	0.26	0.29	0.30	0.29	0.30	0.29	0.30	0.29	0.31	0.29	0.31
inlink_context_paragraph	0.23	0.26	0.27	0.28	0.30	0.29	0.30	0.29	0.31	0.29	0.30	0.30	0.31
inlink_context_2up_2down	0.23	0.26	0.27	0.28	0.30	0.29	0.30	0.29	0.31	0.29	0.31	0.29	0.31
inlink_context_50	0.24	0.27	0.28	0.29	0.30	0.29	0.30	0.29	0.30	0.30	0.30	0.30	0.31
inlink_context_40	0.23	0.27	0.27	0.29	0.30	0.29	0.30	0.29	0.31	0.30	0.31	0.30	0.31
ilc_full_text_1_1only	0.31	0.35	0.35	0.37	0.36	0.37	0.37	0.37	0.36	0.37	0.36	0.37	0.37
ilc_full_text_1_0up_1down	0.32	0.36	0.36	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.37	0.38	0.38
ilc_full_text_1_1up	0.32	0.36	0.36	0.38	0.38	0.38	0.38	0.38	0.37	0.38	0.38	0.39	0.38
ilc_full_text_1_100	0.33	0.36	0.37	0.38	0.37	0.39	0.38	0.39	0.37	0.39	0.38	0.39	0.39
ilc_full_text_1_40	0.33	0.37	0.37	0.39	0.39	0.39	0.39	0.39	0.38	0.39	0.39	0.39	0.40
ilc_full_text_1_1up_1down	0.33	0.37	0.37	0.39	0.39	0.40	0.39	0.39	0.38	0.39	0.38	0.40	0.39
ilc_full_text_1_50	0.34	0.37	0.37	0.39	0.38	0.40	0.39	0.40	0.37	0.39	0.38	0.40	0.40
ilc_full_text_1_30	0.34	0.37	0.37	0.39	0.39	0.40	0.39	0.40	0.38	0.40	0.39	0.40	0.40
ilc_full_text_1_2up_2down	0.34	0.37	0.37	0.39	0.39	0.40	0.39	0.40	0.38	0.39	0.39	0.40	0.40
ilc_full_text_1_paragraph	0.34	0.38	0.38	0.40	0.40	0.40	0.40	0.41	0.39	0.40	0.39	0.40	0.41

Table 3.3.1: Heatmap of average NDCG score for each document representation method (rows) and query extraction method, including context window size (columns) in the AAC experiments, where red represents the lowest scores and green the highest scores. The methods starting with *ilc* are those of type *inlink_context*.

When we look at the maximum average performance of each query extraction method (Figure 3.3.1), we see that the query with the largest amount of terms is *window100_100* and it is the worst performing one, and generally, the fewer terms extracted, the higher performing the query. Of the sentence-based methods, *1up_1down* performs the best, which shows that in the AAC corpus there is often important information in the sentences both before and after a citation token.

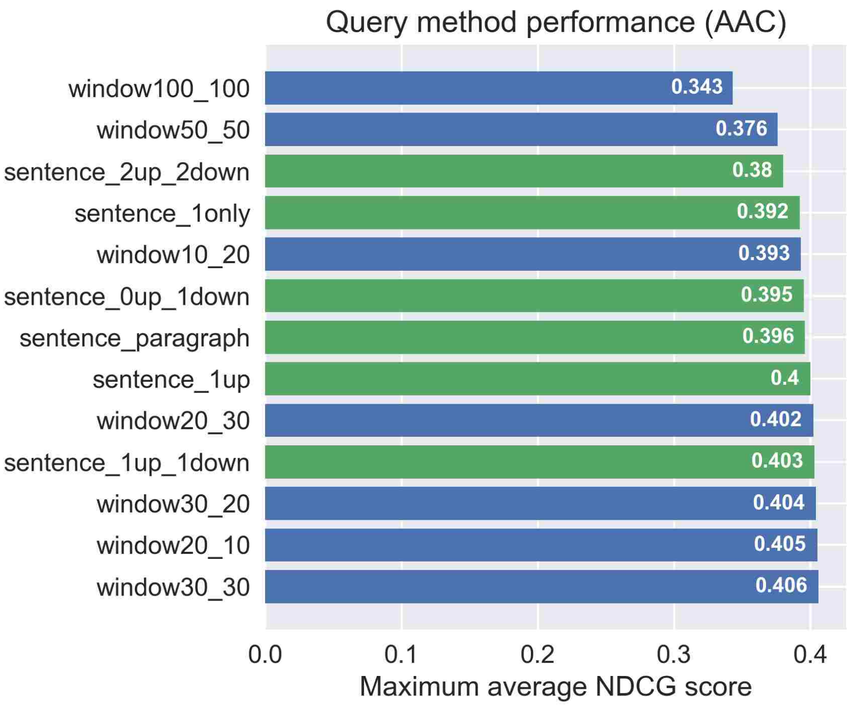


Figure 3.3.1: Maximum average NDCG of the query extraction methods for AAC.

However, it is a closer look at the document representation methods (Figure 3.3.2) that gives us the most interesting insights. When grouped by type of method (internal in red, external in green, mixed in blue) we can clearly see that:

- Title + abstract is the worst performing method, which is unsurprising as the amount of information is necessarily low. Full text very clearly outperforms it.
- All external methods clearly outperform internal ones by a large margin and with not much difference between them. It has been suggested before that this is because the descriptions of the cited papers in the contexts of incoming link citations capture the essence or key relevance of the paper. There may also be an effect of these descriptions originating in a seminal paper and being then propagated through the literature. We have tried to limit the effect of authors reusing their work by filtering out self-citations, but examining this effect in more depth remains an open research question, to be tackled outside of this thesis.
- Mixed methods, which combine internal and external, achieve by far the best results. This is perhaps the most important finding. Again here we see there may

be a sweet spot for how much of the text around an incoming citation to extract in order to concatenate it to the contents of the target document.

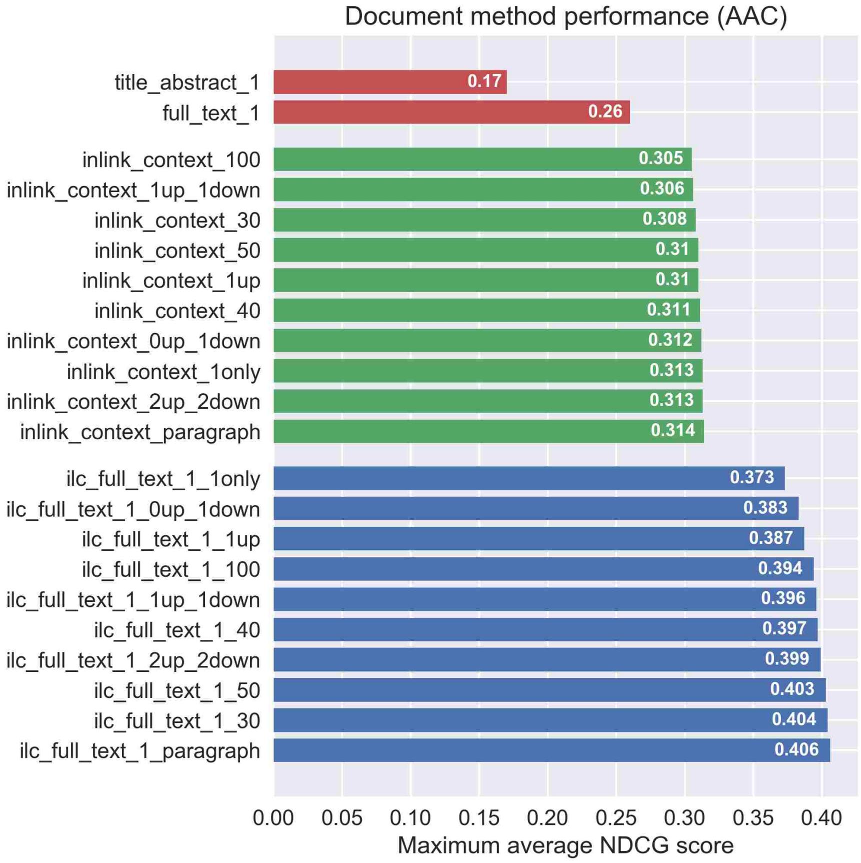
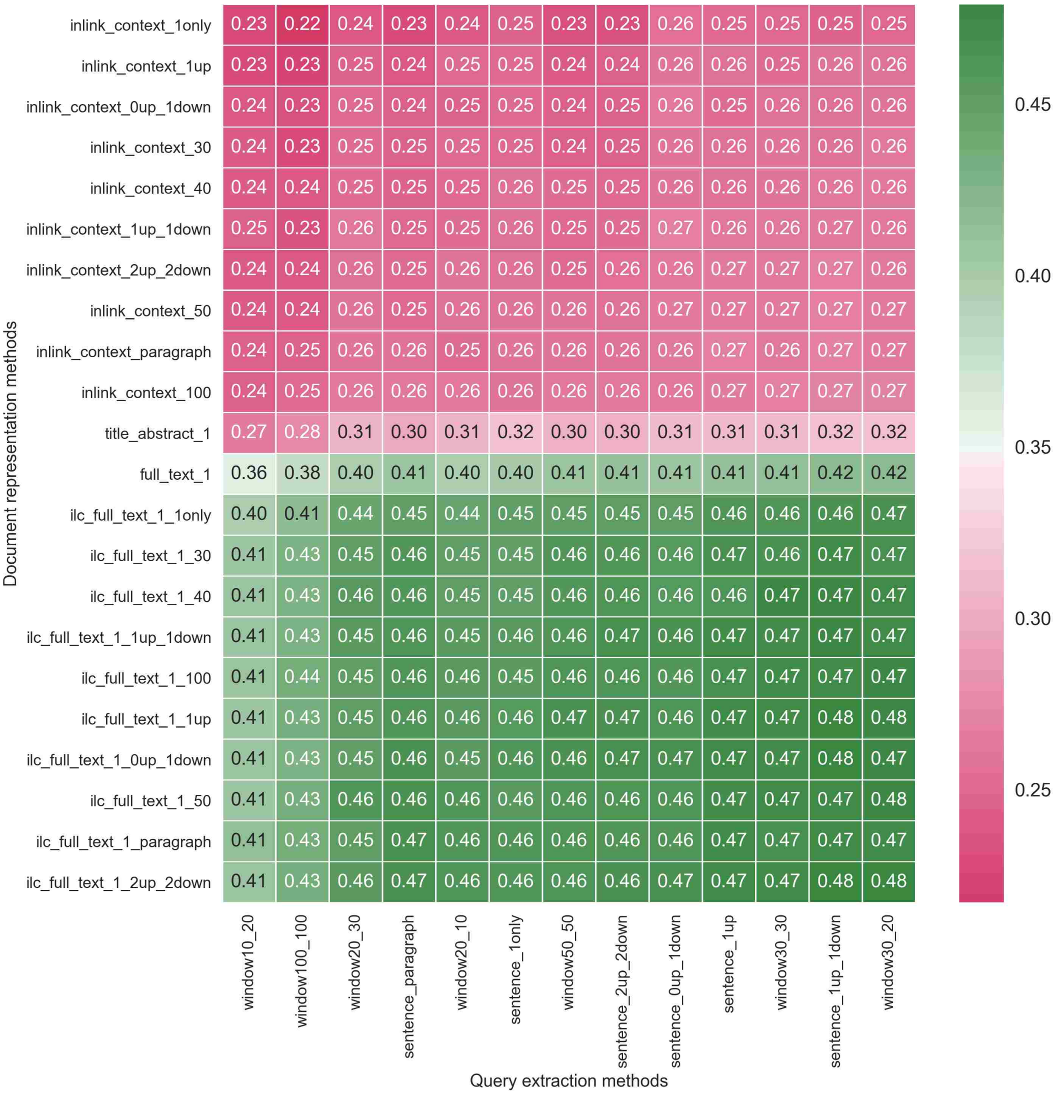


Figure 3.3.2: Maximum average NDCG of the document representation methods for AAC.

3.4 Results: PMC

After running these experiments with the exact same parameters on the PMC corpus, we notice some differences. Figure 3.4.1 shows the full results.

We can see similar results across query method performance for PMC as we did for AAC, with *window30_20* (30 left, 20 right) being the best performing method (where for ACL it was the second best), and *window100_100* (the largest one) being the second worst performing one (similar to AAC, where it is the second worst). In the case of PMC, the stark contrast is that *window10_20* is the absolutely worst performing one, suggesting that in this corpus, the most relevant keywords occur before the citation token rather than after it.



	window10_20	window100_100	window20_30	sentence_paragraph	window20_10	sentence_1only	window50_50	sentence_2up_2down	sentence_0up_1down	sentence_1up	window30_30	sentence_1up_1down	window30_20
inlink_context_1only	0.23	0.22	0.24	0.23	0.24	0.25	0.23	0.23	0.26	0.25	0.25	0.25	0.25
inlink_context_1up	0.23	0.23	0.25	0.24	0.25	0.25	0.24	0.24	0.26	0.26	0.25	0.26	0.26
inlink_context_0up_1down	0.24	0.23	0.25	0.24	0.25	0.25	0.24	0.25	0.26	0.25	0.26	0.26	0.26
inlink_context_30	0.24	0.23	0.25	0.25	0.25	0.25	0.24	0.25	0.26	0.26	0.26	0.26	0.26
inlink_context_40	0.24	0.24	0.25	0.25	0.25	0.26	0.25	0.25	0.26	0.26	0.26	0.26	0.26
inlink_context_1up_1down	0.25	0.23	0.26	0.25	0.25	0.26	0.25	0.25	0.27	0.26	0.26	0.27	0.26
inlink_context_2up_2down	0.24	0.24	0.26	0.25	0.26	0.26	0.25	0.26	0.26	0.27	0.27	0.27	0.26
inlink_context_50	0.24	0.24	0.26	0.25	0.26	0.26	0.26	0.26	0.27	0.27	0.27	0.27	0.27
inlink_context_paragraph	0.24	0.25	0.26	0.26	0.25	0.26	0.26	0.26	0.26	0.27	0.26	0.27	0.27
inlink_context_100	0.24	0.25	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.27	0.27	0.27	0.27
title_abstract_1	0.27	0.28	0.31	0.30	0.31	0.32	0.30	0.30	0.31	0.31	0.31	0.32	0.32
full_text_1	0.36	0.38	0.40	0.41	0.40	0.40	0.41	0.41	0.41	0.41	0.41	0.42	0.42
ilc_full_text_1_1only	0.40	0.41	0.44	0.45	0.44	0.45	0.45	0.45	0.45	0.46	0.46	0.46	0.47
ilc_full_text_1_30	0.41	0.43	0.45	0.46	0.45	0.45	0.46	0.46	0.46	0.47	0.46	0.47	0.47
ilc_full_text_1_40	0.41	0.43	0.46	0.46	0.45	0.45	0.46	0.46	0.46	0.46	0.47	0.47	0.47
ilc_full_text_1_1up_1down	0.41	0.43	0.45	0.46	0.45	0.46	0.46	0.47	0.46	0.47	0.47	0.47	0.47
ilc_full_text_1_100	0.41	0.44	0.45	0.46	0.46	0.45	0.46	0.46	0.46	0.47	0.47	0.47	0.47
ilc_full_text_1_1up	0.41	0.43	0.45	0.46	0.46	0.46	0.47	0.47	0.46	0.47	0.47	0.48	0.48
ilc_full_text_1_0up_1down	0.41	0.43	0.45	0.46	0.45	0.46	0.46	0.47	0.47	0.47	0.47	0.48	0.47
ilc_full_text_1_50	0.41	0.43	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.47	0.47	0.47	0.48
ilc_full_text_1_paragraph	0.41	0.43	0.45	0.47	0.46	0.46	0.46	0.46	0.46	0.47	0.47	0.47	0.47
ilc_full_text_1_2up_2down	0.41	0.43	0.46	0.47	0.46	0.46	0.46	0.46	0.47	0.47	0.47	0.48	0.48

Table 3.4.1: Heatmap of maximum average NDCG score for each document representation method (rows) and query extraction method, including context window size (columns) in the PMC experiments.

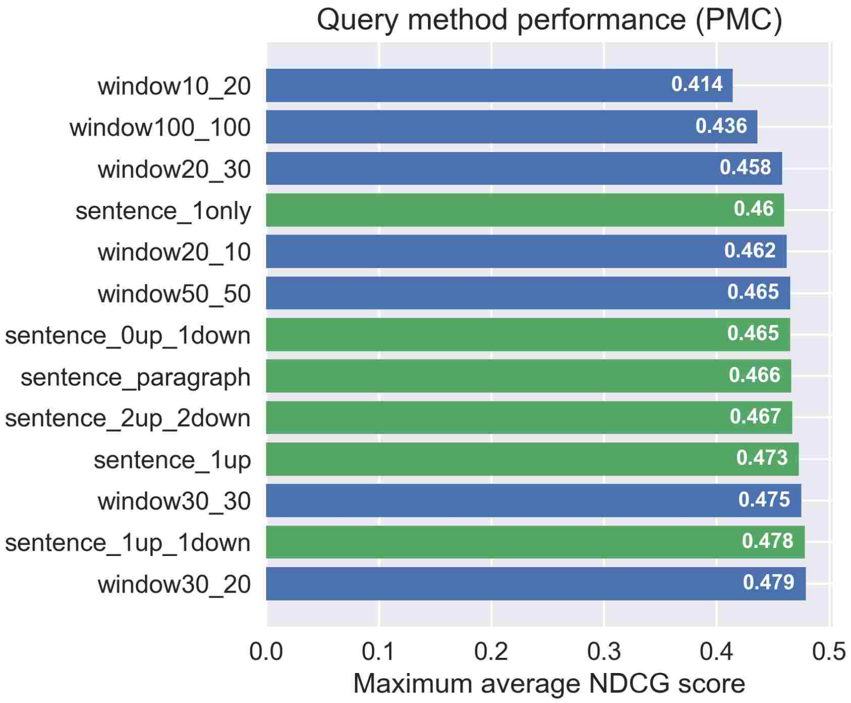


Figure 3.4.1: Maximum average NDCG (across document representation methods) of the query extraction methods for PMC.

The most notable differences with the AAC results can be seen in the document representation methods:

- Once again, *full_text* vastly outperforms *title_abstract*, as is entirely expected. What is noteworthy is how well this performs, approaching the performance of mixed methods.

- However, perhaps the most notable difference here is how much worse all external methods perform on this corpus. Just indexing the title and abstract of a document beats any external method. In AAC we see external methods outperforming internal ones, perhaps partly because the terms used as in the context of a citation are not so prominent in the paper itself, but they are consistently used in contexts of citations to it. However, these results suggest that in the biomedical domain, the incoming link contexts of a document would be less consistent in vocabulary use. At the same time, it seems likely that different parts of an article are cited. We will give these citing patterns more attention in Chapter 5.

- And yet we find that mixed methods are still the best performing, so the incoming link contexts clearly also contain important information that is not found in the documents themselves.

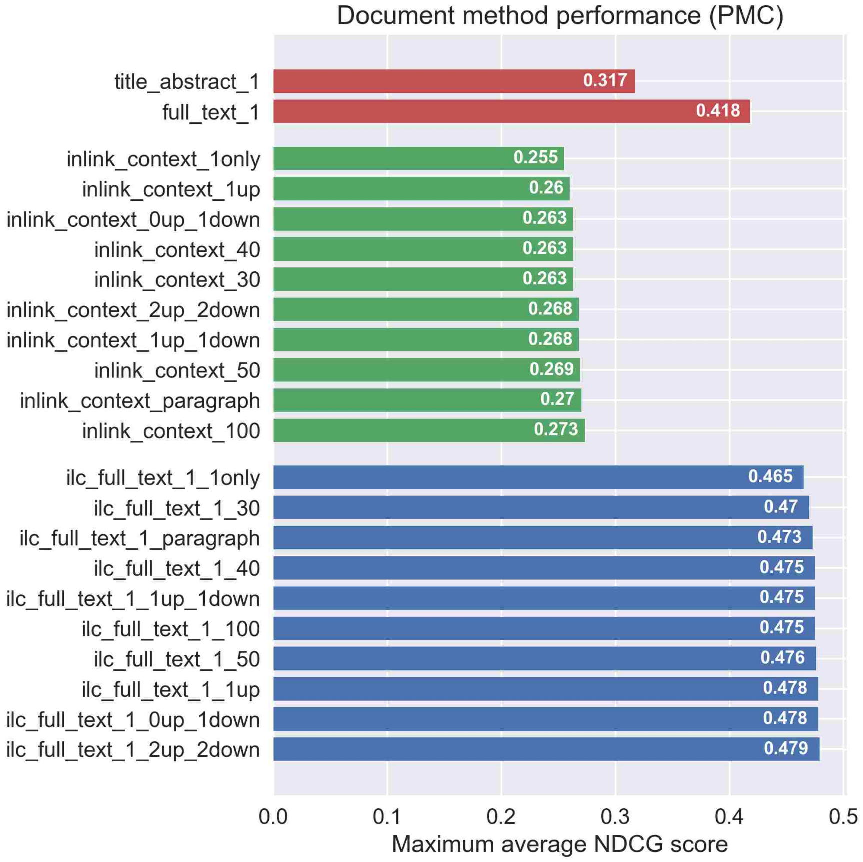


Figure 3.4.2: Maximum average NDCG (across query methods) of the document representation methods for PMC.

3.5 Statistical significance of results

For each corpus, we select the best performing combination of *document representation* method and *query extraction* method for each type of document representation method (*internal*, *external* and *mixed*). In order to establish the statistical significance of these results, a one-way ANOVA was conducted to compare the effect of the type of document representation method on the NDCG scores of the 1000 evaluation queries. Table 3.5.1 presents the best performing combination of methods for each corpus.

For both corpora, we observe that the differences in scores between the top performing combination in each of the three categories of methods (internal, external and mixed) are statistically significant ($p < .001$). To establish the significant differences between each pairing of methods, we apply Tukey's post-hoc test, which also shows clear statistical significance ($p < .001$) between all pairings in both corpora. Table 3.5.2 presents the results for AAC, and those for PMC are shown in Table 3.5.3.

3.6 Discussion of results

From separate experiments with both corpora, we conclude that:

- **Indexing the full text of a document leads to better results than only the title and abstract.** This result is intuitive and holds true for both domains: there is information in the paper that is not captured by the abstract, and citation contexts are likely to refer to aspects of a paper that are more specific or detailed than those that appear in the abstract.

- **A mixture of internal and external methods beats both individually.** Mixed methods consistently and significantly ($p < .001$) improve over both using the document's text and that of incoming link contexts in isolation. This confirms results from previous work on anchor text (e.g. Ritchie (2009)) and we find it holds true for both corpora and therefore both domains.

- **External methods contribute very differently to retrieval on different corpora.** This is perhaps the main novel finding of these experiments. While for AAC we observe that external methods unambiguously beat internal methods, for PMC the text of the document leads to much higher accuracy than the inlink context, both when taking the full document text and just the title and abstract. Inlink context text is useful for both domains, but not to the same degree, which is a strong difference that has not been reported before.

- **External representations benefit from a larger context.** It is not clear what the absolute best setting is, as we can see that for ACL it is *paragraph* and for PMC it is *2up_2down* that obtain the best results, but broadly our results agree with previous results reported by Ritchie et al. (2008) in that, "the longer the citation context, the greater the retrieval effectiveness". In their study, they also reported that symmetric windows of 3 sentences (*1up_1down*) outperformend single sentences (*1only*), and windows of 100 tokens oupterformed those of 50 tokens.

- **A broad query generally performs worse than a narrow one.** That is, adding more terms to the query boosts less relevant papers in the collection and increases

	AAC		PMC	
	Document	**Query**	**Document**	**Query**
Internal	full_text_1	window30_30	full_text_1	window30_20
External	inlink_context_paragraph	sentence_1only	inlink_context_100	sentence_1up_1down
Mixed	ilc_full_text_1_paragraph	window30_30	ilc_full_text_1_2up_2down	window30_20

Table 3.5.1: Best performing document representation methods for each corpus, together with the query extraction methods that yield the highest scores for each of the document representation methods.

ANOVA - ndcg_score (AAC)

Cases	Sum of Squares	df	Mean Square	F	p
method	10.94	2	5.469	52.25	< .001
Residual	313.68	2997	0.105		

Post hoc comparisons

		Mean Difference	SE	t	p_{tukey}
Internal	External	−0.054	0.014	−3.706	< .001
	Mixed	−0.146	0.014	−10.104	< .001
External	Mixed	−0.093	0.014	−6.398	< .001

Table 3.5.2: Results of ANOVA and post-hoc tests for AAC.

ANOVA - ndcg_score (PMC)

Cases	Sum of Squares	df	Mean Square	F	p
method	22.70	2	11.348	82.13	< .001
Residual	414.14	2997	0.138		

Post hoc comparisons

		Mean Difference	SE	t	p_{tukey}
Internal	External	−0.146	0.017	8.775	< .001
	Mixed	−0.062	0.017	−3.701	< .001
External	Mixed	−0.207	0.017	−12.477	< .001

Table 3.5.3: Results of ANOVA and post-hoc tests for PMC.

the rank of the target document, making it appear further down in the list of results (lower rank = higher score). As far a query extraction is concerned, more is not better.

3.7 Conclusion

In this chapter we have explored different ways of building a document representation and of extracting the context around a citation token from which to generate the evaluation queries. Our aim has been to establish a baseline by implementing and systematically testing different combinations of standard approaches to document representation and query extraction.

We have uncovered notable differences between our two corpora, and so between the two domains. One of the most interesting is the effect of anchor text: while in the domain of computational linguistics (AAC corpus) external methods far outperform internal ones, the exact opposite is true of biomedical science (PMC). This is to our knowledge a novel finding that had never been investigated for Contextual Citation Recommendation and that should inform all future approaches to it.

These results also show that the task is far from solved, with the highest NDCG score of 0.406 for AAC and 0.479 for PMC, where the maximum score is 1, which leaves clear room for improvement. We are not interested in extremely fine-tuning the exact parameters for extracting the query and the external document representations using these symmetric window methods, as these could vary depending on the document collection, and tweaking a single parameter for a whole set of different cases that exhibit high variation is neither intellectually satisfying nor likely to lead to notable improvement. We also agree with Ritchie et al. (2008) that choosing a fixed window size over the whole collection is too simplistic, and we believe that a more granular approach to extracting the context and generating the query from it by using NLP techniques would highly benefit this task.

So our research branches out in two different directions from here. First, in the next two chapters we are going to explore a linguistically motivated, resource-intensive approach to classifying the citing sentences and cited sentences in our document collections and trying to finding consistent patterns of citation between these sentence types. Second, in Chapter 6, we will explore more fine-grained approaches to extracting the query from the citing context.

Chapter 4

Scientific Discourse Annotation Schemes: AZ and CoreSC

> “Artificial Intelligence is almost a humanities discipline. It’s really an attempt to understand human intelligence and human cognition."
>
> Sebastian Thrun

4.1 Introduction

In the same way that a sentence is more than the set of words that form it, a written document is much more than a collection of sentences. Any document is a linguistic artifact, which holds together as one unit of discourse, exhibiting coherence and cohesion (Bublitz, 2011). At the same time, it has a purpose: it serves one or more rhetorical aims, such as to inform or to persuade. These aims, as well as many characteristics of scientific discourse are formalised as a set of conventions in every field (Hyland, 2009), which gives scientific publications a formal structure. At the same time the language of academia requires clear argumentation (Hyland, 2009). This has led to the creation of discourse annotation schemes that attend to the rhetorical and argumentative structure of scientific papers.

A number of them have been developed over the years, and it has often been suggested that they could find application in Information Retrieval scenarios such as CCR (Teufel (2000); Nakov et al. (2004); Nanba et al. (2011)). In this chapter we present some of these schemes, focusing on the two that we employ in our experiments: Core

Scientific Concepts (CoreSC) and Argumentative Zoning (AZ), both of them sentence-based scientific discourse annotation schemes for which automatic classifiers exist.

We aim to investigate the utility of incorporating automatic argumentative and rhetorical annotation of documents for the task of CCR. By annotating each sentence in every document with AZ and CoreSC and indexing them separately by sentence class, we will be able to test and evaluate whether this results in a more useful vector-space representation of documents in our collection.

4.2 Scientific Discourse Annotation

Several annotation schemes for scientific publications have been suggested at several levels of granularity, including bio-events and meta-knowledge about them (Nawaz et al., 2010; Liakata et al., 2012b; Thompson et al., 2011) and clauses: segments of text that contain a main verb (Waard et al., 2009). However, the most common approach for schemes that pay attention to the rhetorical and argumentative structure of a publication as a whole has been to take the sentence as the minimum unit of annotation.

Annotation schemes either specifically designed for abstracts (Hirohata et al., 2008) or adapted for them have been explored (Guo et al., 2010), but these are of limited applicability in our scenario, as we are exploring citation behaviour, which occurs almost exclusively in the full text of articles.

Two of the most prominent sentence-based Scientific Discourse Annotation schemes for full text documents are Argumentative Zoning (AZ) (Teufel, 2000) and Core Scientific Concepts (CoreSC) (Liakata et al., 2012a). The former focusses on the relation between current and previous work whereas the latter mostly on the content of a scientific investigation. AZ (Table 4.2.1) frames a publication as an attempt of claiming ownership of a new piece of knowledge and aims to recover the rhetorical structure and relevant stages in this argument (Liakata et al., 2010). In contrast, CoreSC (Table 4.2.2) was specifically developed for the domain of biomedical science and treats papers as “human-readable representations of scientific investigations", aiming to retrieve the structure of the investigation from the paper (Liakata et al., 2012a), in the form of high-level concepts.

These are among the first approaches to incorporate successful automatic classification of sentences contained in a scientific paper, using a supervised machine learning approach.

When one scientific paper is cited by another, this citation serves a particular function in the article, naturally aligning with its argumentative structure. In a seminal paper, Garfield et al. (1965) suggested 15 reasons for citing other papers, including "paying homage to pioneers", "providing background reading" and "giving credit for related work". Much attention has since been paid to the analysis of the function of citations and the argumentative structure of scientific papers from different disciplines, such as applied linguistics (e.g. discourse analysis), history and sociology of science and information science (e.g. bibliometrics and information retrieval) (White, 2004). Given this amount of attention to the matter, a significant number of different categorisations for sentence classification and citation function have been proposed for different purposes. This categorisation is often observed to be very domain-specific, with the purpose and usage of types showing significant variation from one discipline to another. Integrating all citation functions into a comprehensive classification scheme therefore may not be a solved problem yet for the task at hand.

The question we are addressing is this: in order to recommend a citation for a specific textual context, can we classify the citing sentence or context in a way that aligns with the document's argumentative structure? If we then apply the same or another scheme to the contents of a document to be recommended, can we find patterns of citation between them? In order to find out, we start by exploring the most important existing scientific discourse annotation schemes and the resources available to us.

4.2.1 Argumentative Zoning

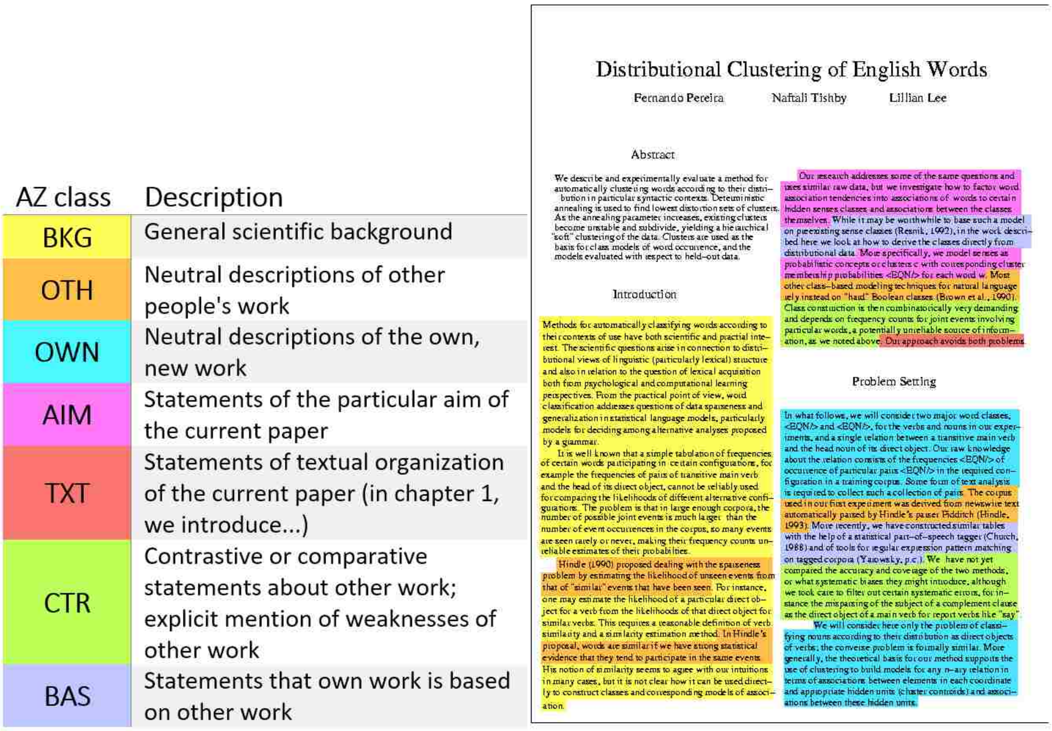
Argumentative Zoning (Teufel et al., 1999) is perhaps the first relevant body of work that has produced not just a classification and annotation scheme, but also a sizeable strictly annotated corpus based on clear guidelines and the computational means for automatically classifying sentences in a scientific document. This was developed specifically for the domain of computational linguistics, examining and annotating a sample set of 80 Computational Linguistics papers from the CMP-LG corpus[1]. The work does not strictly focus on citation function but divides the text into "zones", of which the basic building block is the sentence. The classification of these zones is

[1] http://www.itl.nist.gov/iaui/894.02/related_projects/tipster_summac/cmp_lg.html

inspired by the notion of a “knowledge claim”, as “the act of writing a paper corresponds to an attempt of claiming ownership for a new piece of knowledge, which is to be integrated into the repository of scientific knowledge in the authors’ field by the process of peer review and publication” (Teufel et al., 2009). The original work on Argumentative Zoning (AZ) (Teufel et al., 1999) defines seven categories into which each sentence has to be classified, shown in Table 4.2.1.

Zone	Description
Background (BKG)	General scientific background
Other (OTH)	Neutral descriptions of other people’s work
Own (OWN)	Neutral descriptions of the own new work
Aim (AIM)	Statements of the particular aim of the current paper. The paper’s main knowledge claim, a rhetorical move which may be repeated in the conclusion and the introduction.
Textual (TXT)	Statements of textual organisation of the current paper. Explains the physical location of information e.g. by giving a section overview or presenting a summary of a subsection. (e.g. “in chapter 1 we introduce”...)
Contrast (CTR)	Contrastive or comparative statements about other work; explicit mention of weaknesses of other work
Basis (BAS)	Statements that own work is based on other work

Table 4.2.1: Original Zones for Argumentative Zoning and their description.



AZ class	Description
BKG	General scientific background
OTH	Neutral descriptions of other people's work
OWN	Neutral descriptions of the own, new work
AIM	Statements of the particular aim of the current paper
TXT	Statements of textual organization of the current paper (in chapter 1, we introduce...)
CTR	Contrastive or comparative statements about other work; explicit mention of weaknesses of other work
BAS	Statements that own work is based on other work

Distributional Clustering of English Words

Fernando Pereira Naftali Tishby Lillian Lee

Abstract

We describe and experimentally evaluate a method for automatically clustering words according to their distribution in particular syntactic contexts. Deterministic annealing is used to find lowest distortion sets of clusters. As the annealing parameter increases, existing clusters become unstable and subdivide, yielding a hierarchical "soft" clustering of the data. Clusters are used as the basis for class models of word occurrence, and the models evaluated with respect to held–out data.

Introduction

Methods for automatically classifying words according to their contexts of use have both scientific and practial interest. The scientific questions arise in connection to distributional views of linguistic (particularly lexical) structure and also in relation to the question of lexical acquisition both from psychological and computational learning perspectives. From the practical point of view, word classification addresses questions of data sparseness and generalization in statistical language models, particularly models for deciding among alternative analyses proposed by a grammar.

It is well known that a simple tabulation of frequencies of certain words participating in certain configurations, for example the frequencies of pairs of transitive main verb and the head of its direct object, cannot be reliably used for comparing the likelihoods of different alternative configurations. The problem is that in large enough corpora, the number of possible joint events is much larger than the number of event occurrences in the corpus, so many events are seen rarely or never, making their frequency counts unreliable estimates of their probabilities.

Hindle (1990) proposed dealing with the sparseness problem by estimating the likelihood of unseen events from that of "similar" events that have been seen. For instance, one may estimate the likelihood of a particular direct object for a verb from the likelihoods of that direct object for similar verbs. This requires a reasonable definition of verb similarity and a similarity estimation method. In Hindle's proposal, words are similar if we have strong statistical evidence that they tend to participate in the same events. His notion of similarity seems to agree with our intuitions in many cases, but it is not clear how it can be used directly to construct classes and corresponding models of association.

Our research addresses some of the same questions and uses similar raw data, but we investigate how to factor word association tendencies into associations of words to certain hidden senses classes and associations between the classes themselves. While it may be worthwhile to base such a model on preexisting sense classes (Resnik, 1992), in the work described here we look at how to derive the classes directly from distributional data. More specifically, we model senses as probabilistic concepts or clusters c with corresponding cluster membership probabilities <EQN/> for each word w. Most other class–based modeling techniques for natural language rely instead on "hard" Boolean classes (Brown et al., 1990). Class construction is then combinatorically very demanding and depends on frequency counts for joint events involving particular words, a potentially unreliable source of information, as we noted above. Our approach avoids both problems.

Problem Setting

In what follows, we will consider two major word classes, <EQN/> and <EQN/>, for the verbs and nouns in our experiments, and a single relation between a transitive main verb and the head noun of its direct object. Our raw knowledge about the relation consists of the frequencies <EQN/> of occurrence of particular pairs <EQN/> in the required configuration in a training corpus. Some form of text analysis is required to collect such a collection of pairs. The corpus used in our first experiment was derived from newswire text automatically parsed by Hindle's parser Fidditch (Hindle, 1993). More recently, we have constructed similar tables with the help of a statistical part–of–speech tagger (Church, 1988) and of tools for regular expression pattern matching on tagged corpora (Yarowsky, p.c.). We have not yet compared the accuracy and coverage of the two methods, or what systematic biases they might introduce, although we took care to filter out certain systematic errors, for instance the misparsing of the subject of a complement clause as the direct object of a main verb for report verbs like "say".

We will consider here only the problem of classifying nouns according to their distribution as direct objects of verbs; the converse problem is formally similar. More generally, the theoretical basis for our method supports the use of clustering to build models for any n–ary relation in terms of associations between elements in each coordinate and appropriate hidden units (cluster centroids) and associations between these hidden units.

Figure 4.2.1: Argumentative Zoning colour scheme and an example page as annotated with it.

As the first annotated corpus was a collection of 80 Computational Linguistics papers, the annotation scheme was particularly tailored to this domain. Over several publications, several modifications to the scheme have been proposed.

Mizuta and Collier (2004) adapted the AZ scheme to biomedical text processing, with an emphasis on Information Extraction of factual information related to experimental results. They modified the original AZ scheme in three ways:

1. expanded the OWN category, adding finer-grained annotation of method, result, insight and implication
2. added Connection (CNN) and Difference (DFF) classes to cover relations between findings
3. nested the annotation, to allow for sub-classification like in the case of OWN

While this scheme is relevant, this is an example of a line of work that has produced neither a sizeable annotated corpus nor an automatic classifier we could employ for our experiments.

More recently, AZ has also been adapted to the domain of chemistry (Teufel et al., 2009), and 30 sample chemistry papers were annotated with the new scheme, AZ-II (Table 4.2.2). The authors consider inter-annotator agreement on the new scheme acceptable for computational linguistics and high for chemistry.

A comparison was made of CoreSC and AZ-II, where 36 papers were annotated using both schemes, with CoreSC by expert annotators and with AZ-II by expert-trained non-experts (Liakata et al., 2010). The overlap in annotation is studied in deep detail, examining to what degree a sentence receiving a given label in AZ-II predicts it receiving one in the CoreSC annotation. The conclusion is that the schemes are complementary but that CoreSC provides a higher level of granularity in content-level categories (Object, Goal, Hypothesis, Motivation). In contrast, AZ-II are shown to "cover aspects of knowledge claims that permeate across different CoreSC concepts" (Liakata et al., 2010). The differences are exemplified by referring to outcomes of current work versus those of other work, distinguishing methods used in previous work and in the work reported.

This repeated modification and evolution of the annotation scheme for different domains and tasks suggests that applying it across different scientific domains may benefit from future adjustments. For our experiments, however, we consider AZ the most solid starting point for the domain of computational linguistics, and CoreSC clearly the best suited for biomedical science.

4.2.2 Automatic classification

For the automatic classification of sentences into one of the categories, the original work on AZ (Teufel, 2000) reported an F-score of 72 employing a Naive Bayes classifier and a multitude of features, as shown in Table 4.2.3.

Later work by Merity et al. (2009) claims an increase in the classification accuracy to above 98% F-Score using a Maximum Entropy classifier together with other techniques such as a Gaussian prior distribution over feature weights which "allows many rare, but informative, features to be used without overfitting". They also modified the features, using all n-grams from a sentence instead of just those above a threshold, bucketed the features into smaller sets to reduce sparseness and ignored or simplified some of the features. Moreover, they used feature cut-off (removing features that occur less than four times) and treated the task as a sequence labelling task, that is, basing each prediction on the classification history.

4.2.3 Core Scientific Concepts

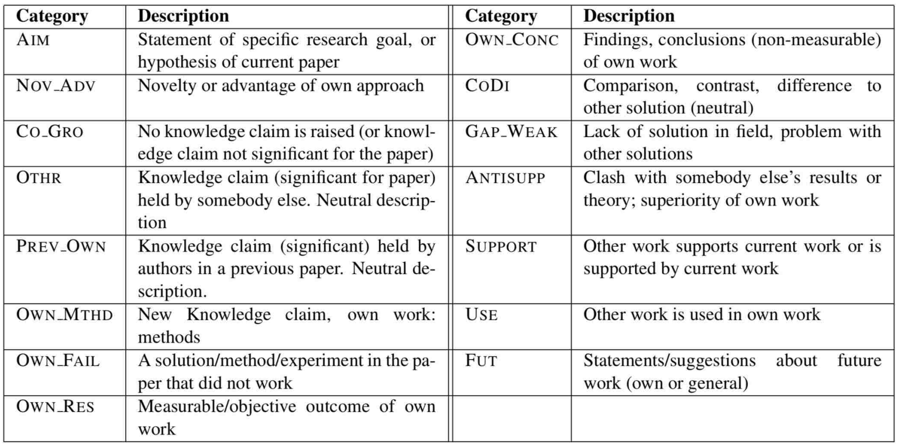
Another substantial body of work that interests us is Core Scientific Concepts (CoreSC), an annotation scheme for scientific papers, particularly tailored to the domain of biomedical science. For this annotation scheme, a number of systems were implemented that allowed for easy manual annotation and then to interface with automatically trained classifiers (Liakata et al., 2010).

The contrast that is often made between Argumentative Zoning and CoreSC is that while AZ is concerned with the citations and their function in the aim of the paper, CoreSC treats papers as “humanly-readable representations of scientific investigations” and aims to retrieve the structure of the investigation from the paper (Liakata et al., 2010).

Looking at the annotation scheme, it is immediately apparent that there exists a degree of overlap in categories. This is studied in detail in Liakata et al. (2010). The categories proposed by CoreSC and their colours are shown in Figure 4.2.2.

Sapienta[2], a Java browser-based tool, enables the easy annotation of sentences with CoreSC, and also the automatic classification of documents by a trained classifier.

[2] http://www.sapientaproject.com/



Category	Description	Category	Description
AIM	Statement of specific research goal, or hypothesis of current paper	OWN_CONC	Findings, conclusions (non-measurable) of own work
NOV_ADV	Novelty or advantage of own approach	CODI	Comparison, contrast, difference to other solution (neutral)
CO_GRO	No knowledge claim is raised (or knowledge claim not significant for the paper)	GAP_WEAK	Lack of solution in field, problem with other solutions
OTHR	Knowledge claim (significant for paper) held by somebody else. Neutral description	ANTISUPP	Clash with somebody else's results or theory; superiority of own work
PREV_OWN	Knowledge claim (significant) held by authors in a previous paper. Neutral description.	SUPPORT	Other work supports current work or is supported by current work
OWN_MTHD	New Knowledge claim, own work: methods	USE	Other work is used in own work
OWN_FAIL	A solution/method/experiment in the paper that did not work	FUT	Statements/suggestions about future work (own or general)
OWN_RES	Measurable/objective outcome of own work		

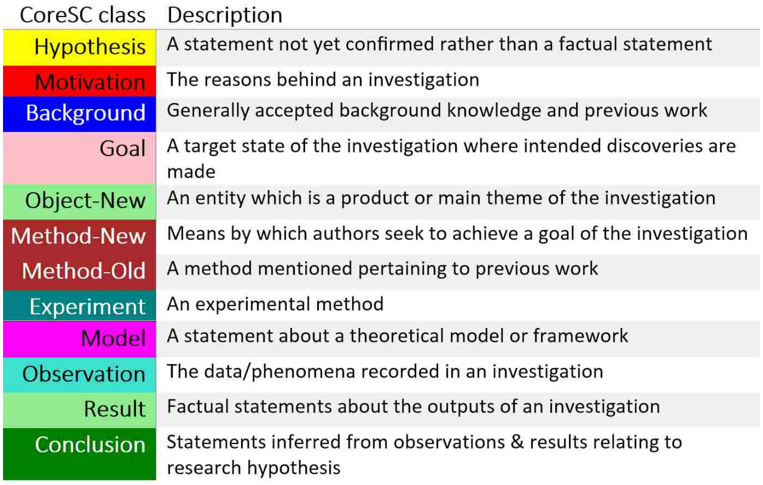
Table 4.2.2: The AZ-II scheme.

Name	Description	Type
Cont-1	Does the sentence contain “significant terms” as determined by the tf-idf measure	Yes/No
Cont-2	Does the sentence contain words in the title or heading (excluding stop words)	Yes/No
TLoc	Position of the sentence in relation to 10 segments	(A-J)
Struct-1	Position within a section	7 values
Struct-2	Relative position of sentence within a paragraph	Initial, Medial, Final
Struct-3	Type of headline of the current section	16 prototypical headlines or Non-prototypical
TLength	Is the sentence longer than N words? (N=15)	Yes/No
Syn-1	Voice of the first finite verb in the sentence	Active or Passive or No Verb
Syn-2	Tense of the first finite verb in the sentence	9 simple and complex tenses or No Verb
Syn-3	Is the first finite verb modified by a modal auxiliary?	Modal or no Modal or No Verb
Cit-1	Does the sentence contain a citation or the name of an author contained in the reference list?	Citation, Author Name or None
Cit-2	Does the sentence contain a self citation?	Yes or No or NoCitation
Cit-3	Location of citation in sentence	Beginning, Middle, End or NoCitation
Formu	Type of formulaic expression occurring in sentence	20 Types of Formulaic Expressions + 13 Types of Agents or None
Ag-1	Type of agent	13 different types or None
Ag-2	Type of action (with or without negation)	20 different Action Types X Negated/Non-negated, or None

Table 4.2.3: Original features for Argumentative Zoning as proposed in Teufel (2000).

Feature	Description
Absolute location (absloc)	Document divided into 10 unequal segments (larger segments in the middle of the paper, as in Loc of AZ)
SectionId	A sequentially incremented section number (up to 10) is assigned to each section and inherited at sentence level.
Struct-1	The location of a sentence within seven unequal segments of a section.
Struct-2	Location within a paragraph split in five equal segments.
Struct-3	One of 16 heading types assigned to a sentence by matching its section heading against a set of regular expressions (a variant on Struct-3 of AZ).
Location in section (sectionloc)	Like Struct-2 but at section level.
Length	Sentences are assigned to one of nine bins, representing a word count range.
Citation	Three cases: no citations, one citation, and two or more citations present.
History	The CoreSC category of the previous sentence.
N-grams	Binary values for significant unigrams, bigrams and trigrams.
Verb POS (VPOS)	For each verb within the sentence we determine which of the six binary POS tags (VBD, VBN, VBG, VBZ, VBP and VB) representing the tense, aspect and person of a verb are present.
Verbs	All verbs in the training data with frequency >1.
Verb Class	Ten verb classes, obtained by clustering together all verbs with a frequency >150.
Grammatical triples (Grs)	Dependency–head-dependent triples (Briscoe and Carroll format) generated using C&C tools. Using the supertagger model trained on biomedical abstracts and applied self-training on the papers in the corpus. Considered dependencies are *subj*, *dobj*, *iobj* and *obj2* with frequency > 3.
Other GR:	Subjects (Subj), direct objects (Dobj), indirect objects (Iobj) and second objects of ditransitive verbs (Obj2) with frequency >1.
Passive (P)	Whether any verbs are in passive voice.

Table 4.2.4: Core Scientific Concepts features for automatic classification, as proposed in Liakata et al. (2012a), all of them binary in nature.



CoreSC class	Description
Hypothesis	A statement not yet confirmed rather than a factual statement
Motivation	The reasons behind an investigation
Background	Generally accepted background knowledge and previous work
Goal	A target state of the investigation where intended discoveries are made
Object-New	An entity which is a product or main theme of the investigation
Method-New	Means by which authors seek to achieve a goal of the investigation
Method-Old	A method mentioned pertaining to previous work
Experiment	An experimental method
Model	A statement about a theoretical model or framework
Observation	The data/phenomena recorded in an investigation
Result	Factual statements about the outputs of an investigation
Conclusion	Statements inferred from observations & results relating to research hypothesis

Figure 4.2.2: Core Scientific Concepts classes and colour scheme.

4.3 Classification of citation function

The schemes we have seen so far aim to annotate the document at the sentence level, paying attention to the contribution of the sentence to either an argumentative aim (own work, contrast in methods) or as a characteristic of a scientific investigation (background, method, conclusion). Many other scientific discourse annotation schemes have concerned themselves instead with the *function* of the citation.

Of all schemes proposed that attend to citation function, perhaps the most relevant scheme is the Citation Function Classification (CFC) of Teufel et al. (2006), which has been influential to many others proposed after (e.g. Angrosh et al. (2013); Schäfer and Kasterka (2010); Jha et al. (2017)). This scheme builds on the Argumentative Zoning work and defines 12 categories of citation function, which can be seen in Table 4.3.1. The authors annotated a corpus of 116 articles and 2,829 citation instances and trained a classifier using a combination of features that significantly overlap with those used by AZ (Figure 4.2.3) but with the notable difference of new manually defined cue phrases.

Other examples of relevant approaches include:

- Nanba et al. (2011) take inspiration from Garfield's original 15 categories of citation function (Garfield et al., 1965), but group them together to create just three classes of citation: B (citations that show other researchers' theories or methods for the theoretical basis), C (citations that point out gaps or problems in related work) and O (anything else). They then manually define 160 rules for the classification of citation type, based on cue phrases. What is particularly interesting about this research is that they evaluate their automatic classification of citation function using bibliographic coupling (i.e. whether two given documents cite the same third one), which they measure using a retrieval approach where they index different parts of the document. They evaluate full text, title and abstract and a selection of different sentence types which they identify simply using cue phrases (e.g. a sentence is considered of type Method if it contains the words "our methods"). While much simpler than what we attempt in our own experiments and with a different purpose, there are several conceptual similarities in this prior work.

- Another early attempt at automatic classification is Garzone and Mercer (2000), where the authors define a very detailed scheme of 34 sub-categories for citation in the domains of biochemistry and physics (e.g. "Citing work totally disputes some aspect of cited work"). Their automated classification is also based on POS tags and manually selected cue words and syntactic patterns.

- Schäfer and Kasterka (2010) define a citation classification scheme for the ACL Anthology (Agree, PRecycle, Negative, Neutral, Undef) that supports a graphical navigation of the citation network.

- Angrosh et al. (2013) takes inspiration from the AZ and CFC schemes presented above and redefines another annotation scheme that includes both non-citing sentences and a taxonomy of citation types. The scheme consists of the sentence classes Background, Issues, Gaps, Description, Current Work, Outcome and Future Work and seven classes of citation, attending to fine grained differentiators such as using outputs from the cited work and contrasting results with the current paper. Other formalisations of citation types have been proposed as an ontology for publishing on the Semantic Web (Shotton, 2010).

- Jha et al. (2017) define a annotation for citation purpose with the explicit aim of evaluating and summarising the impact of papers in the ACL Anthology Network corpus, working in the domain of scientometrics. They also draw significant inspiration from the CFC scheme above and define the categories Criticizing, Comparison, Use, Substantiating, Basis and Neutral for citation function.

All of these approaches above have endeavoured to produce automatic classifiers of citation function. As we have seen, linguistic features for these classifiers are common, but features that seem to make a difference are the manually defined cue phrases and grammatical patterns that the authors define for each citation type.

Function	Description	Percentage
Weak	Weakness of cited approach	3.1%
CoCoGM	Contrast/Comparison in Goals or Methods (neutral)	3.9%
CoCo-	Author's work is stated to be superior to cited work	1.0%
CoCoR0	Contrast/Comparison in Results (neutral)	0.8%
CoCoXY	Contrast between 2 cited methods	2.9%
PBas	Author uses cited work as basis or starting point	1.5%
PUse	Author uses tools/algorithms/data/definitions	15.8%
PModi	Author adapts or modifies tools/algorithms/data	1.6%
PMot	This citation is positive about approach used or problem addressed (used to motivate work in current paper)	2.2%
PSim	Author's work and cited work are similar	3.8%
PSup	Author's work and cited work are compatible/provide support for each other	1.1%
Neut	Neutral description of cited work, or not enough textual evidence for above categories, or unlisted citation function.	62.7%

Table 4.3.1: Citation Function Classification scheme from Teufel et al. (2006) and percentage of instances of each class as manually annotated in the corpus.

4.4 Automatic classifiers

As manual annotation is prohibitively expensive for a practical application, our approach depends crucially on being able to use an automatic classifier for these annotation schemes which performs with acceptably high accuracy.

Of all the schemes for citation classification we have presented, the one that we consider the most mature and for which we could obtain the largest annotated corpus is the Citation Function Classification of Teufel et al. (2006). A classification scheme that attends to the function of a citation seems a priori promising for the task of CCR.

However, using the automated classification approach of Teufel et al. (2006) was unfortunately not feasible. No canonical implementation of a trained classifier was made available by the original authors and an initial attempt at an own implementation did not produce a useful classifier.

This is a difficult task; as originally annotated, 67% of all citations in the corpus were assigned the *Neutral* category. On the one hand, two-thirds of all citations being "neutral" calls into question the usefulness of the scheme for finding any citing patterns, and on the other, this works to completely drown out all other types in automatic classification. Further, this scheme (like others), while theoretically focussed on the citation phenomena, in practice is sentence-based: it assigns the same class to all citations in the same sentence. Given this and in the absence of a purpose-specific annotation scheme for citation instances, we decide here to use the classes from sentence-based annotation schemes as proxies for citation function. In our formulation then, we will attempt to find patterns between types of citing sentences and types of cited sentences.

4.4.1 Argumentative Zoning

For AZ we employ the AZPrime classifier [3]. This classifier was written by Dain Kaplan, while working as a Research Associate at the University of Cambridge. It is a reimplementation of Teufel's original Perl code. In spite of this work being as yet unpublished, is as close to an officially sanctioned classifier as we have obtained access to. In testing with the original AZ corpus, it yields 69% accuracy over all sentences (see Table 4.4.1 and Table4.4.2).

	Precision	Recall	F1
OWN	0.806	0.858	0.831
AIM	0.566	0.585	0.575
BKG	0.353	0.350	0.351
OTH	0.414	0.338	0.372
CTR	0.330	0.245	0.281
BAS	0.327	0.277	0.300

Table 4.4.1: AZPrime per-class precision, recall and F1 scores.

[3] The code is available at https://bitbucket.org/argzone/azprime

	OWN	AIM	BKG	OTH	CTR	BAS
OWN	**7406**	88	271	613	186	67
AIM	82	**176**	17	15	4	7
BKG	304	14	**271**	149	30	6
OTH	1050	10	149	**676**	67	46
CTR	276	12	54	99	**147**	12
BAS	65	11	6	82	11	**67**

Table 4.4.2: AZPrime (Argumentative Zoning) classifier confusion matrix

Once we have annotated the entire AAC corpus, we can see that a disproportionate number of sentences end up being labelled as OWN, to the detriment of all other zones, but most importantly OTH, CTR and importantly TXT which practically disappears.

	Manual annotation	**%**	**Automatic annotation**	**%**
BKG	720	5.80	196,057	6.35
OTH	2,013	16.21	114,570	3.71
OWN	8,433	67.89	2,705,401	87.69
AIM	209	1.68	39,040	1.27
TXT	223	1.80	2	0.00
CTR	597	4.81	18,032	0.58
BAS	227	1.83	12,167	0.39
Total	12,422	100	3,085,269	100

Table 4.4.3: Statistics of automatic annotation on AAC as performed by the AZPrime classifier, compared to the original manual annotation of the training corpus.

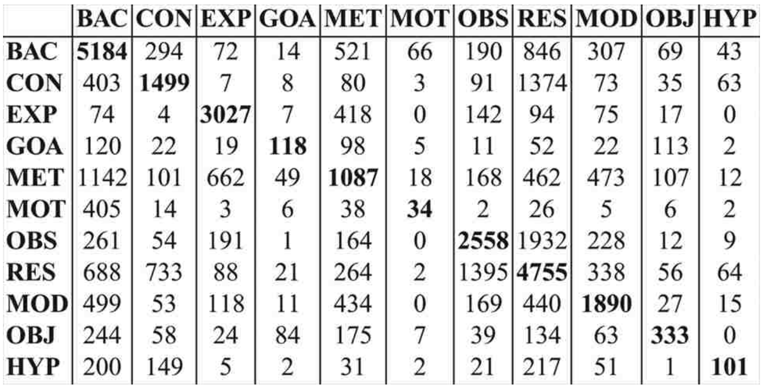
4.4.2 Core Scientific Concepts

The Sapienta classifier[4] was especially developed for CoreSC annotation of biomedical papers. It uses an SVM classifier that employs the features presented in Table 4.2.4. Evaluated using 9-fold cross-validation on the ART corpus (Liakata et al., 2010), it achieves a per-class precision of 51.6%, broken down by class in Table 4.4.4. The confusion matrix is presented in Table 4.4.5. Its performance has recently been independently tested on a different corpus from the originally annotated corpus used to train it. It yielded 51.9% accuracy over all eleven classes, improving on the 50.4% 9-fold cross-validation accuracy over its training corpus (Ravenscroft et al., 2016).

[4] http://www.sapientaproject.com

	Precision	Recall	F1
Bac	69.9	84.7	76.7
Con	58	50.3	53.9
Exp	78	89.5	83.4
Goa	60.1	49.4	54.2
Hyp	25	95.9	13.9
Met	65	52.9	58.3
Mod	0	0	0
Mot	52	38	43.9
Obj	37	21.2	27
Obs	50.3	37.6	43
Res	66.8	69.9	68.3

Table 4.4.4: Sapienta (CoreSC) classifier per-class precision, recall and F1 scores.



	BAC	CON	EXP	GOA	MET	MOT	OBS	RES	MOD	OBJ	HYP
BAC	**5184**	294	72	14	521	66	190	846	307	69	43
CON	403	**1499**	7	8	80	3	91	1374	73	35	63
EXP	74	4	**3027**	7	418	0	142	94	75	17	0
GOA	120	22	19	**118**	98	5	11	52	22	113	2
MET	1142	101	662	49	**1087**	18	168	462	473	107	12
MOT	405	14	3	6	38	**34**	2	26	5	6	2
OBS	261	54	191	1	164	0	**2558**	1932	228	12	9
RES	688	733	88	21	264	2	1395	**4755**	338	56	64
MOD	499	53	118	11	434	0	169	440	**1890**	27	15
OBJ	244	58	24	84	175	7	39	134	63	**333**	0
HYP	200	149	5	2	31	2	21	217	51	1	**101**

Table 4.4.5: Sapienta (CoreSC) classifier confusion matrix

	Manual annotation	%	Automatic annotation	%
Hyp	780	1.95	1,774,662	1.36
Mot	541	1.36	376,927	0.29
Bac	7,606	19.06	58,059,805	44.56
Goa	582	1.46	1,652,837	1.27
Obj	1,161	2.91	1,093,018	0.84
Met	4,281	10.73	18,282,904	14.03
Exp	3,858	9.67	14,546,557	11.16
Mod	3,656	9.16	9,967,284	7.65
Obs	5,410	13.55	6,507,400	4.99
Res	8,404	21.05	11,952,711	9.17
Con	3,636	9.11	6,074,544	4.66
Total	39,915	100	130,288,649	100

Table 4.4.6: Statistics of automatic annotation on PMC as performed by the Sapienta classifier, compared to the original manual annotation of the training corpus.

Annotating the AAC corpus (around 22k documents) is not too computationally demanding and can be achieved using AZPrime on a single machine in under a day. However, applying the Sapienta annotator to every document in the PMC corpus (over one million papers) was a very resource-intensive and time consuming endeavour which was only achieved in collaboration with several other researchers. This annotated corpus is as yet unpublished but has already been provided to other researchers who requested it.

4.5 Summary

This chapter has provided a review of sentence-based scientific discourse annotation schemes, and has presented in depth the two of them that we apply in our research: Argumentative Zoning and Core Scientific Concepts. We have also presented the existing work on training automated classifiers for annotating a scientific document with AZ and CoreSC, including the features and classifiers employed and statistics on corpus annotation. We present the trained classifiers we will be using to annotate our corpora and their and their accuracy. The current error rate of these automatic annotators might raise the question of whether enough signal can be found in the noise, and so whether we can indeed apply them to our CCR task. We investigate this in the next chapter, where we look for consistent citing patterns between different types of sentences in each domain.

Chapter 5

Applying Discourse Annotation Schemes to citation recommendation

> “A vital ingredient of success is not knowing that what you’re attempting can’t be done."
>
> Terry Pratchett

5.1 Introduction

Document structure and markup is often exploited at indexing time in IR scenarios, even when the document is only semi-structured (Kruschwitz, 2005). Sometimes Information Extraction techniques have been applied to this (Soderland, 1999), in the hope that indexing information at a higher level of granularity can lead to higher relevance. Other NLP techniques have also commonly found application in this domain, such as word sense disambiguation, part-of-speech tagging, stemming and lemmatisation (Lewis and Spärck Jones, 1996).

We can take this one step further. It has sometimes been suggested that scientific discourse annotation schemes such as Argumentative Zoning (AZ) and Core Scientific Concepts (CoreSC) could be applied to Information Retrieval as a way of automatically segmenting the document into semantically meaningful classes. Indeed, some experimental academic retrieval tools have tried applying them to different modes of document retrieval.

For example, Schäfer and Kasterka (2010) built a tool that enables graphical navigation through the citation graph of a small subset of the ACL Anthology, and each citation link is automatically classified according to a custom scheme that aims to capture its function to the paper. Ravenscroft et al. (2013) built another relevant experimental retrieval tool that enables to restrict the keyword search to specific types of sentences as automatically classified using CoreSC, which we also employ in our research and we present in depth below. Angrosh et al. (2013) defined another classification scheme for both sentences in an academic paper and types of citations, and applied this to building an experimental Semantic Web service to provide both search and browsing capabilities. A notable feature is a visualisation of all sentences citing a given paper presented on a timeline. We are interested in exploring the potential of more deeply integrating tools like these to provide citation recommendation as part of the writing process.

These schemes have already been experimentally applied as an adjunct to other text processing tasks, particularly in the domain of biomedical text mining; for document summarisation (see Contractor et al. (2012); Teufel and Moens (2002)), and for information extraction (see Mizuta et al. (2006)).

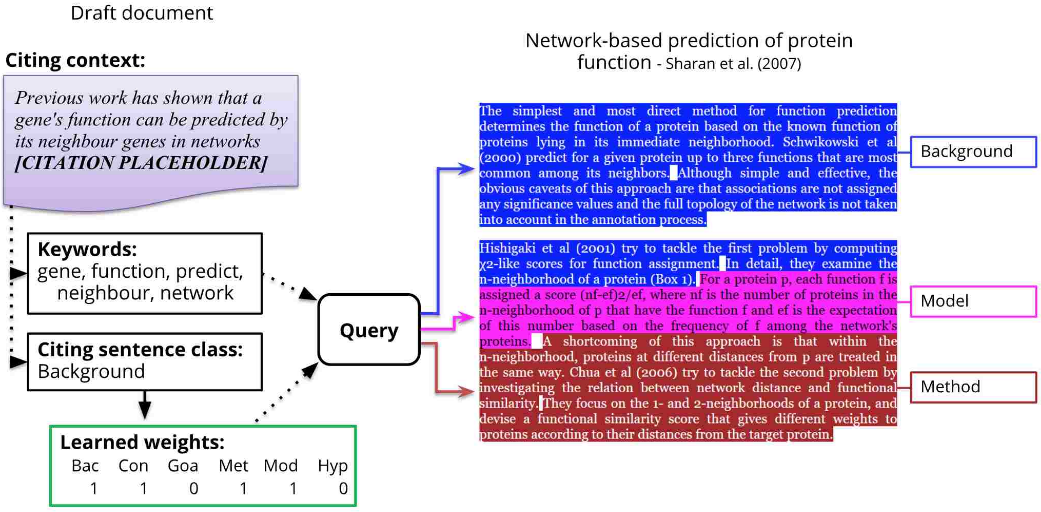
As part of this research we explore the potential application of these schemes to a deeper integration with the writing process. Our aim is to make automatic citation recommendation as relevant as possible to the author's needs and to ultimately integrate it into the authoring workflow. Automatically recommending contextually relevant academic literature can help the author identify relevant previous work and find contrasting methods and results.

Following the framework developed in previous chapters, we aim to recommend a citation for each *citation placeholder*: a special token inserted in the text of a draft paper where the citation should appear, which in our case is simply an anonymised existing citation.

The *citing sentence* is the sentence in which the prospective citation appears. It determines the function of this citation and therefore provides information that can be exploited to increase the relevance of the suggested citations. We use the class of the citing sentence as the *citation type* which in turn is the same as the *query type.*

In this chapter we explore whether the semantic categorisations developed by AZ and CoreSC can help us discover useful correlations between types of citing sentences and types of cited sentences in our corpora. We will investigate this both for document text and for the incoming link contexts, that is, the text surrounding all citations to a document. Our results show that given types of citing sentences consistently cite given types of sentences, which suggests scientific discourse annotation schemes could be exploited for this and other tasks.

5.2 Methodology



Previous work has shown that a gene's function can be predicted by its neighbour genes in networks ***[CITATION PLACEHOLDER]***

Bac	Con	Goa	Met	Mod	Hyp
1	1	0	1	1	0

The simplest and most direct method for function prediction determines the function of a protein based on the known function of proteins lying in its immediate neighborhood. Schwikowski et al (2000) predict for a given protein up to three functions that are most common among its neighbors. Although simple and effective, the obvious caveats of this approach are that associations are not assigned any significance values and the full topology of the network is not taken into account in the annotation process.

Hishigaki et al (2001) try to tackle the first problem by computing χ2-like scores for function assignment. In detail, they examine the n-neighborhood of a protein (Box 1). For a protein p, each function f is assigned a score (nf-ef)2/ef, where nf is the number of proteins in the n-neighborhood of p that have the function f and ef is the expectation of this number based on the frequency of f among the network's proteins. A shortcoming of this approach is that within the n-neighborhood, proteins at different distances from p are treated in the same way. Chua et al (2006) try to tackle the second problem by investigating the relation between network distance and functional similarity. They focus on the 1- and 2-neighborhoods of a protein, and devise a functional similarity score that gives different weights to proteins according to their distances from the target protein.

Figure 5.2.1: The intuition behind this approach. From the given *citing context*, we extract the *keywords* that will form the *query*. We treat the *citing sentence class* (as automatically labelled) as the class of citation: for each citation, we search for these keywords in different classes of sentences. In practice, this is implementing as applying different *learned weights* to different fields of the indexed document, having indexed all sentences of a single type into one field. In this example, we can see that for the citing sentence class Background, we obtain a query that searches for all keywords in sentences of classes Background, Model, Method (pictured) and Conclusion, but not in Hypothesis or Goal.

Given that we can classify each sentence according to its type according to AZ and CoreSC, we aim to find whether for each type of sentence, and so, for each type of citation, we can find a set of weights, one for each type of sentence in the documents to recommend, that will be consistent for the collection and will maximise the evaluation score.

We proceed as follows:

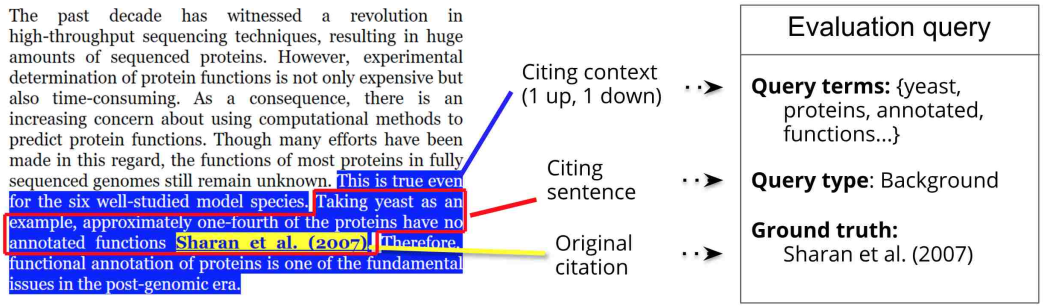
1. First, we automatically label the sentences in each document in our collection using a pre-trained CoreSC classifier for PMC and an AZ one for AAC. As we motivated in Chapter 4, each annotation scheme was developed for a given scientific domain, and we apply each to the corpus that is most in-domain.

2. We then index these documents: for each document in the Lucene index we create a separate indexing field for each class of sentence. We index all sentences of the same class into the same field for each document. That is, we index a bag-of-words for each class (Hypothesis, Background, Method, etc.), which contains

all the words from all the sentences of that class present inside the document (in the case of internal methods).

3. We extract the query from the context of a resolvable citation. We label the *insertion context* with a rhetorically-motivated class that encodes the citation function. In our implementation, this is just the class of the sentence as classified in the previous step, so a citation's type is the CoreSC/AZ class of the sentence containing it. See Figure 5.2.2 for an illustration of this.

4. We test different weights for each citation type/query type (as labelled in step 3) and compare the results with the baseline of using all weights equally set to 1. The evaluation method is described below. We evaluate by performing K-fold cross-validation and comparing the results over all fold combinations. What we expect to find is not only improvement on average in the scores for a particular citation type over the baseline, but consistency across folds in weights and obtained improvement.



The past decade has witnessed a revolution in high-throughput sequencing techniques, resulting in huge amounts of sequenced proteins. However, experimental determination of protein functions is not only expensive but also time-consuming. As a consequence, there is an increasing concern about using computational methods to predict protein functions. Though many efforts have been made in this regard, the functions of most proteins in fully sequenced genomes still remain unknown. This is true even for the six well-studied model species. Taking yeast as an example, approximately one-fourth of the proteins have no annotated functions Sharan et al. (2007). Therefore, functional annotation of proteins is one of the fundamental issues in the post-genomic era.

Citing context
(1 up, 1 down)
Citing
sentence
Original
citation
Evaluation query
Query terms: {yeast, proteins, annotated, functions...}
Query type: Background
Ground truth: Sharan et al. (2007)

Figure 5.2.2: Illustration of a query being extracted for the context of a resolvable citation in the PMC corpus.

Our hypothesis is that the relevance of suggested citations can be increased by applying a set of automatically-trained per-field weights to the similarity function, which depend on the class of the citing sentence, as automatically classified. That is, we attempt to find the best combination of weights to set for each class of citing sentence, which we treat as a proxy for the citation function. For example, we may find that for all citations of type Method, if we gave higher weight to the content of sentences of type Method in the documents in the collection, we would achieve a higher accuracy (see Figure 5.2.1 for an illustration).

5.3 Experimental setup

Corpus split: These experiments employ the exact same corpus split for AAC, with a with cut-off year of 2010 and so the same documents in the document collection and in the test set as those found in Chapter 3. However, for PMC we use two different indices. For testing the internal methods (Section 5.5), we build our index in the same way as in Chapter 3. However, for testing with incoming link contexts (Section 5.6), we use 2014 as the year for the split instead of 2013. We observed that allowing for documents from 2014 instead of 2013 to be indexed and limiting our queries to documents from 2015 and beyond would be helpful with increasing the number of sentences in the inlink contexts for each document and the variety of sentence classes in these contexts.

Set of queries: For both AAC and PMC, we select an entirely different set of resolvable citations from which to generate queries for these experiments. We define severaldistinct sets of queries for each collection:

- AAC Queries A: 1,000 queries used in the Chapter 3 experiments
- PMC Queries A: 1,000 queries used in the Chapter 3 experiments
- AAC Queries B: 3,951 queries used in all Chapter 5 experiments
- PMC Queries B: 1,955 queries used in the Section 5.5 experiments
- PMC Queries C: 6,112 queries used in the Section 5.6 experiments

Resolvable citations: We want to generate sufficient numbers of queries of each type, but as we saw in the previous chapter, the ratios of different sentence types as automatically labelled can be very skewed. In order to maximise the number of queries of the less prevalent types, for the Queries B and C sets we select a larger and entirely different subset of resolvable citations to generate queries from. Where before we sorted the test documents by the amount of collection-internal references in descending order and generated the queries one by one, here we extract queries until we reach $1,000$ queries of each type or exhaust the test set. While it is not feasible to obtain $1,000$ citing sentences of each type, we do make an effort to balance the classes in the set as much as possible.

Annotation: All documents in each collection are annotated using the relevant automated classifier: Sapienta for PMC and AZPrime for AAC, as presented in Chapter 4.

Context extraction: As a baseline approach, we extract the context of a citation using a symmetric window of 3 sentences: 1 before the citation, the sentence containing the citation and 1 after. This is a frequently applied method (Huang et al., 2015a; Ritchie et al., 2008) and as we saw in Chapter 3 it is roughly equivalent in performance to the best performing ones, both for AAC (0.403 NDCG vs 0.406 for window30_30) and PMC (0.478 NDCG vs 0.479 for window30_20).

Query generation: The *query* is then formed of all the terms in the citation's context, excluding those found in a short list of stopwords[1]. Lucene queries take the basic form *field:term*, where each combination of *field* and *term* form a unique term in the query. We want to match the set of extracted terms to all fields in the document, as each field represents one class of CoreSC.

We apply no stemming or lemmatisation in these experiments, as we have observed in other preliminary experiments that this lowers the overall scores across the board. In these experiments, we are in a sense "overfitting" by taking advantage of the lexical priming in effect: authors of the citing document are likely to use the exact terms that were used in the cited document, which may be infrequent in the collection, therefore boosting the score of the cited paper. For the experiments we present in this chapter, this in fact aligns with our objective: where this lexical priming is in effect, it will help identify the cited sentences in the cited papers, allowing us to better find the patterns of citation.

Similarity formula: The Lucene DefaultSimilarity formula gives a boost to a term matching across multiple fields (see Chapter 1), which in our case would be likely to introduce spurious results. That is, a document could receive a higher score simply because a term appears in sentences of more classes, and this itself is noisy given the noisy automatic annotation. To avoid this, we employ a type of query called DisjunctionMax, where only the top scoring *field:term* match is evaluated out of a number of fields. Therefore, having one query term for each of the classes of CoreSC for each distinct token (e.g. *Bac:"method"*, *Goa:"method"*, *Hyp:"method"*, etc.), only the one with the highest score will be evaluated as a match. This avoids the scores being spuriously increased by a spread of terms over types of sentences.

[1] *~, the, and, or, not, of, to, from, by, with, a, an*

5.4 Weight training

We want to find the optimal query weight to apply to a term to be found in a document sentence of type Y given a query of type X. Testing all possible weight combinations is infeasible due to the combinatorial explosion, so we adopt the greedy heuristic of trying to maximise the objective function at each step.

Our weight training algorithm can be summarised as “hill climbing with restarts". For each fold, and for each citation type, we aim to find the best combination of weights to set on sentence classes that will maximise our metric, in this case the NDCG score that we compute by trying to recover the original citation. We keep the queries the same in structure and term content and we only change the weights applied to each field in a document to recommend. Each field, as explained above, contains only the terms from the sentences in the document of one CoreSC/AZ class.

The weights are initialised at 1 and they move by -1, 6, and -2 in sequence[2], going through a minimum of 3 iterations. Each time a weight movement is applied, it is only kept if the score increases. Otherwise the previous weight value is restored.

This simple algorithm is not guaranteed to find a globally optimal combination of parameters for the very complex function we are optimising, but it is sufficient for our current objective and its simplicity allows us to work with the huge delay in evaluating the results every single time.

One more optimisation we include is that we do not re-run the full set of queries[3] at every change of weights, as this would make this approach impossible to run in a reasonable time on our available hardware. These are the steps of our method:

1. Run the set of queries a single time with all weights set to 1, which is our baseline.

2. Retrieve the top $K = 200$ documents as usual, and for each document in this set we then retrieve its *explanation.* IR engines like Lucene provide explanations of matches between a query and a document. In the case of Lucene these are very detailed, listing for each term matched in each field, what its score is due to.

3. Extract from the explanation the formula that the score was computed with in a structured format, for each of the top 200 results, for each of the queries.

[2] These specific values were set empirically after a number of experiments.

[3] In our 4-fold cross-validation setup, this would be $3/4$ of all queries of the given type, e.g. $750/1000$ in the case of CoreSC type Bac.

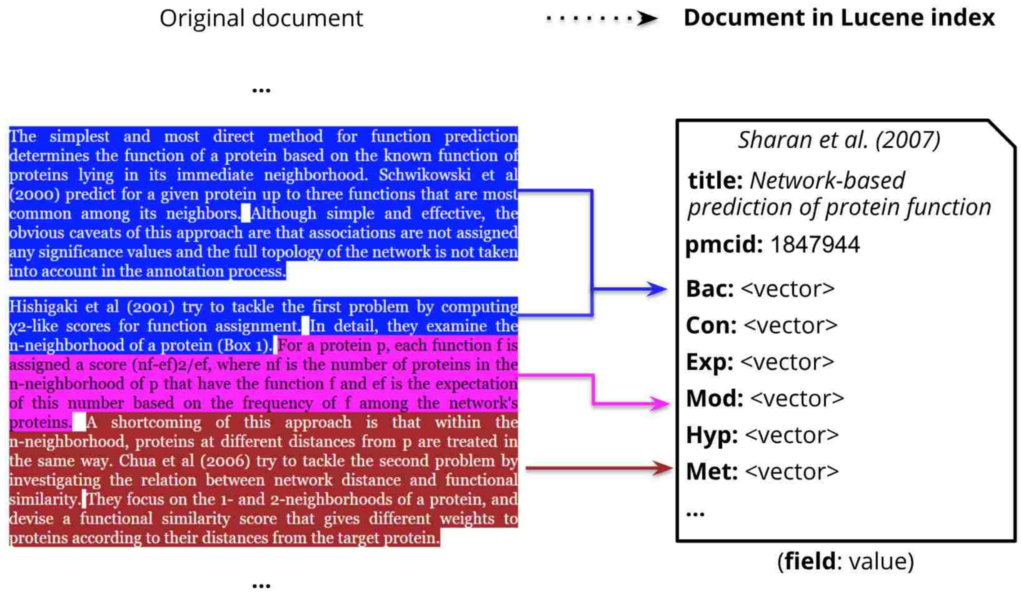
4. Search the space of weights in order to optimise the score by reordering those 200 results, separately from the retrieval engine but simulating the computations it would perform.

Given that we are only reranking the top 200 results, this is an approximation of the real scores. However, given that we are reranking the top 200 results for several hundred citations at each iteration, this is sufficient for our purpose of finding weights that consistently maximise the score.

We need to test whether our conditional weighting of text spans based on CoreSC classification is actually reflecting some underlying truth and is not just a random effect of the dataset. To this end, we perform K-fold cross-validation on the corpus (for $K = 4$), where we learn the weights for $K - 1$ folds and test their impact on one fold. Our main interest in this experiment is consistency across folds.

5.5 Internal methods

As in the experiments in Chapter 3, the document collection is indexed using the Lucene retrieval engine through the interface provided by elasticsearch. For each document, we create one field for each sentence class (11 for CoreSC, 7 for AZ), and index into each field all the words from all sentences in the document that have been labelled with that class, as represented in Figure 5.5.1.



The simplest and most direct method for function prediction determines the function of a protein based on the known function of proteins lying in its immediate neighborhood. Schwikowski et al (2000) predict for a given protein up to three functions that are most common among its neighbors. Although simple and effective, the obvious caveats of this approach are that associations are not assigned any significance values and the full topology of the network is not taken into account in the annotation process.

Hishigaki et al (2001) try to tackle the first problem by computing χ2-like scores for function assignment. In detail, they examine the n-neighborhood of a protein (Box 1). For a protein p, each function f is assigned a score (nf-ef)2/ef, where nf is the number of proteins in the n-neighborhood of p that have the function f and ef is the expectation of this number based on the frequency of f among the network's proteins. A shortcoming of this approach is that within the n-neighborhood, proteins at different distances from p are treated in the same way. Chua et al (2006) try to tackle the second problem by investigating the relation between network distance and functional similarity. They focus on the 1- and 2-neighborhoods of a protein, and devise a functional similarity score that gives different weights to proteins according to their distances from the target protein.

Figure 5.5.1: Illustration of per-class indexing of sentences in the full text of documents in the document collection.

5.5.1 CoresSC and PMC

First we run each query assigning a default weight of 1 to each field that represents one type of sentence in the target document, and the score we obtain is our baseline. We can immediately see that these scores (Table 5.5.1) are much lower than those we obtained with our project-wide baseline in Chapter 3 (0.42 NDCG for full_text and sentence_1up_1down, which match the current setup), where all document text was indexed as a single bag-of-words. This shows that we are not taking full advantage of the weighting scheme when indexing by sentence class.

However, in these experiments we do not aim to improve on the overall scores obtained in Chapter 3, but rather to discover whether there are relationships to find between types of citing sentences and types of cited sentences. We believe that if we find consistent connections between these, we can then exploit this knowledge to increase the relevance of recommendations as reflected in the evaluation scores.

Query type	# queries	Avg NDCG	Avg top-1 accuracy
Bac	700	0.232	0.113
Con	133	0.248	0.159
Exp	16	0.308	0.188
Goa	44	0.130	0.068
Hyp	31	0.290	0.188
Met	602	0.207	0.100
Mod	161	0.308	0.174
Mot	8	0.000	0.000
Obj	65	0.065	0.031
Obs	17	0.410	0.213
Res	178	0.239	0.107

Table 5.5.1: Baseline scores for CoreSC/PMC queries.

Figure 5.5.2 shows the results of evaluating with 4-fold cross-validation by citation type, ordered by number of folds improved and standard deviation. The citation type is the CoreSC class of the sentence containing the citation. We can see that 6 out of 11 types of citation exhibit improvement across all folds, and there is a relationship between the standard deviation of the improvement in scores and the number of citations of that type.

Figure 5.5.3 expands on this and shows the best weight values that were found for each fold, for all 6 citation types for which there was improvement on all folds. On the right-hand side are the scores obtained *after testing* and the percentage increase over the baseline, in which all weights are set to 1.

Type	Citations	Fold 1	Fold 2	Fold 3	Fold 4	**Folds improved**	**% impr.**	**Std. Dev.**
Con	133	6.63	16.96	4.24	9.84	4	9.42	5.53
Bac	700	28.77	7.56	28.32	24.44	4	22.27	10.00
Met	602	23.70	43.78	18.44	19.28	4	26.30	11.88
Res	178	33.87	54.18	7.28	30.95	4	31.57	19.21
Goa	44	73.74	46.47	32.38	28.40	4	45.25	20.52
Obj	65	547.57	19.32	91.31	18.30	4	169.13	254.60
Mod	161	-3.64	17.59	32.66	18.33	3	16.24	14.96
Obs	17	-18.78	37.06	9.57	102.40	3	32.56	51.84
Exp	16	-23.04	46.50	105.23	31.63	3	40.08	52.73
Hyp	31	43.14	-37.93	-38.43	6.03	2	-6.80	39.28
Mot	8	0	0	0	0	0	0.00	0.00

Figure 5.5.2: Results of evaluating with 4-fold cross-validation by citation type, ordered by number of folds showing improvement and by standard deviation. The citation type is the CoreSC class of the sentence containing the citation. In bold, citation types for which there was improvement across all folds. Percentile improvement is the average across the 4 folds.

			Weights											Scores	
Type	Citations	Fold	Bac	Con	Exp	Goa	Hyp	Met	Mod	Mot	Obj	Obs	Res	NDCG*	Accuracy*	% imp.
Bac	700	1	1	1	1	0	0	1	1	0	0	0	1	0.27825	0.183	**28.77**
		2	1	0	1	0	0	1	0	0	0	1	1	0.30492	0.160	**7.56**
		3	1	1	0	0	0	1	1	0	0	1	1	0.23861	0.109	**28.32**
		4	1	1	1	0	0	1	1	0	0	0	1	0.30314	0.143	**24.44**
Con	133	1	1	1	1	0	0	0	0	0	0	0	1	0.14824	0.059	**6.63**
		2	1	1	0	0	0	1	1	0	0	0	0	0.36623	0.212	**16.96**
		3	1	0	1	0	0	1	0	0	0	0	0	0.3399	0.242	**4.24**
		4	1	1	1	0	0	1	1	0	0	0	0	0.23478	0.152	**9.84**
Exp	16	1	7	1	0	1	0	4	1	0	0	1	0	0.28861	0.250	**-23.04**
		2	6	7	0	1	0	0	1	0	0	1	0	0.15773	0.000	**46.50**
		3	1	1	0	0	0	0	1	0	0	1	0	0.3	0.250	**105.23**
		4	1	1	0	0	0	0	0	0	0	0	0	0.79453	0.750	**31.63**
Goa	44	1	1	0	1	0	0	1	0	0	0	0	0	0.3181	0.182	**73.74**
		2	1	0	0	0	0	1	0	0	0	0	0	0.08766	0.000	**46.47**
		3	1	0	0	0	0	1	0	0	0	1	0	0.2142	0.182	**32.38**
		4	1	0	0	0	0	1	0	0	0	0	0	0.14827	0.091	**28.40**
Hyp	31	1	1	0	1	0	1	1	1	0	0	1	1	0.03763	0.000	**43.14**
		2	7	0	1	0	0	1	0	0	0	1	1	0.31037	0.125	**-37.93**
		3	7	0	1	0	0	1	1	0	0	1	1	0.15392	0.125	**-38.43**
		4	1	1	1	0	1	1	1	0	0	1	1	0.40836	0.000	**6.03**
Met	602	1	1	0	0	0	0	1	0	0	0	0	1	0.27991	0.146	**23.70**
		2	2	0	0	0	0	0	1	0	0	0	0	0.23014	0.139	**43.78**
		3	2	0	0	0	0	1	0	0	0	0	1	0.32347	0.173	**18.44**
		4	2	0	0	0	0	0	1	0	0	0	0	0.19959	0.107	**19.28**
Mod	161	1	2	0	0	1	0	0	1	0	0	0	0	0.22371	0.073	**-3.64**
		2	3	0	0	0	0	0	1	0	0	0	0	0.47049	0.325	**17.59**
		3	2	0	0	0	0	0	1	0	0	0	0	0.41699	0.225	**32.66**
		4	1	1	0	0	0	1	1	0	0	1	1	0.33979	0.200	**18.33**
Obj	65	1	7	0	1	0	0	1	0	0	0	1	1	0.05882	0.059	**547.57**
		2	7	0	1	1	0	1	0	0	0	1	1	0.08476	0.063	**19.32**
		3	7	0	1	0	0	1	0	0	0	1	7	0.07878	0.000	**91.31**
		4	13	0	1	1	0	4	0	0	0	1	0	0.16296	0.063	**18.30**
Obs	17	1	7	0	0	0	0	7	0	0	0	0	1	0.63851	0.200	**-18.78**
		2	1	0	0	0	0	0	0	0	0	0	1	0.55885	0.500	**37.06**
		3	1	0	0	0	0	1	0	0	0	0	1	0.40913	0.250	**9.57**
		4	1	0	0	1	0	1	0	0	0	0	1	0.14499	0.000	**102.40**
Res	178	1	1	0	0	0	0	0	0	0	0	1	1	0.26249	0.111	**33.87**
		2	1	0	0	0	0	0	0	0	0	1	1	0.24294	0.133	**54.18**
		3	1	0	0	0	0	0	0	0	0	0	1	0.44217	0.273	**7.28**
		4	1	0	0	0	0	0	0	0	0	0	1	0.25063	0.091	**30.95**

Figure 5.5.3: Weight values for the citation types that improved across all folds. The weight values for the 4 folds are shown, together with test scores and improvement over the baseline. The weight cells are shaded according to their value, darker is higher. In bold, citation types that consistently improve across folds. On the right-hand side are the scores obtained through testing and the percentage increase over the baseline, in which all weights were set to 1. *NDCG and Accuracy (top-1) are averaged scores over all citations in the test set for that fold.

As is to be expected, the citations are skewed in numbers towards some CoreSC classes. A majority of citations occur within sentences that were automatically labelled Background and Methodology, no doubt due to a pattern in the layout of the content of articles. This yields many more Background and Methodology citations to evaluate on, and for this reason we set a hard limit to the number of citations per zone to 700 in these experiments.

While these are initial results, a number of patterns are immediately evident. For all citation functions, it seems to be universally useful to know that the candidate document matches the query term in sentences from its Background sentences or Method sentences. It is also possible that this is partly an effect of there being more sentences of type Background and Method in a candidate paper.

Similarly, it seems it is better to ignore other classes of sentences in candidate papers, such as Motivation and Observation. An important thing to remind ourselves here is that when a weight combination was found where the best weight for a CoreSC class is 0, it does not mean that including information from this zone is *not useful* but rather that it is in fact *detrimental*, as eliminating it actually increased the NDCG score.

Interestingly, for citations of type Result, only Background, Result and, to a lesser degree Observations sentences in candidate documents seem to contain useful information. This is not surprising, and it allows for easy interpretation: when reporting results, these are often compared with previous results reported in other papers.

The degree of consistency varies across citation types. For Background, Contrast, Goal, Methodology, Object and Result, improvements are found at each fold and it seems that some consistency can be found in the trained weights. These are also types with a larger number of citations available. Experiment, Hypothesis, Model and Observation are the ones that are inconsistent in improvement: for some folds, the trained weights actually decrease the score, which suggests that no clear pattern is to be found. These are generally classes with fewer citations available, which could go some way towards explaining this. However, the exception here is Model, which, in spite of a important number of citations (161), still exhibits inconsistency, with the first fold decreasing in score.

It is important to note that our evaluation pipeline necessarily consists of many steps, and encounters issues with XML conversion, matching of citations with references, matching of references in papers to references in the collection, etc., where each step in the pipeline introduces a degree of error that we have not estimated here. Perhaps the single most significant one is that of the automatic sentence classifier.

Query type	# queries	Avg NDCG	Avg Precision
Bac	700	0.232	0.113
Con	133	0.248	0.159
Exp	16	0.308	0.188
Goa	44	0.130	0.068
Hyp	31	0.290	0.188
Met	602	0.207	0.100
Mod	161	0.308	0.174
Mot	8	0.000	0.000
Obj	65	0.065	0.031
Obs	17	0.410	0.213
Res	178	0.239	0.107

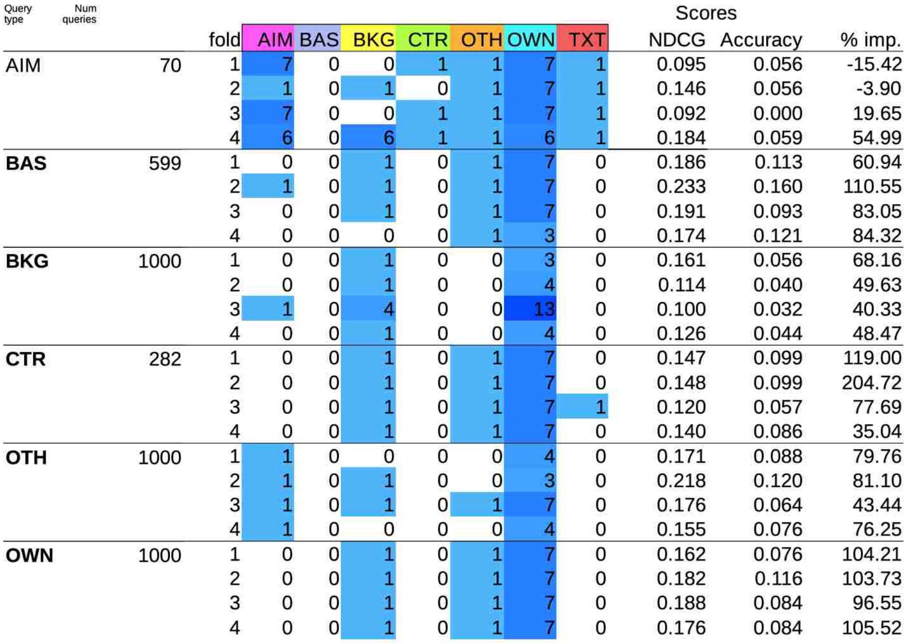
Table 5.5.2: Baseline scores for AZ/AAC queries.

We judge that the consistency of correlations we find confirms that what we can see in Figure 5.5.3 is not due to random noise, but rather hints at underlying patterns in the connections between scientific articles in the corpus. This also seems to confirm our assumption that the CoreSC class of the sentence that a citation appears in can be used as a proxy for the function of this citation.

5.5.2 Argumentative Zoning and AAC

For AAC and AZ, it is only the queries of type AIM that do not show consistent improvement. For all the other types, the results are presented below in Figure 5.5.4. From these results it can be seen that sufficient consistency can be found in the zones for which higher weights lead to higher scores, in the same way as for CoreSC-annotated documents in the previous section. Some noteworthy observations are:

- Sentences of type OWN are universally found to be useful, and in fact consistently converge on very high weights. This is unsurprising given that, as we saw in Chapter 4, 87.69% of all sentences in AAC are classified as OWN, which partly goes to explain the huge mass of weight that ends up assigned to sentences of that type.
- However, representing a much lower ratio of AAC sentences, 6.35% for BKG and 3.71% for OTH, these two zones receive much of the weight mass across all zones. We can see the weights going to these zones being sonsistent enough across query types. Sentences of type AIM are also notable for converging on consistent weights for all folds of queries of type OTH.



Query type	Num queries	fold	AIM	BAS	BKG	CTR	OTH	OWN	TXT	Scores		
										NDCG	Accuracy	% imp.
AIM	70	1	7	0	0	1	1	7	1	0.095	0.056	-15.42
		2	1	0	1	0	1	7	1	0.146	0.056	-3.90
		3	7	0	0	1	1	7	1	0.092	0.000	19.65
		4	6	0	6	1	1	6	1	0.184	0.059	54.99
BAS	599	1	0	0	1	0	1	7	0	0.186	0.113	60.94
		2	1	0	1	0	1	7	0	0.233	0.160	110.55
		3	0	0	1	0	1	7	0	0.191	0.093	83.05
		4	0	0	0	0	1	3	0	0.174	0.121	84.32
BKG	1000	1	0	0	1	0	0	3	0	0.161	0.056	68.16
		2	0	0	1	0	0	4	0	0.114	0.040	49.63
		3	1	0	4	0	0	13	0	0.100	0.032	40.33
		4	0	0	1	0	0	4	0	0.126	0.044	48.47
CTR	282	1	0	0	1	0	1	7	0	0.147	0.099	119.00
		2	0	0	1	0	1	7	0	0.148	0.099	204.72
		3	0	0	1	0	1	7	1	0.120	0.057	77.69
		4	0	0	1	0	1	7	0	0.140	0.086	35.04
OTH	1000	1	1	0	0	0	0	4	0	0.171	0.088	79.76
		2	1	0	1	0	0	3	0	0.218	0.120	81.10
		3	1	0	1	0	1	7	0	0.176	0.064	43.44
		4	1	0	0	0	0	4	0	0.155	0.076	76.25
OWN	1000	1	0	0	1	0	1	7	0	0.162	0.076	104.21
		2	0	0	1	0	1	7	0	0.182	0.116	103.73
		3	0	0	1	0	1	7	0	0.188	0.084	96.55
		4	0	0	1	0	1	7	0	0.176	0.084	105.52

Figure 5.5.4: Results for AZ-annotated document contents

5.6 External methods

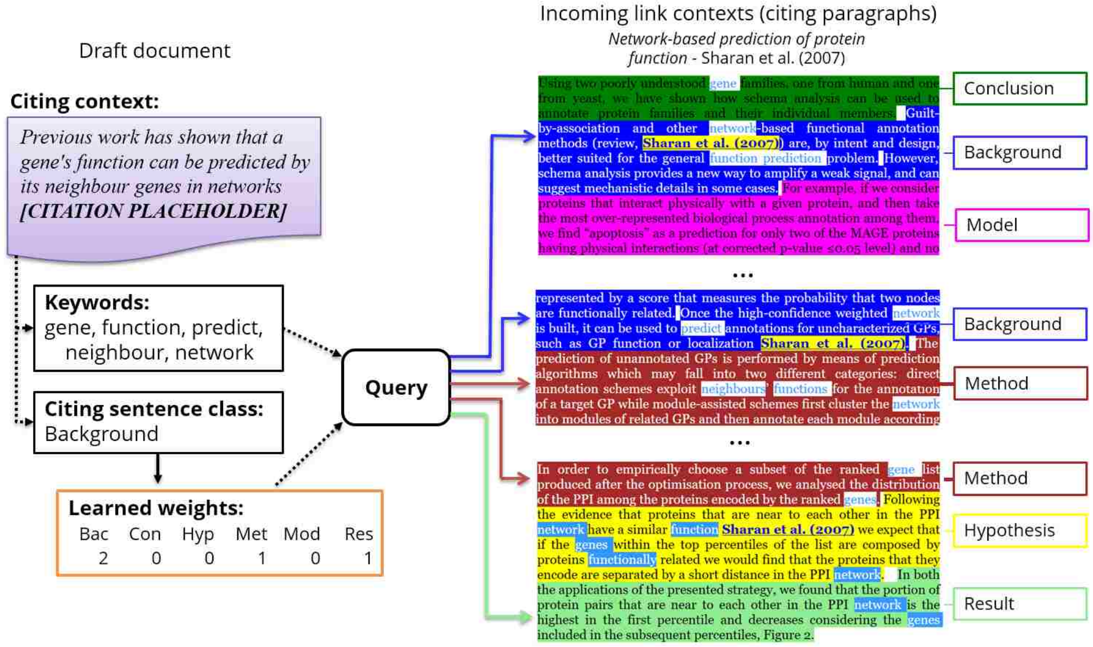
In the previous section we have seen experiments where we tried to find consistent per-sentence-class weights for the different sentence types in an in-collection document so as to maximise the retrieval accuracy score given the type of citing sentence. These weights are applied to the different fields of a document, which are populated with internal methods, and so with the contents of the paper itself.

Now we can take this one step further and investigate whether we can find similarly consistent weights that apply to the class of sentence found in the incoming link contexts of citations to a document. Figure 5.6.1 illustrates this approach.

In practical terms, this means that instead of indexing all of these sentences into a single field as a single bag-of-words, or combine them with the full_text, we index them by type into a different field for each type, as illustrated in Figure 5.6.2.

5.6.1 Core Scientific Concepts and PMC

Similarly clear yet different patterns are visible for CoreSC-annotated Incoming Link Contexts below. We show the 7 classes for which we find consistent improvement out of 11.



Previous work has shown that a gene's function can be predicted by its neighbour genes in networks ***[CITATION PLACEHOLDER]***

Using two poorly understood gene families, one from human and one from yeast, we have shown how schema analysis can be used to annotate protein families and their individual members. Guilt-by-association and other network-based functional annotation methods (review, **Sharan et al. (2007)**) are, by intent and design, better suited for the general function prediction problem. However, schema analysis provides a new way to amplify a weak signal, and can suggest mechanistic details in some cases. For example, if we consider proteins that interact physically with a given protein, and then take the most over-represented biological process annotation among them, we find "apoptosis" as a prediction for only two of the MAGE proteins having physical interactions (at corrected p-value ≤0.05 level) and no

...

represented by a score that measures the probability that two nodes are functionally related. Once the high-confidence weighted network is built, it can be used to predict annotations for uncharacterized GPs, such as GP function or localization **Sharan et al. (2007)**. The prediction of unannotated GPs is performed by means of prediction algorithms which may fall into two different categories: direct annotation schemes exploit neighbours' functions for the annotation of a target GP while module-assisted schemes first cluster the network into modules of related GPs and then annotate each module according

...

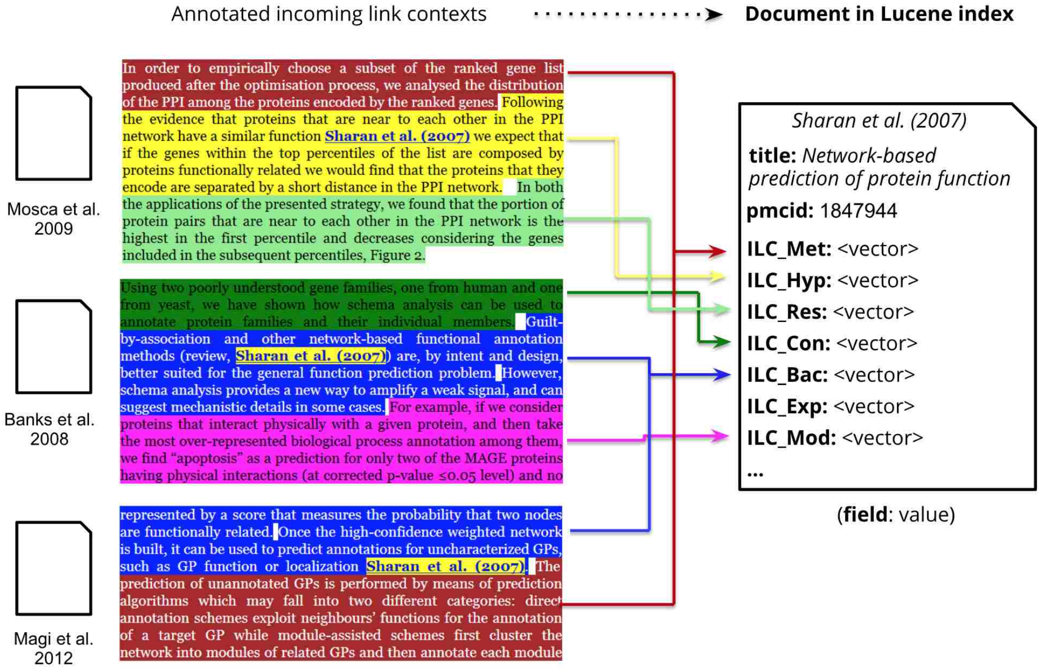
In order to empirically choose a subset of the ranked gene list produced after the optimisation process, we analysed the distribution of the PPI among the proteins encoded by the ranked genes. Following the evidence that proteins that are near to each other in the PPI network have a similar function **Sharan et al. (2007)** we expect that if the genes within the top percentiles of the list are composed by proteins functionally related we would find that the proteins that they encode are separated by a short distance in the PPI network. In both the applications of the presented strategy, we found that the portion of protein pairs that are near to each other in the PPI network is the highest in the first percentile and decreases considering the genes included in the subsequent percentiles, Figure 2.

Figure 5.6.1: A high-level illustration of the approach. The citation type, and in turn the *query type* determines the set of weights to apply to the classes of sentences in the anchor paragraphs of links to documents in the collection. In this example, for Bac, only 3 classes have non-zero weights: Bac, Met and Res. The extracts from 3 different citing papers, exemplify terms matching in different classes of sentences.

Query type			Weights											Scores		
	Citations	Fold	Bac	Con	Exp	Goa	Hyp	Met	Mod	Mot	Obj	Obs	Res	NDCG*	Accuracy*	% imp.
Bac	1000	1	2	0	0	0	0	1	0	0	0	0	1	0.201	0.120	14.54
		2	2	1	0	0	0	1	1	0	0	0	1	0.149	0.080	36.22
		3	2	0	0	0	0	1	0	0	0	0	0	0.187	0.100	31.68
		4	2	0	0	0	0	1	0	0	0	0	1	0.145	0.068	21.07
Con	278	1	2	0	0	0	0	1	0	0	0	0	1	0.167	0.100	24.14
		2	2	0	0	0	0	1	0	0	0	0	0	0.173	0.100	27.74
		3	7	6	0	0	0	1	0	0	0	0	0	0.080	0.014	30.03
		4	7	0	0	0	0	0	0	0	0	0	0	0.103	0.072	7.78
Goa	49	1	7	1	0	1	0	1	0	0	1	0	0	0.088	0.000	113.27
		2	6	1	0	1	0	1	0	1	1	0	0	0.218	0.167	49.28
		3	7	1	0	1	0	1	0	0	1	0	0	0.174	0.083	11.50
		4	7	1	0	0	0	1	0	0	0	0	0	0.103	0.083	23.14
Hyp	91	1	7	1	0	0	0	1	1	0	0	0	1	0.126	0.087	201.92
		2	7	1	0	0	0	1	0	0	0	0	1	0.166	0.087	53.01
		3	7	1	0	0	0	0	1	0	0	0	1	0.145	0.087	36.77
		4	7	0	0	0	0	1	1	0	0	0	1	0.179	0.045	85.81
Met	893	1	1	1	0	0	0	1	0	0	0	1	0	0.227	0.138	2.92
		2	1	0	0	0	1	1	0	0	0	0	1	0.227	0.130	11.39
		3	1	0	0	0	0	1	0	0	0	0	1	0.261	0.139	9.35
		4	1	1	0	0	0	1	0	0	0	0	1	0.203	0.112	10.70
Obj	70	1	1	0	0	0	0	1	0	0	0	0	1	0.103	0.056	23.29
		2	1	0	0	0	0	1	0	0	0	0	1	0.210	0.167	29.32
		3	1	0	0	0	0	1	0	0	0	0	1	0.195	0.118	8.04
		4	1	0	0	0	0	1	0	0	0	0	1	0.109	0.059	40.53
Res	420	1	2	1	1	0	0	0	0	0	0	0	1	0.104	0.057	22.81
		2	2	1	0	0	0	1	0	0	0	1	1	0.152	0.086	36.52
		3	2	1	0	0	0	0	0	0	0	0	1	0.196	0.133	34.13
		4	7	0	0	0	1	1	1	0	0	0	1	0.183	0.105	3.44

Figure 5.6.3: Results for CoreSC-annotated Incoming-Link Contexts

When compared side-by-side in a Sankey diagram, we can more clearly see both



In order to empirically choose a subset of the ranked gene list produced after the optimisation process, we analysed the distribution of the PPI among the proteins encoded by the ranked genes. Following the evidence that proteins that are near to each other in the PPI network have a similar function Sharan et al. (2007) we expect that if the genes within the top percentiles of the list are composed by proteins functionally related we would find that the proteins that they encode are separated by a short distance in the PPI network. In both the applications of the presented strategy, we found that the portion of protein pairs that are near to each other in the PPI network is the highest in the first percentile and decreases considering the genes included in the subsequent percentiles, Figure 2.

Using two poorly understood gene families, one from human and one from yeast, we have shown how schema analysis can be used to annotate protein families and their individual members. Guilt-by-association and other network-based functional annotation methods (review, Sharan et al. (2007)) are, by intent and design, better suited for the general function prediction problem. However, schema analysis provides a new way to amplify a weak signal, and can suggest mechanistic details in some cases. For example, if we consider proteins that interact physically with a given protein, and then take the most over-represented biological process annotation among them, we find "apoptosis" as a prediction for only two of the MAGE proteins having physical interactions (at corrected p-value ≤0.05 level) and no

represented by a score that measures the probability that two nodes are functionally related. Once the high-confidence weighted network is built, it can be used to predict annotations for uncharacterized GPs, such as GP function or localization Sharan et al. (2007). The prediction of unannotated GPs is performed by means of prediction algorithms which may fall into two different categories: direct annotation schemes exploit neighbours' functions for the annotation of a target GP while module-assisted schemes first cluster the network into modules of related GPs and then annotate each module

Sharan et al. (2007)

title: *Network-based prediction of protein function*

pmcid: 1847944

ILC_Met: <vector>

ILC_Hyp: <vector>

ILC_Res: <vector>

ILC_Con: <vector>

ILC_Bac: <vector>

ILC_Exp: <vector>

ILC_Mod: <vector>

...

Figure 5.6.2: Illustration of per-class indexing of sentences in incoming link contexts.

similarities and differences in the best average weights found for internal and external methods.

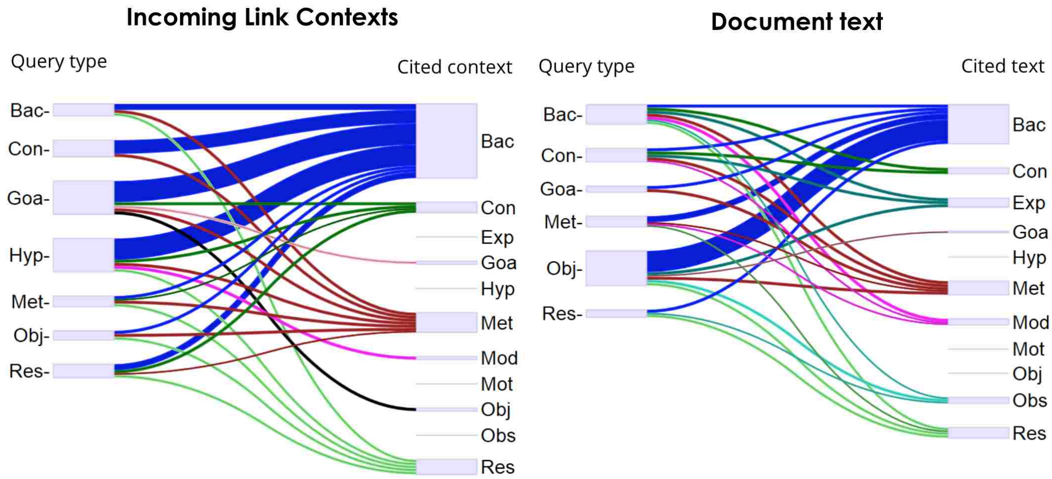


Figure 5.6.4: Sankey representation of the best average weights found during weight training for PMC.

We can examine some of these patterns in detail. First, we can see that Background would appear to be very widely cited. This is also the most common type of label assigned by the classifier, so we do not expect to learn much from these results. The interesting contrast we do see is the lack of consistent weights found for document text for sentences of type Hypothesis. When citing while presenting a research hypothesis, it seems sentences of type Background are not cited often enough.

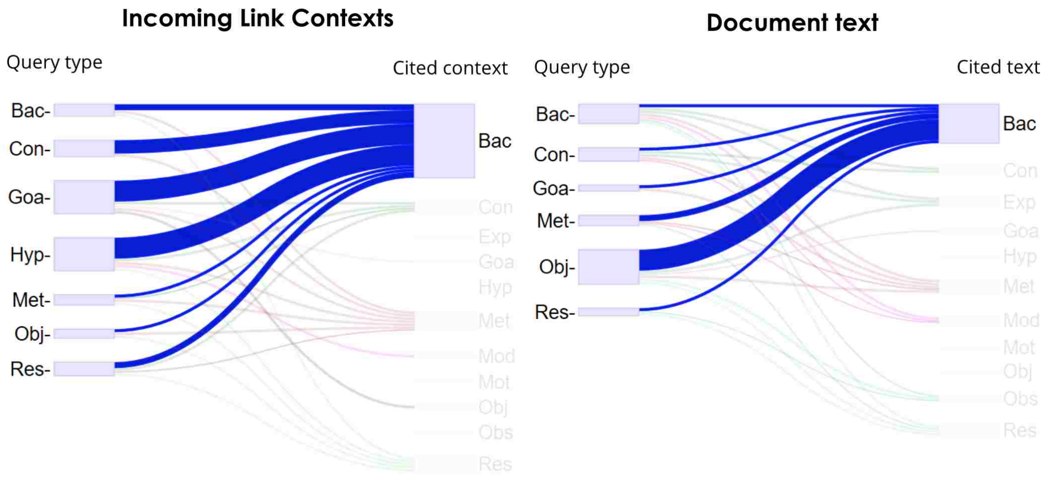


Figure 5.6.5: Sankey representation: best average weights for target sentences of type Background.

A more interesting case are sentences of type Model. These are cited by papers from sentences of type Background, Conclusion and Method. What is remarkable is that when formulating a research Hypothesis, we find a good match for this text in sentences of type Model that appear in incoming link contexts to the document we seek. So, another document cites this one and discusses a Model in the same paragraph that cites it.

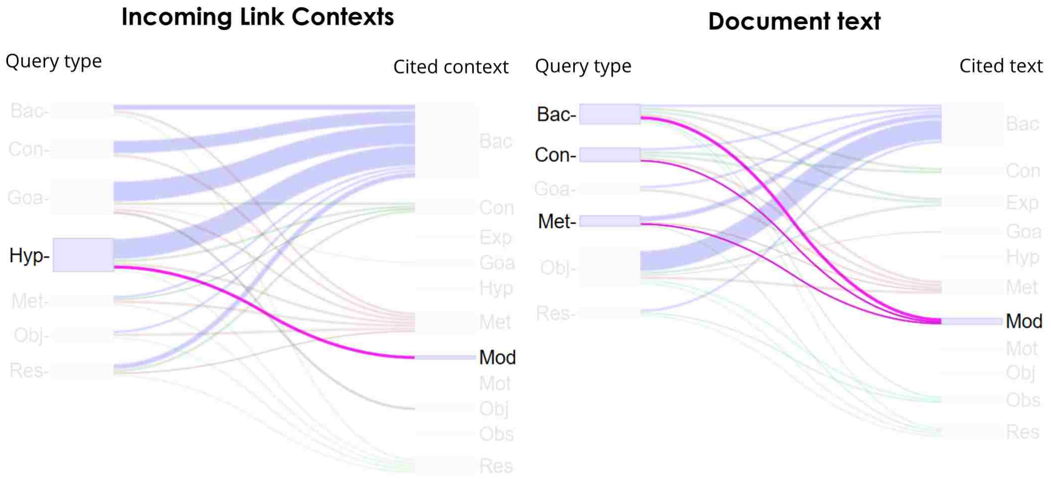


Figure 5.6.6: Sankey representation: best average weights for target sentences of type Model.

Another interesting pattern emerges for sentences of type Conclusion. A paper's

Conclusion is cited in sentences that discuss Background or present the Conclusions of the citing paper. However, we also find good matches for sentences of type Conclusion in the vicinity of a citation fo a paper when our citing sentence type (query type) is Goal, Hypothesis, Methodology or Result.

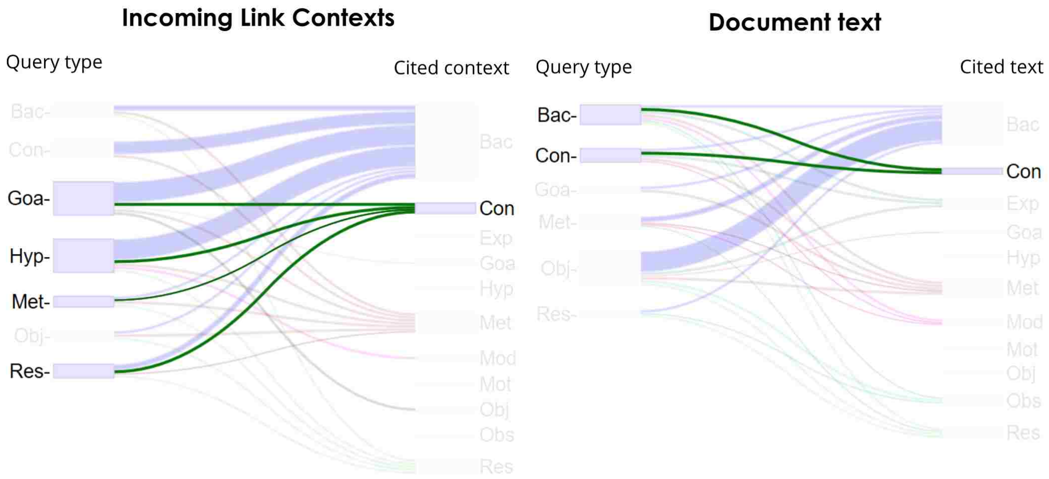


Figure 5.6.7: Sankey representation: best average weights for target sentences of type Conclusion.

Finally, we can see a similar pattern emerge for sentences of type Result, where Bacground, Methodology, Object and Result consistently cite sentences of type Result in the cited documents and also those of type Result in the anchor text of citations to these documents.

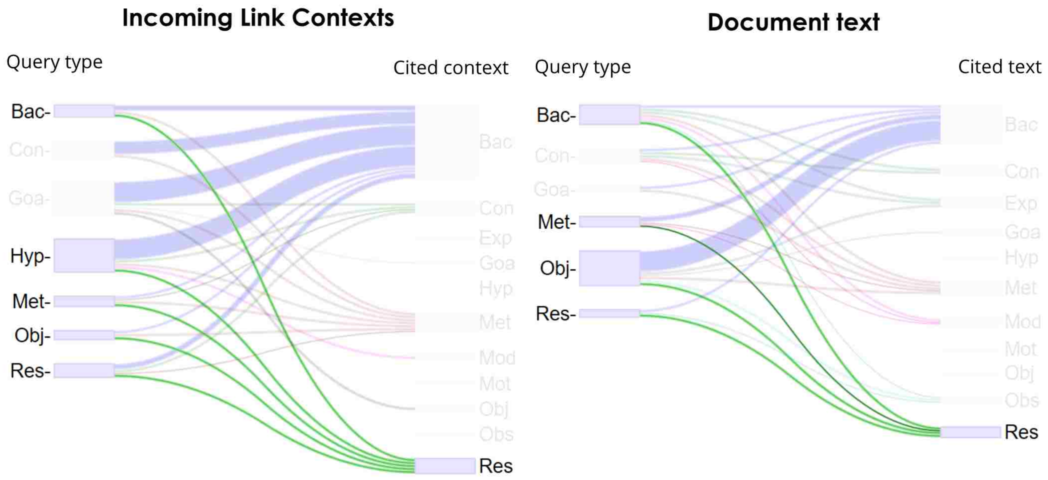
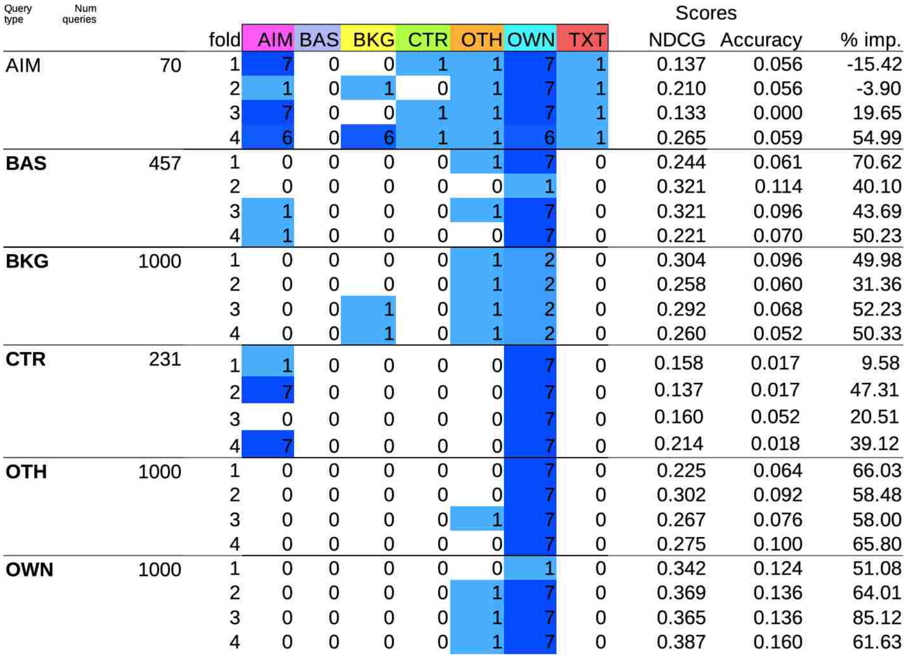


Figure 5.6.8: Sankey representation: best average weights for target sentences of type Result.

5.6.2 Argumentative Zoning and AAC

In comparison with the CoreSC-based results on PMC, we see in Figure 5.6.9 that the results for AZ are much more skewed, with OWN taking the vast majority of the weight assigned.

However, some results are still noteworthy. One is the consistency in finding that zones of type OTH are useful for citations in zones of type BKG. Another one is how apparently important the very few sentences of type AIM can be for citations found in CTR zones. Also, it is noteworthy that as compared to the contents of the document, the sentences around citations to it of type BKG are not as important, and only seem to add something when the citation itself is in a zone of type BKG.



Query type	Num queries									Scores		
		fold	AIM	BAS	BKG	CTR	OTH	OWN	TXT	NDCG	Accuracy	% imp.
AIM	70	1	7	0	0	1	1	7	1	0.137	0.056	-15.42
		2	1	0	1	0	1	7	1	0.210	0.056	-3.90
		3	7	0	0	1	1	7	1	0.133	0.000	19.65
		4	6	0	6	1	1	6	1	0.265	0.059	54.99
BAS	457	1	0	0	0	0	1	7	0	0.244	0.061	70.62
		2	0	0	0	0	0	1	0	0.321	0.114	40.10
		3	1	0	0	0	1	7	0	0.321	0.096	43.69
		4	1	0	0	0	0	7	0	0.221	0.070	50.23
BKG	1000	1	0	0	0	0	1	2	0	0.304	0.096	49.98
		2	0	0	0	0	1	2	0	0.258	0.060	31.36
		3	0	0	1	0	1	2	0	0.292	0.068	52.23
		4	0	0	1	0	1	2	0	0.260	0.052	50.33
CTR	231	1	1	0	0	0	0	7	0	0.158	0.017	9.58
		2	7	0	0	0	0	7	0	0.137	0.017	47.31
		3	0	0	0	0	0	7	0	0.160	0.052	20.51
		4	7	0	0	0	0	7	0	0.214	0.018	39.12
OTH	1000	1	0	0	0	0	0	7	0	0.225	0.064	66.03
		2	0	0	0	0	0	7	0	0.302	0.092	58.48
		3	0	0	0	0	1	7	0	0.267	0.076	58.00
		4	0	0	0	0	0	7	0	0.275	0.100	65.80
OWN	1000	1	0	0	0	0	0	1	0	0.342	0.124	51.08
		2	0	0	0	0	1	7	0	0.369	0.136	64.01
		3	0	0	0	0	1	7	0	0.365	0.136	85.12
		4	0	0	0	0	1	7	0	0.387	0.160	61.63

Figure 5.6.9: Results for AZ-annotated Incoming Link Contexts

5.7 Testing the trained weights

What remains now is to test the weights we found with the queries used in the experiments in Chapter 3 and observe if this leads to an improvement in scores. In the previous sections we have found notable consitency in the sets of weights that maximise the evaluation scores in 4-fold cross-validation. We have presented before the hypothesis that these weights can be exploited to improve on our task, so our aim here is now not to find the most efficient way to take advantage of these findings, but to explore whether:

1. these findings can indeed be exploited to increase the relevance of recommendations.
2. these findings transfer to completely unseen queries in the same domain.

First we must select, for each corpus, for each query type and for each document representation method (internal and external), a single set of weights from those found across folds. Given the notable amount of noise introduced by the automatic classification, we decide to allow for some small variation when selecting the final weights:

- All folds must yield some improvement. If any of the 4 folds leads to a decrease in scores, we discard all weights found for that query type and revert to using a weight of 1 for every field.
- If 3 out of 4 folds find consistent weights and the variation in standard deviation is small, we conclude that the weights we have found are sufficiently consistent and select the weight of the folds which agree.
- If 2 out of 4 folds agree, or two pairings of 2 folds, we take the average.

These criteria lead us to select the final set of weights we will test, which we list in full in Appendix B. We call these sets of weights *trained weights* (TW).

We run the Queries A sets for AAC and PMC using these weights, and we find that they notably underperform versus the baseline. Carefully tuning the query, we are able to improve very slightly on AAC, but not consequentially and not worth reporting. For PMC we never manage to even approach the baseline. In conclusion, although we find enough consistency in the results of our experiments, we are unable to find a way to use this for our original aim of improving recomendations for CCR.

5.8 Conclusion

As we have seen, we were able to find consistent weights across folds to apply to different sentence types for several types of queries (type of citing sentences), for both corpora examined. This confirms our hypothesis that consistent relations exist between types of citing sentences and cited sentences, which holds true for both AZ and CoreSC. This also confirms that explicitly modelling the semantics of scientific discourse is a worthwhile endeavour.

We have not found a way of exploiting these findings to improve on the scores in our evaluation task, which was our original aim, but we see clear applications of these patterns in analysing the spread of scientific knowledge and so in automated ways of measuring the impact of publications.

Beyond this, the future research question remains of how adequate for our needs any of these schemes may be. Perhaps a purpose-designed one would yield better results. Ideally we would want to capture these semantics in a more flexible, softer way, and at the same time learn the semantics from the corpus with less manual annotation required, as this approach is both expensive and brittle. We discuss this in our Future Work section in Chapter 7.

In the next chapter we change track and we take a closer look at the extraction of the contexts of citations we are evaluating against and investigate fine-grained query extraction and weighting.

Chapter 6

Query generation using keyword boosting

> “Artificial Intelligence is a collection of brittle hacks that, under very specific circumstances, mimic the surface appearance of intelligence."
>
> David Mimno

6.1 Introduction

In Chapter 2 we discussed a variety of approaches to CCR when framed as an information retrieval task, together with the key conceptual dimensions which appear to underlie this variation, which we proposed as *aspects* to our research. The main dimensions we identified were:

- **Indexing / document representation**: how do we build the index of documents from which to retrieve our recommendations? What parts of the original papers, if any, should be included, and what other text?
- **Query extraction:** what exactly do we mean by the first “C” in CCR, i.e. what is the “context”, and how do we generate a suitable query from it?
- **Similarity:** what algorithms or functions do we employ to provide the similarity metric between the query and each potential document to recommend?

A review of the relevant literature (Section 2.10) suggests that of these, the aspects that seem to have received the least attention are *context extraction* and *query generation*, or more succinctly, transforming the extracted text into a suitable query. It has often been assumed that a symmetric window over either words or sentences around the citation token should contain all the necessary keywords for retrieving the most relevant papers. However, as we saw in Chapter 3, different types and sizes of windows noticeably affect performance. Further, the generation of the query from this context has been limited to either selecting keywords using simple heuristics or thresholding on tf-idf (or a similar metric). Alternatively query generation has been combined with the similarity metric by using a fully neural-based approach.

The methodology presented in Chapter 5 maintained the approach of extracting symmetric windows of sentences in order to compare to the baseline approaches, but in this chapter we move on to explore whether a more fine-grained approach could yield better results. Specifically, we examine different methods for selecting the keywords that should be included in the query and for determining their relative weights, in order to maximise the relevance of recommendations. We also revisit our approach to query generation for evaluation and investigate what queries would maximise our evaluation score, and then examine supervised machine learning approaches for query extraction.

We can break down our hypothesis H4 we defined in Chapter 1 into these sub-hypotheses:

- H4-1: We can increase the evaluation score of a query by selecting a subset of the available keywords from an insertion context and assigning a specific weight to each of them.

- H4-2: A supervised machine learning algorithm can learn the right selection and weighting of terms using linguistic and morphological features in the citation's immediate context. These features include Part-Of-Speech tags and dependency arcs, as well as the type of sentence according to the discourse annotation schemes we explored in Chapters 4 and 5.

- H4-3: The annotated data for training this algorithm can be automatically generated, using our existing implicitly annotated data and the indexed document collections.

In the following, we set out to test each of these.

6.2 Related work: reference scope resolution

There is a sizeable body of relevant work that deals with automatically determining which spans of text in a citing document contain information about a given cited document. This has received various names, among which *reference scope resolution* and *citation context identification*.

One example of this is Qazvinian and Radev (2010), who aimed to identify sentences in the proximity of a citation that contain information about a specific secondary source but do not explicitly cite it. They refer to these non-explicit citing sentences as *context sentences*, which we adopt for this short review[1]. They manually annotated a corpus of 203 citations with sentences relevant to each citation within a 4-sentence window (2 up, 2 down). Three features inform their approach: 1. whether the sentence is an explicit citing sentence, 2. whether it contains a number of hand-defined bigrams (e.g. “this work”, “their approach”) and 3. the lexical similarity between the sentence and the target document. They use a Markov Random Field, which in their experiments outperforms a discourse-based baseline, a similarity-based baseline and an SVM. Jha et al. (2017) would later employ the exact same approach to identifying context sentences, but their application is in the field of scientometrics.

Our task is very different from theirs: they either aim to measure the impact of a publication or to extract extra background information about a paper. In either case, the citation is already provided and the task is only to determine which spans of text in the citing document are relevant to the cited one. This allows them to derive features for classification that are based on the similarity between the citing paper and cited paper, which is conceptually similar to the automatic annotation we carry out in Section 6.6 but very different from the posterior extraction of keywords and assignment of weights.

Kaplan et al. (2009) had previously presented an approach to identifying these context sentences based on following chains of coreference in the text. They manually annotated a small corpus (50 citations) with sentences relevant to each citation and trained a coreference resolver using an SVM classifier, for which the task is to determine whether a given anaphor matches a candidate antecedent. They employ features such as agreement in number and gender, and features of the anaphor: whether it is a pronoun, indefinite or demonstrative, etc. Kim and Webber (2006) carried out a similar study where they examined the coreference of the pronoun *they* in astronomy papers and also annotated a small corpus and also trained a supervised classifier.

[1] Note that this differs from the meaning of *context* and *context sentence* outside of this section.

Abu-Jbara and Radev (2011) also implemented a supervised machine learning method for reference scope identification, but their aim is multi document summarisation. Conceptually, their approach is much closer to our own experiments, as they use an SVM supervised model that uses as features:

- the similarity of a sentence to the target paper using tf-idf-weighted cosine distance
- the type of section the sentence is in (out of 10 canonical section headlines)
- the relative position of the sentence in the section and in the paragraph
- whether first person pronouns appear
- tense of the first verb
- relative position of the first determiner present in the sentence (*this*, *that*, *those*, etc.)

In work partially inspired by this, Athar and Teufel (2012) undertake the manual annotation of a corpus consisting of 20 ACL papers and over 1,700 citation contexts to these papers and train an SVM classifier to detect context sentences using a variety of binary features. These are based on the sentence containing, for example: the name of the primary author; the date of publication; an acronym that is also used in an explicit citation; a determiner followed by a work noun (e.g. “this technique”); a third person pronoun in initial position (e.g. “they”, “their”); a different citation than the one under review, etc. This approach yields a best micro-averaged F-score of 0.89 when using a window of 4 sentences up, 4 down, showing once again (as we saw in Chapter 4) the importance of these domain-specific manually defined patterns.

While of conceptual relevance to our own work, identifying the context of an existing citation differs significantly from our CCR task. First, these approaches rely heavily on the coherence and cohesion of text, as well as specific patterns of scientific text. This assumes that the text was woven around the citation, which is necessarily true in existing publications, but is an unreasonable assumption in a real-life CCR application. Second, keywords that will allow us to retrieve a given paper do not necessarily need to be in sentences that clearly refer to that one citation, which leads us to investigate keyword extraction, which we will explore later in this chapter.

However, before we look at keyword extraction, we take advantage of their annotation efforts in this area for our own experiments below. Before delving into fully testing H1 above for our task, we first carry out an experiment where we evaluate reference scope resolution as presented above in order to gauge the potential gains from reducing the context window from which the query should be extracted.

6.3 The relevance of sentences in a citation's context

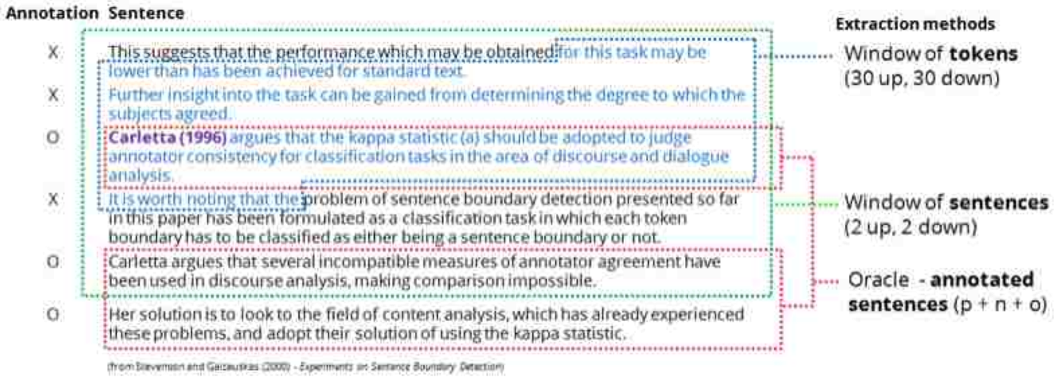
As discussed in Chapter 3, previous approaches to context extraction fall into two main groups: windowing approaches and sentence selection approaches. Windowing approaches use either a window of tokens, where the context is considered to be *n* tokens before the citation token and *n* tokens after it, or a window of sentences, where the citing sentence is included, plus *n* sentences before and/or after it.

As we presented in Chapter 2, these have been common approaches: He et al. (2010) used a symmetric window of words (50 before, 50 after) as did Liu et al. (2014) (300 before, 300 after). He et al. (2012) used passages (splitting the article into half-overlapping fixed-size windows of words). Huang et al. (2015a) used a window of sentences: citing sentence + 1 before + 1 after.

It is unsurprising that always using a fixed window size and not dealing with linguistic structure will add noise to the query, leading to false positives and false negatives in the extracted keywords. Instead of dealing with this noise exclusively by using weighting schemes based on topic modelling or word embeddings (e.g. Huang et al. (2015a)), we propose that those approaches could also benefit from an extra processing step in the pipeline where the potential keywords in the context are explicitly selected or weighted for their relevance. This follows the suggestion of Ritchie et al. (2006b) that NLP methods would be superior for selecting terms to index from a citation's context as "the amount of text that refers to a citation can vary greatly, multiple citations in close proximity can interact with each other and affect 'ownership' of surrounding terms".

6.3.1 Manual annotation

As first step in this direction we evaluate using a human oracle, namely the corpus of Athar and Teufel (2012) to select sentences that contain text from the context of a citation that is relevant to the cited document. In this corpus, 20 papers were selected from the ACL Anthology, and 1,741 citation contexts to these papers were manually annotated by a single annotator. Within a window of 4 sentences before the citing sentences and 4 after (4 up, 4 down), each sentence receives two annotations: a) whether it is relevant to the citation and b) its sentiment. The sentiment can be one of: *p* - positive, *n* - negative and *o* - objective/neutral. A visual representation of these different context extraction methods is presented in Figure 6.3.1.



Annotation	Sentence
X	This suggests that the performance which may be obtained for this task may be lower than has been achieved for standard text.
X	Further insight into the task can be gained from determining the degree to which the subjects agreed.
O	**Carletta (1996)** argues that the kappa statistic (a) should be adopted to judge annotator consistency for classification tasks in the area of discourse and dialogue analysis.
X	It is worth noting that the problem of sentence boundary detection presented so far in this paper has been formulated as a classification task in which each token boundary has to be classified as either being a sentence boundary or not.
O	Carletta argues that several incompatible measures of annotator agreement have been used in discourse analysis, making comparison impossible.
O	Her solution is to look to the field of content analysis, which has already experienced these problems, and adopt their solution of using the kappa statistic.

Extraction methods
Window of **tokens** (30 up, 30 down)
Window of **sentences** (2 up, 2 down)
Oracle - **annotated sentences** (p + n + o)
(from Stevenson and Gaizauskas (2000) - *Experiments on Sentence Boundary Detection*)

Figure 6.3.1: A visualisation of corpus annotation. In red, the human oracle: the sentences the annotator chose as relevant to the citation. In green, a window over sentences (2 up, 2 down) and in blue a window over tokens (30 up, 30 down).

6.3.2 Experiments and results

In what follows, we report the results as originally published in Duma et al. (2016c). We only index the full text of the document (i.e. only use one internal method) and the following methods for extracting a citation's context are compared:

- **window**: a window of *n* tokens, the same number before and after the citation token.
- **sentence**: a window of sentences.
  - 1only: only the citing sentence.
  - [n]up [m]down: n sentences before (up) and m after the citing sentence (down). This window always includes the citing sentence.
  - paragraph: the full paragraph where the citation appears.
- **annotated sentence:** sentences that were human-annotated as being relevant to the citation. Each sentence was annotated with its sentiment polarity: (p)ositive, (n)egative or (o)bjective. For example, as labeled below, *annotated_sentence_po* is the set of sentences selected by the annotator as relevant that had (p)ositive or (o)bjective polarity, excluding the ones that were labeled as (n)egative.

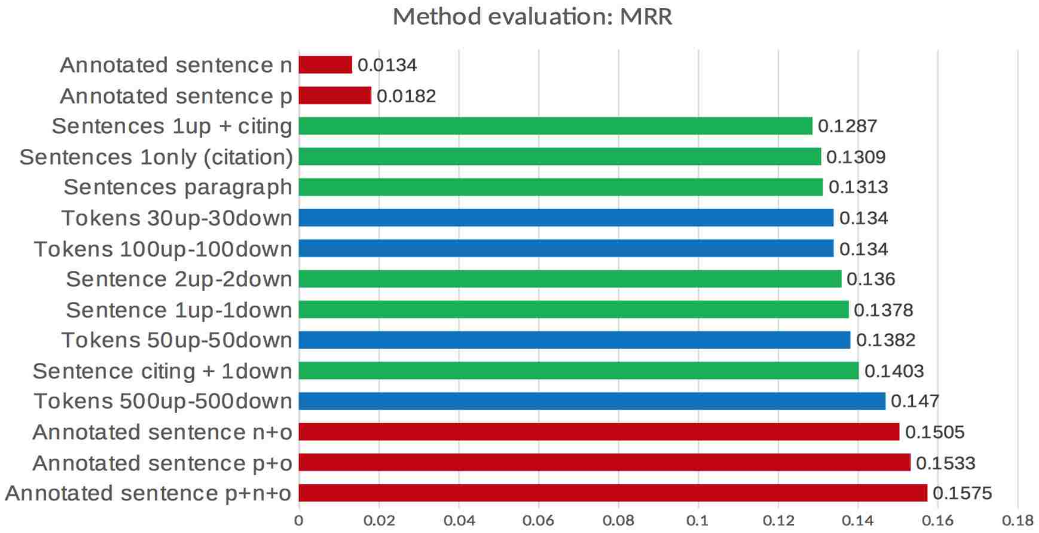


Figure 6.3.2: Experiment results. Manually selecting sentences within a 5-sentence context is superior to symmetric methods, irrespective of sentiment annotation. In red, the human oracle. In green, window of sentence methods. In blue, window over tokens methods.

The results are shown in Figure 6.3.2. They indicate that forming the context out of sentences that were manually annotated to be relevant to the citation leads to generating superior queries than using any symmetric method. Interestingly, selecting sentences based on their annotated sentiment polarity produces worse results, leading us to deduce that sentiment classification, at least as present in this corpus, is not a useful feature for this task.

In related work, Kang and Kim (2012) manually annotated a corpus of 56 research articles with "citation units", which according to their definition can be of five types: phrase, clause, sentence, multi-sentence, and "other" (which refers to figures, tables or other non-textual content). While they did not attempt any automated methods to identify these spans, their analysis of the annotated data is informative: a vast majority of annotated citation units are sentences (about 75%) and 5.2% are multi-sentence, but "phrases" account for 10.6% and "clauses" for 7.8%. This suggests that there are potential gains in investigating sub-sentence resolution of text spans. Our results in this preliminary exploration, together with those from Chapter 3, encourage us to explore this further.

6.4 Related work: keyphrase extraction

Perhaps the area of research most conceptually relevant to our task is *keyphrase extraction*. In a publication that is foundational to the field, (Turney, 1999) describes this as "the automatic selection of important and topical phrases from the body of a document". In essence, approaches to keyphrase extraction have largely focussed on extracting keywords that describe the topics that a document deals with (Hasan and Ng, 2014). Some examples of keyphrases would be “machine learning” or “statistical machine translation”.

Many of the most popular approaches to keyphrase extraction are unsupervised, of which perhaps the best known is TextRank (Mihalcea and Tarau, 2004). TextRank builds a graph of the document based on co-occurences of words within a sliding window, and applies some filtering based on POS tags. The potential keyphrases are then ranked using an algorithm derived from PageRank (Page et al., 1999). While we do not follow the approach of TextRank and instead opt for a supervised approach, we do employ it as part of our examination of the contribution of phrases to the query in Section 6.7.

It is however the fully supervised approaches that we take inspiration from. Prime early examples of this are GenEx (Turney, 1999) and KEA (Frank et al., 1999). These systems use two features that can be classified as *statistical features* (Hasan and Ng, 2014): tf-idf and the normalised position of first occurrence from the beginning of the document. Later approaches have used *structural features* such as whether the phrase appears in the title or abstract of a scientific publication. Nguyen and Kan (2007) explore this, as well as syntactic and morphological features: sequences of POS tags and sequences of suffixes. Using domain-specific knowledge, they also employ heuristics for labelling words as acronyms.

While our task is different, we draw inspiration from these previous approaches and adapt these previously used features to our supervised prediction task. These early approaches typically operate in two stages. The first stage collects keyphrase candidates, typically using a number of heuristic filters such as POS tags, NP chunks, n-grams (Hulth, 2003) or morphological characteristics of the words themselves, such as suffixes or labelling whether the word is an acronym using domain-specific heuristics. The second stage treats the task as binary classification: given a candidate keyphrase, should it be extracted? We discuss the potential formulation of our task as classification or regression in Sections 6.6 and 6.10.

Some approaches have used information external to the document collection, such as the frequency of a phrase as a link on Wikipedia (Medelyan et al., 2009). Similarly, KEA uses the frequency of a keyphrase being manually tagged by authors as a feature, but we have no equivalent human annotation to draw on. We believe using these external sources of annotation would be highly applicable to our task, but in the experiments reported here we avoid using any external resources.

A more general case to *keyphrase extraction* would be *keyphrase generation*, where the keyphrases assigned to a given document do not need to appear in it (Turney, 1999). This can be seen as a more ideal formulation for our task, which would help bridge the lexical gap between citation context and cited document. While we do not tackle this here, future work on CCR might benefit from this more general formulation.

The machine learning methods employed over the years for keyword extraction have varied, including naïve Bayes (KEA), decision trees (GenEx), maximum entropy (Kim and Kan, 2009), multi-layer perceptrons and support vector machines (Lopez and Romary, 2010). Promising recent approaches to keyphrase generation use deep neural networks and specifically the popular sequence-to-sequence encoder-decoder architecture (Meng et al., 2017). Recent work leads us to believe sequential neural models are well suited to keyphrase generation, but with our experiments here we aim to establish a baseline for keyword extraction, for which we employ more traditional approaches and per-token regressors.

6.5 Context-based keyword extraction

We want to maximise the performance of a system using a standard keyword-based retrieval backend by focussing on improving the extraction of the query. Here we focus on testing our hypothesis H4-2, which we can formulate more briefly as: can we extract a better query by using machine learning models which employ linguistic, surface and corpus statistics features?

For this, we propose to frame query extraction as a supervised machine learning problem. The main novelty is that we annotate the relative importance (boost) of a term in a query based on its scoring by the retrieval backend, relative to other top scoring documents. We then train machine learning models that use textual and linguistic features to recover this ideal boost value. We name the two distinct steps to our approach *keyword annotation* and *keyword extraction*.

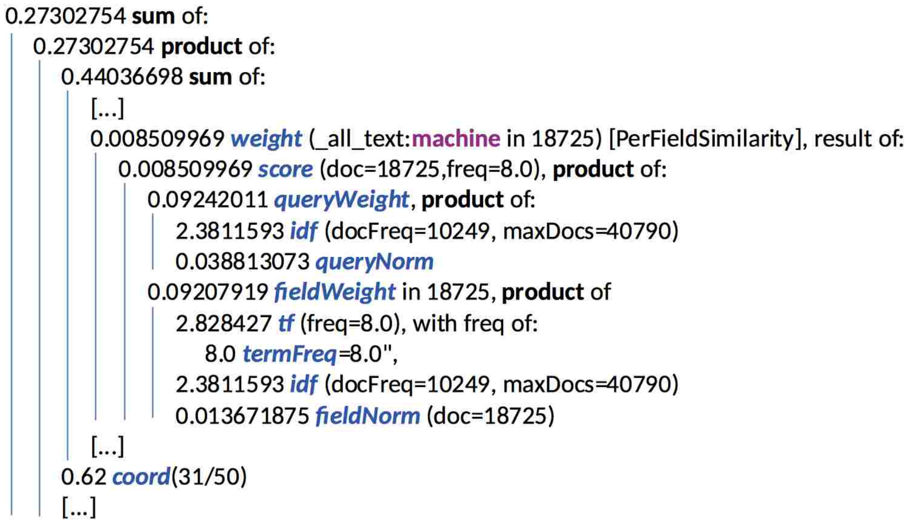
1. **Keyword annotation**: In order to train and evaluate our approach, first we need

to annotate the ground truth: given a context and a resolvable citation, we need to annotate the optimal query, which would maximise the score for this citation placeholder. In our approach, a query consists of keywords together with weights for each, so we want to annotate which terms should be extracted, and what weight each of them should receive. The optimisation objective is to maximise the joint score of all originally cited documents for that context.

2. **Keyword extraction**: Once we have annotated the contexts of resolvable citations with the keywords that should be extracted and their weights, we use this as our ground truth to train and test supervised machine learning models. We explore and evaluate different approaches to keyword extraction.

6.6 Keyword annotation

Modern information retrieval engines like Elasticsearch provide a convenient way of obtaining scores for how important a term is to the final similarity score of a document and a query. This is known in IR as an *explanation* and provides insight into the relevance of each query term to a given indexed document. Elasticsearch provides an API for retrieving detailed explanations, including the successive operations that were executed in reaching the final score for each document matching the query. We can see an example of an explanation in Figure 6.6.1.



```
0.27302754 sum of:
  0.27302754 product of:
    0.44036698 sum of:
      [...]
      0.008509969 weight (_all_text:machine in 18725) [PerFieldSimilarity], result of:
        0.008509969 score (doc=18725,freq=8.0), product of:
          0.09242011 queryWeight, product of:
            2.3811593 idf (docFreq=10249, maxDocs=40790)
            0.038813073 queryNorm
          0.09207919 fieldWeight in 18725, product of
            2.828427 tf (freq=8.0), with freq of:
              8.0 termFreq=8.0",
            2.3811593 idf (docFreq=10249, maxDocs=40790)
            0.013671875 fieldNorm (doc=18725)
      [...]
    0.62 coord(31/50)
    [...]
```

Figure 6.6.1: Example of an elasticsearch explanation, here rendered in plain text following the Lucene format, but which is delivered as a JSON-serialised version.

In this example we can see in detail all the factors that went into giving the term "machine" its ultimate score for a given document. We can use these explanation scores to inform our choice of which keyphrases we should extract in order to maximise the evaluation score, and what weight score we should set for each. This weight parameter as employed by the Lucene retrieval engine is known as the *boost.*

Two intuitive ways to frame the per-term annotation task are:

- *keyword selection*: a binary annotation task, where each term in the context is assigned a label *{True, False}* which defines whether it should be extracted as part of the query in order to maximise the evaluation score for the query
- *keyword boosting*: each term in the context is assigned a boost to be used in the query

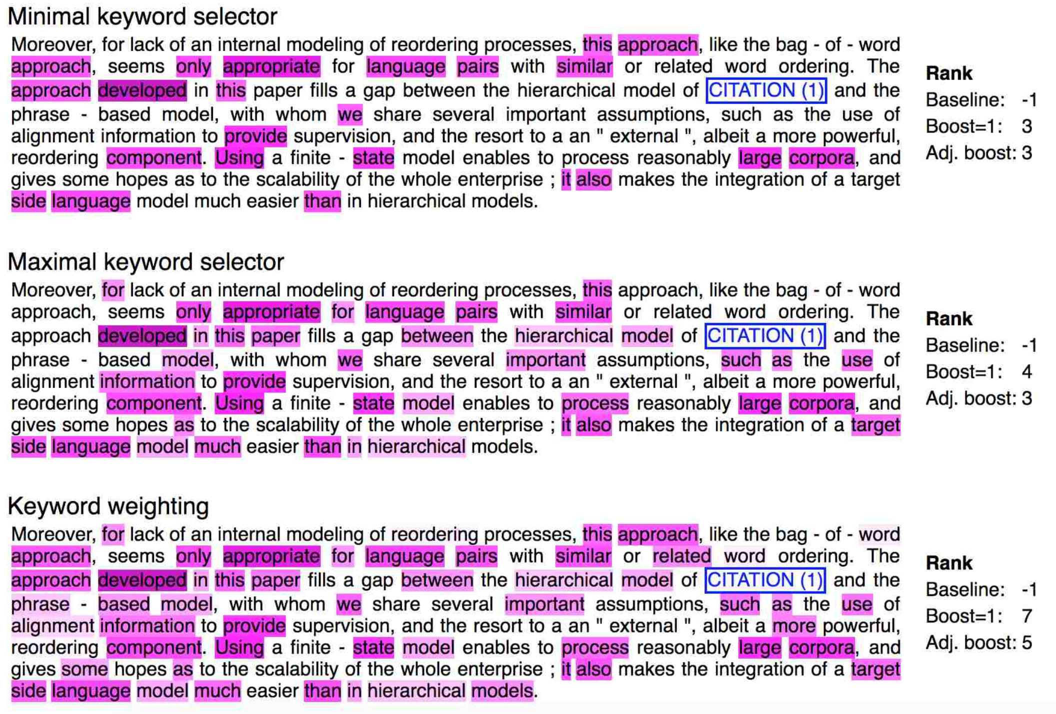
Regarding keyword selection, one option would be to select the *minimal set* of keyphrases that yield the highest score in trying to resolve a citation. This, however, leads to unintended effects due to *artefacts* of the corpus. That is, the resulting keyword set can contain terms which are not in any sense important content words for the document in question but nevertheless contribute to the maximal score. Authors are often primed by the language used in the specific paper they cite (or that is employed in the citation contexts to the same paper, in other articles they have read), and so the context around the citation will often contain terms that appear in the target paper or in other citations to it. Although we have not systematically studied this phenomenon, often these artefacts are words or expressions with discourse functions in a specific context (e.g. "thus" or "surprisingly"). See Figure 6.6.2 for examples.

This presents a problem for our annotation, as simply selecting the words that only appear in the target document (in the majority situation where a single paper is cited per context) will immediately maximise the score for that document but be of little value in generalising the choice of those keywords.

6.6.1 Keyword selection

We have tested different annotation approaches for solving this. One of them is to instead select a *maximal* set of keywords that will still maximise the joint score of all target documents. Although an improvement, this method still suffers from artefacts. Another potential improvement is to limit our training set to the contexts where more than one paper is cited. This again provides an improvement but does not solve the problem. Our solution to these issues is to focus on *keyword boosting*: annotating every matching term with its adjusted boost value. This does not in itself remove the artefacts but it minimises their impact.

In Figure 6.6.2 below we show the keywords that are selected for extraction in bold, with their background colour showing their relative weight (the darker the colour, the higher the weight), for three different keyword annotation methods. We can already see that selecting the keywords to extract leads to better performance than just removing stopwords and setting the appropriate boost to each term. However, we are satisfied to aim for a lower upper bound but to generate annotated data from which patterns are easier to extract by a supervised machine learning model.



Minimal keyword selector

Moreover, for lack of an internal modeling of reordering processes, this approach, like the bag - of - word approach, seems only appropriate for language pairs with similar or related word ordering. The approach developed in this paper fills a gap between the hierarchical model of CITATION (1) and the phrase - based model, with whom we share several important assumptions, such as the use of alignment information to provide supervision, and the resort to a an " external ", albeit a more powerful, reordering component. Using a finite - state model enables to process reasonably large corpora, and gives some hopes as to the scalability of the whole enterprise ; it also makes the integration of a target side language model much easier than in hierarchical models.

Rank
Baseline: -1
Boost=1: 3
Adj. boost: 3

Maximal keyword selector

Moreover, for lack of an internal modeling of reordering processes, this approach, like the bag - of - word approach, seems only appropriate for language pairs with similar or related word ordering. The approach developed in this paper fills a gap between the hierarchical model of CITATION (1) and the phrase - based model, with whom we share several important assumptions, such as the use of alignment information to provide supervision, and the resort to a an " external ", albeit a more powerful, reordering component. Using a finite - state model enables to process reasonably large corpora, and gives some hopes as to the scalability of the whole enterprise ; it also makes the integration of a target side language model much easier than in hierarchical models.

Rank
Baseline: -1
Boost=1: 4
Adj. boost: 3

Keyword weighting

Moreover, for lack of an internal modeling of reordering processes, this approach, like the bag - of - word approach, seems only appropriate for language pairs with similar or related word ordering. The approach developed in this paper fills a gap between the hierarchical model of CITATION (1) and the phrase - based model, with whom we share several important assumptions, such as the use of alignment information to provide supervision, and the resort to a an " external ", albeit a more powerful, reordering component. Using a finite - state model enables to process reasonably large corpora, and gives some hopes as to the scalability of the whole enterprise ; it also makes the integration of a target side language model much easier than in hierarchical models.

Rank
Baseline: -1
Boost=1: 7
Adj. boost: 5

Figure 6.6.2: Example of a context with its annotated keywords to extract, for each annotation method, where the darker the colour, the higher the weight (white represents boost = 0, dark purple represents boost = 1) On the right hand side, each method's evaluation performance, measured as the average rank at which the matched documents appear. *Boost=1* is using a query where each term receives a boost score of 1, and *Adj. boost* is employing the annotated adjusted boost for the query. The value of -1 for rank represents the document did not appear within the top 200 results.

Given the differences between our evaluation task (to retrieve previously used citations in published documents) and our ultimate aim, which is to find ways of improving contextual literature recommendation given a citation placeholder in draft document, we need to strive to avoid these artefacts. We want to find terms that both maximise the score of the papers originally cited in the citation token and that contain keywords that are not artefacts of the particular paper cited. That is, the aim is a set of keywords with a high chance of matching other papers other than just the target documents.

When setting the boost for every term in a query, these weights are in their effect relative to the sum total. We have experimentally attested this by multiplying and dividing, for a set of queries, all term boosts by a constant and obtaining identical evaluation scores. This means that scores can be normalised, which is a useful processing step when training a classification or regression machine learning model.

6.6.2 Keyword boosting

For all experiments in this chapter, we fix our context extraction to be *1up_1down*, that is, one sentence before the sentence containing the citation, this sentence itself, and one sentence after. For document representation, we use *ilc_full_text_1_paragraph*, which means we index a bag of words formed of the full text of the document concatenated with the text of each paragraph in our collection that contains a citation to the document.

Our method for annotating the boost for each term is:

1. Retrieve the *explanation* for each of the top 200 results for the query. Extract the term scores from each explanation.

2. Retrieve the *explanation* for each of the target documents in the citation token.

3. Normalise the score of every term that matches one of the originally cited documents by dividing by the sum total of the scores of that term over all 200 documents.

We have tested a number of schemes for normalising the weights, including several tf-idf inspired variants, and here we report results using the best performing one. We adjust the score of each term that matches a *match document* by dividing it by the one plus the squared sum of the scores of this term for all top 200 matches. We use the formula below to adjust the term weight for each match document. In the following definition, $boost_{t,m}$ is the boost score for term t for original document m; $matchscore(m,t)$ is the score for term t and document m as retrieved from the IR explanation; $docscore(d,t)$ is the score for term t in document d which belongs to the set of top 200 retrieved documents D.

	AAC		PMC	
System	**Boost = 1**	**Adjusted boost**	**Boost = 1**	**Adjusted boost**
Minimal keyword selector	0.904	0.873	0.583	0.552
Maximal keyword selector	0.720	0.879	0.498	0.595
All keywords	0.716	0.808	0.498	0.516

Table 6.6.1: NDCG upper bounds for each keyword annotation method.

$$boost_{t,m} = \frac{matchscore(m,t)}{1 + (\sum_{d \in D} docscore(d,t))^2} \tag{6.6.1}$$

This approach gives us an approximation of the boost to assign to each term that matches between the query and the target documents in order to maximise the evaluation score of the query. When applying this method to adjusting the boost scores and annotating our testing contexts with it, the upper bound for evaluation (making the query of all annotated terms with their corresponding boost) is shown in Table 6.6.1.

For AAC, it is clear that selecting the minimal set of keywords that maximise the match documents' score and setting their weight simply to 1 is the highest performing method. However, this method yields the fewest training examples, which in our experiments were not sufficient to train an extractor with good performance, so as we said in the previous section, we choose to annotate all matching terms for the context as terms to extract and to adjust their boost scores using this normalisation approach. The results for PMC paint a slightly different picture: if the weight is kept constant, then the minimal set selector is superior, but when using the boost scores, selecting the maximal set of keywords still obtains a higher score.

6.7 Keywords or keyphrases?

As we saw earlier, approaches to keyphrase extraction share these two characteristics: they focus on extracting keyphrases rather than keywords, and they deal with extracting these that are relevant to the document as a whole rather than a specific passage inside it.

However, so far we have represented both a citation's context and a document as bags-of-words, and as a result, we have only considered individual keywords rather than keyphrases. Are we losing accuracy in retrieval by ignoring phrases? To find out, we first need to define a way of identifying and extracting keyphrases in our evaluation

Method	Keyphrases
1. TextRank	(’aforementioned model’, 0.0282), (’standard bag-of-words base model’, 0.0282), (’latent concept retrieval model’, 0.0282), (’latent concept model base’, 0.0282), (’mixed model’, 0.0282), (’dual-space semantic model’, 0.0282), (’dual space model’, 0.0282), (’esa model’, 0.0282), (’bag-of-words model’, 0.0282), (’model’, 0.0282), (’dual-space re-ranking model’, 0.0282), (’mixture model lda’, 0.0282), (’mixture model’, 0.0282) [...]
2. Frequency	(’latent model’, 19), (’explicit model’, 16), (’dual-space model’, 11), (’initial ranker’, 10), (’semantic space’, 8), (’information retrieval’, 6), (’test collections’, 6), (’space model’, 6), (’statistically significant’, 6), (’multinomial distribution’, 5), (’explicit semantic’, 5), (’external knowledge’, 5), (’initial retrieval’, 5), (’mixed model’, 5), (’document retrieval’, 5), (’vector space’, 4), (’latent concept’, 4), (’latent semantic’, 4) [...]
3. Intersection	[...] (’vector space model’), (’cosine similarity’), (’re-ranking setting’), (’latent semantic indexing’), (’explicit semantic analysis’), (’re-ranking score’), (’document retrieval’), (’initial retrieval’), (’generalpurpose retrieval’), (’information retrieval’), (’semantic relatedness’), (’document re-ranking’), (’dual-space re-ranking model’), (’final ranking score’), (’initial ranker’), (’ranking process’), (’re-ranking problem’), (’retrieval performance’) [...]

Figure 6.7.1: The top row shows a sample of keyphrases generated by TextRank. Row two shows the top 18 keyphrases according to n-gram frequency. Finally, the bottom row illustrates the intersection of the sets derived from TextRank and n-gram frequency. The intersection of sets helps filter out spurious keyphrases.

contexts.

A first step in identifying potential keyphrases in the context to a citation is to first identify them for the source document as a whole. For this, we can employ the algorithms mentioned in Section 6.4. Specifically, for our experiments we employ the well known TextRank (Mihalcea and Tarau, 2004) algorithm, with some modifications. From observation of the output of the algorithm, it was evident that the generated keyphrases were not of the quality required, and included many spurious keyphrases, such as “aforementioned model” and standard “bag-of-words base model” in the example we show in Figure 6.7.1.

Several approaches to filtering keyphrases exist, notably filtering by part-of-speech patterns, such as ADJ AJD NOUN (matching “Gaussian random variable”) or ADJ NOUN NOUN (“Statistical machine translation”) (Justeson and Katz, 1995). Others have used the structure of a document, e.g. extracting n-grams from the title and abstract (HaCohen-Kerner, 2003).

Here we opt for a lightly structure-aware frequency-based approach: we introduce restrictions on keyphrase length and impose a threshold on the minimum number of occurrences of the potential keyphrase in the text. Specifically, we select the top 200 keyphrases identified by the algorithm, and we filter each potential phrase by these conditions:

- It is either a bigram (two terms) or a trigram (three terms) *AND*
- It appears in the title or in the abstract *OR*
- It occurs at least twice in the body text of the article

This allows us to filter out most of the spurious keyphrases normally found by TextRank, as we can see in Figure 6.7.1. While boosting precision, this approach at the same time necessarily decreases recall in the extracted keyphrases. This is however not an important concern given that our task here is simply to determine whether using keyphrases as part of retrieval would be beneficial, and so, worth pursuing.

We obtain the full set of potential keyphrases for the citing document and we annotate the keyphrases from that set that are present in resolvable citation's context. We evaluate the following keyphrase testing methods:

Baseline: full set of matching keywords between query and match documents, using a single *match_query* for each term and no keyphrases.

Keyphrases_add_sum: we add an optional *phrase* query to the elasticsearch DSL query. If this phrase matches, its score is added to the overall score for the document. The boost of the keyphrase is set to the sum of the boost values of the individual terms. This is the boost that we obtain using the *all keywords* method defined in Section 6.6.2.

Keyphrases_add_avg: same as above, but the boost of the keyphrase is set to the average of the boost values of the individual terms.

Keyphrases_sub_sum: we substitute individual *match_query* terms with a *match_phrase* term. We take into account the term frequency counts: for example, given a keyphrase "w1 w2", if $tf(w1) = 3$ and $tf(w2) = 1$, the resulting query will contain one phrase query for "w1 w2" with boost of 1, a single match query for w1 with boost of 2 instead of 3, and no match query for w2, as it will have been "used up" in the phrase query.

Keyphrases_sub_avg: same as above, but like before, the boost of the keyphrase is set to the average of the boost values of the individual terms.

Context	To overcome the NP-hard problems that derive from the need to consider all possible permutations of the **source sentence**, we make here a radical simplification and consider training the **translation model** given a fixed segmentation and reordering. This idea is not new, and is one of the grounding principle of n-gram-based approaches **[CITATION]** in SMT. The novelty here is that we will use this assumption to recast **machine translation** (MT) in the familiar terms of a sequence labelling task.
Keyphrases	['machine translation', 'translation model', 'source sentence']
Baseline	{"bool": { "should": [... {"term": { "_all_text": { "value": "machine", "boost": 0.024711478337076334 } }, }, {"term": { "_all_text": { "value": "translation", "boost": 0.019476058881485417 } } }, ...] } }
Keyphrases_add_sum	{"bool": { "should": [... { "term": { "_all_text": { "value": "machine", "boost": 0.024711478337076334 } }, }, { "term": { "_all_text": { "value": "translation", "boost": 0.019476058881485417 } } }, { "match_phrase": { "_all_text": { "query": "machine translation", "boost": 0.03444950777781904 } } }, ...] } }

Figure 6.7.2: Example of a context containing identified keyphrases and highlights from the elasticsearch DSL query for two keyphrase testing methods.

For all of these experiments, we are using the set of 1000 queries from the Chapter 3 experiments, and removing the same minimal set of stopwords. In these experiments, we are only generating queries made of the terms that we have already established match one of the original documents cited, which explains the very high scores. In essence, all terms not matching the *match document* have already been filtered out.

The Lucene/elasticsearch query language allows for very complex queries to be formed, but it is not our aim to explore them all here. So, in principle it remains possible that keyphrase matching could actually be exploited for higher scores using a different structure of query. However, when restricting ourselves to simple *match*, *term* and *phrase* queries, and treating all of them as optional elements (that is, logically linked with OR prefixes), Table 6.7.1 shows that using keyphrases does not lead to improved results over simply assigning the right weight to each individual query term.

We conclude then that our focus in what follows should be to find ways of automatically determining the ideal boost to set to each term in the query.

	AAC		PMC	
	Term boost = 1	**Adjusted boost**	**Term boost = 1**	**Adjusted boost**
No keyphrases	0.680	0.760	0.498	0.516
Keyphrases_add_sum	0.654	0.736	0.493	0.510
Keyphrases_add_avg	0.654	0.753	0.493	0.514
Keyphrases_sub_sum	0.655	0.721	0.487	0.505
Keyphrases_sub_avg	0.655	0.733	0.487	0.508

Table 6.7.1: Results from experiments integrating keyphrases into the evaluation queries.

6.8 Stopwords

One aspect that became unexpectedly important for the research carried out in this chapter was the selection of a set of stopwords to be removed from the query. Intuitively one would expect the weighting scheme to meaningfully reduce their contribution to the point where they would be insignificant, but as we shall see, this is not the case.

We define *stopwords* as function words that add no semantic value to a query. Some examples of stopwords are *and*, *if*, *is* and *would*: all of these are very frequent function words in English and occur in most if not all documents in the collection. Stopword removal is a widely used technique in both information retrieval and text categorisation (Silva and Ribeiro, 2003) and a common approach is to define a stopword as a term that has a similar likelihood of occurring in documents not relevant to the query as in those that are (Wilbur and Sirotkin, 1992).

We adopt a mixed approach enriched with a set of heuristics, counting as stopwords those terms that:

1. Occur in at least 30% of the documents in the collection. If a term cannot be used to discriminate between a very important portion of the document collection, it should not be considered for inclusion in a query.

2. Are not nouns or adjectives. Purely because of their frequency in the collection, adjectives like “higher” and “large” and nouns like “levels” and “group” could be classed as stopwords. However, to the author of a paper it is evident that they can still be important signals as part of a query. In practice, any POS tagger

introduces an error rate, so for each term, we use its most common POS tag as classified in our evaluation queries as its true tag. The same word form can serve different syntactic purposes, so this is indeed a simplifying assumption, but one that works to our benefit given the noise in the data we are dealing with.

3. Are under three characters in length. This filters out not just function words such as “of” and “in”, but is important particularly for our AAC corpus, given the notable noise added by the PDF to XML conversion (badly converted formulas, tables, markup, etc.)

The boolean combination of these criteria is *not_noun AND not_adj AND (document_ratio >= 0.3 OR term_length < 3).* We label the stopwords selected in this way the *collection-specific stopwords*. We compare removing these to removing the same set of *basic stopwords* used in the experiments in Chapter 3 and to removing *no stopwords* at all: the results are reported in Table 6.8.1. We are particularly interested in the effect of stopword removal once we assign the adjusted boost scores to each query term.

6.8.1 Basic stopwords vs collection-specific stopwords

	AAC		**PMC**	
	Term boost = 1	**Adjusted boost**	**Term boost = 1**	**Adjusted boost**
No stopwords	0.380	-	0.366	-
Basic stopwords	0.405	0.761	0.368	0.516
Collection-specific stopwords	0.404	0.783	0.357	0.535

Table 6.8.1: Average NDCG scores for the test queries for each domain when employing different stopword lists.

The conclusions are consistent for both domains: stopword removal does increase average scores, and the collection-specific stopword list does perform slightly worse, which signals that some information is being lost. This effect is strongest for PMC, in spite of the fact that the stopwords list is shorter than for AAC. However, once we assign the adjusted boost scores to each term, we obtain the best overall results. This effect is much more noticeable for AAC, where we almost double the NDCG score.

It is a common assumption that *tf-idf* will on its own sufficiently dampen the importance of stopwords to a query. However, from our experiments it emerges that tuning the default dampening introduced by the *idf* term is a promising approach to improving relevance.

In order to establish the statistical significance of these results, we run four separate T-Tests, the results of which can be seen in Table 6.8.2. All differences are statistically significant ($p < .001$), except that between using the basic stopword list and the collection-specific stopword list for AAC when keeping the *term boost* fixed to 1. The differences become significant when applying the adjusted boost, although the adjusted Cohen's *d* effect size falls within the range considered "small" as defined by Sawilowsky (2009).

		t	df	p	Cohen's d
Term boost = 1	Basic stopwords - AAC stopwords	0.923	999	0.356	0.029
	Basic stopwords - PMC stopwords	3.526	999	< .001	0.111
Adjusted boost	Basic stopwords - AAC stopwords	-7.444	999	< .001	-0.235
	Basic stopwords - PMC stopwords	-6.499	999	< .001	-0.206

Table 6.8.2: Results of a paired samples T-Test in order to establish statistical significance of differences in stopword results.

6.8.2 Contribution of stopwords to query score

To analyse the importance, individually and as a whole, of the stopwords that we have identified, we measure their raw and adjusted match scores. We report three measures:

- **Score %:** the ratio that each term contributes to the total score of all terms over the first 200 documents returned by the query.
- **Match %:** same as above, but only taking into account the *explanations* for the originally cited documents.
- **Adjusted boost %:** same as above, once we adjust the ratio that the stopwords contribute to the weights we select for that query.

We see that for AAC, the total mass of stopwords weight in the scores is very high at 29%, where for PMC it is lower at around 16%. This is partly due to the corpus being much larger, and so fewer stopwords being chosen using our current frequency-based method. For the AAC queries, adjusting the term boost reduces the overall ratio of stopword weight, but for PMC it unexpectedly increases it. This is not of significance to the posterior evaluation, given that we remove all these stopwords from every query.

In conclusion, stopwords matter. The methodology for their selection and removal makes a big impact in evaluation scores.

	AAC				PMC		
Term	**Score %**	**Match %**	**Boost %**	**Term**	**Score %**	**Match %**	**Boost %**
in	5.45	4.07	0.99	in	5.61	2.71	0.69
is	2.9	2.17	0.78	is	1.34	0.7	0.58
for	2.85	2.16	0.93	for	1.19	0.6	0.5
we	2	1.54	0.68	that	0.83	0.33	0.37
as	1.54	1.18	0.83	are	0.61	0.41	0.46
that	1.32	0.99	0.67	on	0.57	0.24	0.35
on	1.14	0.88	0.7	as	0.56	0.28	0.38
are	1.02	0.79	0.58	this	0.5	0.33	0.53
this	0.91	0.73	0.67	was	0.42	0.19	0.3
based	0.68	0.56	0.57	be	0.4	0.21	0.37
be	0.56	0.41	0.44	have	0.31	0.14	0.31
which	0.56	0.41	0.58	were	0.28	0.14	0.17
using	0.51	0.39	0.55	also	0.27	0.17	0.48
used	0.44	0.35	0.48	at	0.26	0.18	0.35
our	0.44	0.33	0.36	been	0.23	0.11	0.3
can	0.34	0.26	0.37	has	0.23	0.16	0.4
have	0.34	0.26	0.42	can	0.2	0.14	0.3
has	0.33	0.25	0.43	it	0.18	0.11	0.29
use	0.31	0.24	0.44	these	0.17	0.08	0.21
it	0.31	0.22	0.3	based	0.16	0.06	0.09
been	0.29	0.21	0.38	used	0.13	0.08	0.2
such	0.27	0.22	0.4	using	0.11	0.06	0.15
these	0.26	0.2	0.35	more	0.11	0.07	0.17
set	0.24	0.19	0.23	found	0.09	0.03	0.1
each	0.23	0.2	0.31	during	0.08	0.06	0.17
TOP 25	**25.24**	**19.21**	**13.44**	**TOP 25**	**14.84**	**7.59**	**8.22**
TOTAL PCT	**29.63**	**22.54**	**20.35**	**TOTAL PCT**	**16.36**	**8.45**	**11.18**

Table 6.8.3: Percentage of scores contributed by the top 25 terms that we have identified as a *collection specific* stopword, excluding the *basic stopwords* we were already filtering and those terms consisting of only one character. For the full list, see Appendix C.

6.9 Manual annotation

In some previous studies (Turney (1999); Nguyen and Kan (2007); Hulth (2003)), human annotators were asked to select the keywords relevant to finding articles for a given context. Building on the approach we followed in Section 6.3, we decide to follow this to evaluate our automatic approaches to a human oracle and so undertake manual annotation of a set of evaluation contexts. The task for the human annotator is defined in summary as “if, using a search engine (e.g. Google Scholar), you are trying to find the original paper cited where **[CITATION HERE]** appears, which words from the text would you select?”.

6.9.1 Inter-Annotator Agreement

Two annotators manually annotated 100 contexts for each domain, selected for diversity, from among those used to generate evaluation queries for the previous experiments. The first annotator (Annotator A), the author of this thesis, is a domain expert in computational linguistics (AAC corpus) but not in biomedical science (PMC), while the second annotator (Annotator B) is a researcher in efficient GPU computation, but not familiar with any of the domains (AAC or PMC). Annotation instructions were developed and iterated over to align the biases and priors of both annotators, and these are included in Appendix D.

Following standard practice, we measure inter-annotator agreement using Precision, Recall and F-score. We take Annotator A as the gold standard for our experiments and we report the scores of Annotator B's as the measure of agreement.

For each annotator, we automatically filter both the general and corpus-specific stopwords we have identified before generating the evaluation queries, and we count each selected keyword once, which keeps the task as a more realistic and useful overlap of selected keywords.

	AAC	**PMC**
Precision	0.591	0.854
Recall	0.787	0.659
F1 score	0.615	0.710

Table 6.9.1: Inter-annotator agreement for both corpora.

From the results in Table 6.9.1, it is immediatley apparent that the annotators differ in their choice of keywords to a notable degree, and that this degree is domain-dependent. The resulting F1 score for PMC is notably higher than that for AAC, which perhaps shows an influence of Annotator A introducing their own bias as a domain expert. When looking at precision and recall scores, we see that, while precision is remarkably higher for PMC (0.854), recall is much lower than AAC (0.659 vs 0.787).

The relatively low inter-annotator agreemen is hardly surprising, as predicting what keywords are likely to occur in an unknown paper does not have an immediate intuitive solution for a human annotator. This shows how widely an author's criteria for selecting relevant keywords may vary.

6.9.2 Quantitative evaluation

We compare the results of creating evaluation queries from the manually annotated keywords of Annotator A with those generated using a simple baseline where we remove both the basic stopwords and the collection-specific stopwords identified above.

With the aim of measuring how external representations affect this task, we run these queries against two indices built with two different document representation methods:

- **Internal:** the full text of the document
- **Mixed:** full text of the document concatenated with all of the text from incoming link contexts to the document (full text + incoming link contexts paragraph).

		AAC			PMC		
		Ann. A	**Ann. B**	**Baseline**	**Ann. A**	**Ann. B**	**Baseline**
	Terms selected	1,589	2,008	4,214	3,329	2,609	5,277
	Unique terms selected	1,033	1,299	3,488	2,220	1,676	4,301
	Percent terms selected	22.17%	28.23%	58.79%	39.60%	30.78%	62.28%
Mixed	**Top-1 accuracy**	0.157	0.187	0.230	0.350	0.260	0.330
	NDCG	0.385	0.394	**0.440**	**0.540**	0.481	0.534
Internal	**Top-1 accuracy**	0.092	0.082	0.069	0.270	0.200	0.250
	NDCG	**0.260**	0.240	**0.260**	**0.459**	0.404	0.445

Table 6.9.2: Results of human annotation (Annotator A, Annotator B) versus the baseline. Average NDCG and top-1 accuracy, as well as percentage of terms selected, on the small corpus of 100 human-annotated contexts for each domain.

The results are remarkable in two ways: first, a simple baseline outperforms Annotator A in their expert domain (AAC), and second, Annotator A matches and very slightly improves over the baseline in the other domain (PMC). At the same time, Annotator B is consistently outperformed by either Annotator A or the baseline (for AAC) or both (for PMC). This suggests that an intuition for selecting the best performing keywords may depend on the degree of expertise in a domain, but also on having an understanding of the inner workings of keyword-based retrieval systems.

These results throw light on our prior findings in Section 6.3 and Chapter 3. On the one hand, they support the hypothesis that careful selection of the relevant spans of text to a citation can perform better than a simple baseline. On the other, this result only holds for both domains when the text we are indexing is the full text of the paper. For AAC, when we use external methods for building a document representation and therefore include the anchor text of links to that document, there is value in extracting keywords that the best performing human annotator does not judge to be directly relevant to the citation token we are concerned with.

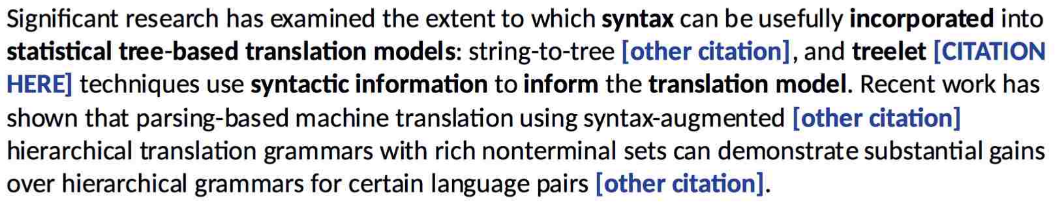
As we saw in Chapter 3, external representations are far superior to internal ones for the AAC corpus, but the opposite is true for PMC. The results of manual annotation show that this is because of similarities in citing patterns in the AAC corpus, which lends support to the popular approach of employing topic models for CCR.

6.9.3 Qualitative analysis of annotation

The manual annotation enables qualitative analysis of the data, from which important patterns emerge for each corpus.

In spite of extensive cleaning, there is still text conversion noise in the AAC data, such as broken sentences, extraneous material (table captions and footnotes inside running paragraph text), plus hyphenated word wrap is not resolved in the corpus. This is not exclusive to our corpus; converting PDF documents to machine-readable flowing text is always challenging (Constantin et al., 2013). Manual examination of examples from the CiteSeerX collection confirms the same situation. We avoided manually annotating the examples where the text was judged too noisy.

As we saw in Chapter 4, there exists extensive work in classifying citations in a research paper. According to one of the aspects of the classification scheme of Swales (1986), anecdotal evidence suggests that citations in AAC are much more often perfunctory, i.e. "mainly an acknowledgement that some other work in the same general area has been performed" rather than organic "cited work is needed for the understanding of the citing paper". Figure 6.9.1 shows us an example of this: often, the very few words preceding the citation token (e.g. "treelet"), together with the most salient keyphrases preceding it (e.g. "tree-based translation models") is an ideal query to be extracted.



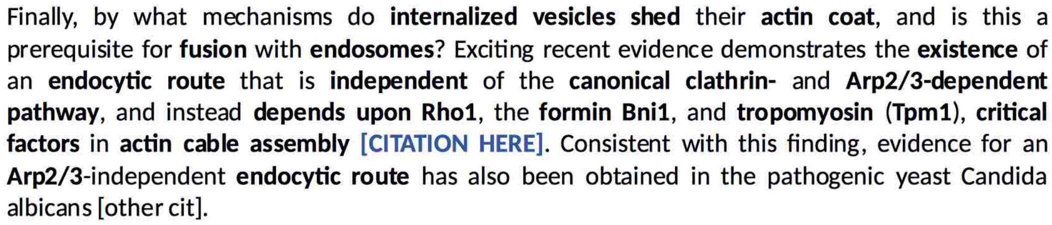
Significant research has examined the extent to which **syntax** can be usefully **incorporated** into **statistical tree-based translation models**: string-to-tree [other citation], and **treelet** [CITATION HERE] techniques use **syntactic information** to **inform** the **translation model**. Recent work has shown that parsing-based machine translation using syntax-augmented [other citation] hierarchical translation grammars with rich nonterminal sets can demonstrate substantial gains over hierarchical grammars for certain language pairs [other citation].

Figure 6.9.1: Example of a manually annotated AAC context. In bold, the keywords selected for extraction by the annotator.

This contrasts with PMC, where often the text around the citation token contains very specific details of what the document contains, e.g. "the C-terminal ADFH domain in twinfilin binds F-actin in addition to G-actin and can interact with the barbed end of the filament [CITATION HERE]". As we said above, this is consistent with the results of Chapter 3, where we saw that for the PMC corpus, indexing a document's contents yields vastly superior results to the incoming link contexts.

There is a strong connection here also with our experiments in Chapter 5, where we proposed that if we were able to classify the different types of citations in the document, this could inform the extraction of a better query. We make a first approach in our experiments in Section 6.10 by adding the labels of the AZ and CoreSC type of the citing sentence as features.

Another finding is that the heuristics we employed for sentence splitting and tokenisation seem a reasonable fit for AAC text, but PMC text would benefit from applying techniques tailored to the domain. The example in Figure 6.9.2 offers us a good intuition of how the domains differ. The span of text “Arp2/3-dependent”, marked for extraction by the human annotator, is split by the standard Elasticsearch tokenizer into the three tokens “Arp2 3 dependent”. This removes the information that this was a single token in the original text, and moreover, our heuristic of disregarding terms of length less than three removes the token “3”. In this instance the query retains the information about “Arp2”, that is, “Actin-Related Protein 2”, which is one of seven subunits of the “Arp2/3 complex” that the text is referring to (Pollard, 2007). Whether this actually affects the relevance of the results or not, it is clear that domain specific preprocessing techniques would be preferred to treating all text the same.



Finally, by what mechanisms do **internalized vesicles shed** their **actin coat**, and is this a prerequisite for **fusion** with **endosomes**? Exciting recent evidence demonstrates the **existence** of an **endocytic route** that is **independent** of the **canonical clathrin-** and **Arp2/3-dependent pathway**, and instead **depends upon Rho1**, the **formin Bni1**, and **tropomyosin (Tpm1)**, **critical factors** in **actin cable assembly** **[CITATION HERE]**. Consistent with this finding, evidence for an **Arp2/3**-independent **endocytic route** has also been obtained in the pathogenic yeast Candida albicans [other cit].

Figure 6.9.2: Example of a manually annotated PMC context.

Natural Language Processing of biomedical text (BioNLP) is a highly active field of research that deals with the idiosyncrasies of this domain. Beyond tokenisers custom to the domain in the early stages of the processing pipeline, Named Entity Recognition (NER) is one of the most commonly researched tasks in the automatic processing of biomedical scientific articles (Goulart et al., 2011). The recognised entities are commonly disambiguated and linked to large databases of genes, proteins, chemical compounds, etc. We discuss this more in Future Work, Section 7.3.1.

It is clear then that at least in the case of the biomedical domain, applying these NLP techniques to preprocess the text to be indexed as well as the text from which to extract the query could greatly help the task, with linking entities in the text and substituting their mentions with unique identifiers as a necessary next step. While this is hardly a surprise, it remains an important consideration for future approaches.

6.10 Keyword extraction

Now that we have generated our annotated data of input contexts and ideal output queries, we can turn to exploring the performance of different algorithms for extracting these queries.

In considering the goal of training a supervised machine learning model for query extraction, we need to make a number of choices. First, we can treat the task as *classification* (deciding to extract a term in the context or not) or *regression* (predicting the weight that this token should have in the query). Second, we can either train *per-token classifiers* that operate independently on each token in the input context, or employ models that use the whole context as input and make a joint decision on the terms to include in the query and their relative weights.

Further, we could separate the task into several stages, for example first identifying promising spans of text to extract, then selecting keywords, and then assigning a weight to each selected keyword. When considering a sequential input mapped to a sequential output, we could think of this task as similar to a number of established tasks, such as:

- *Machine translation (MT)*: Map from a source language to a target language. The input context would here be the source language and the target language, the terms to extract. There is one important difference in the data here: while in both source and target language in MT there is structure to the sequence of tokens, this would not be true in this case for the target side. The keywords to be extracted would not conform to any language model.

- *Part-of-speech (POS) tagging / Named Entity Recognition (NER)*: These are both sequence labelling tasks: for each input token, generate an output label. In our case, those labels are not parts of speech or whether the token is the beginning/middle/end of a recognised entity, or the type of this recognised entity, but whether a token should be extracted or not. For our boost setting task, we define a set of labels that represent different bins of boost to assign to the given token.

These formulations are interesting avenues for future work. In what follows, however, we focus on establishing a baseline using a simpler formulation where extracting and boosting decisions are made on a per-token basis.

6.10.1 Experiments

Our hypothesis H4-2 is that there is enough information in the combination of the context of a citation token, the shape of the words and in statistics about the words in the collection, to reason how important each context word should be to the extracted query. To test this, we train machine learning models based on these features, using the annotated context we generated as gold standard training data.

In these experiments, we make an effort to remove the most obvious artefacts, so we remove author names from the text using a gazetteer built from the references of each paper. At the same time, in order to minimise the noise introduced by PDF conversion, particularly in the conversion of formulas, we increase the minimum length required for a term to be considered as part of a query to three characters, up from two in Section 6.8. This is mainly due to the amount of artefacts in AAC introduced by PDF conversion. Because of this, the scores reported below are necessarily slightly different from those reported in this chapter so far. All other parameters are kept equal, including the indexing, document representation, basic stopwords, etc.

We test and compare these systems here:

- **Baseline 1:** The full citation context, once cleaned of basic stopwords (including all words of under three characters in length, and the most obvious cues such as author names). We want to avoid any possible artefacts in this close examination of the source text.

- **Baseline 2**: Builds on Baseline 1 by further removing the collection-specific stopwords we found in Section 6.8.

- **Linear Regression:** A basic linear regression model.

- **Extra Trees Regressor:** A standard supervised machine learning model[2], employing an ensemble of decision trees. We employ 10 estimators.

- **Multi Layer Perceptron:** A feed-forward neural network with shape (300, 200, 100), trained with the Adam optimiser, using ReLU activation functions, adaptive learning rate and an optimisation tolerance of *1e-5*.

The features for the supervised models are listed in Table 6.10.1. We avoid using a unique per-term identifier as a feature, as we are neither interested in overfitting to our

[2] We use the scikit-learn library version 0.19.2 (http://scikit-learn.org/)

training data, nor in finding whether particular terms tend to be more important in our document collection. Instead, we aim to use features that would generalise better and employ the same set for both domains, in order to see their relative contribution.

Feature	Description	Type
word_len	Word length in characters.	String
dist_cit	Distance to the citation placeholder, in tokens. Negative if before, positive if after.	Float
abs_dist_cit	Absolute distance to the citation placeholder.	Float
tf	Term Frequency in the context.	Integer
tf_idf	TF-IDF score as computed by Lucene from the index.	Float
caps_num	Number of characters that are uppercase.	Integer
caps_ratio	Percentage of the word that consists of uppercase letters.	Float
az	AZ class, for AAC. One of ["AIM", "BAS", "BKG", "CTR", "OTH", "OWN", "TXT"]	String
csc_type	CoreSC type, for PMC. One of ["Hyp", "Mot", "Bac", "Goa", "Obj", "Met", "Exp", "Mod", "Obs", "Res", "Con"]	String
suffix_2, suffix_3, suffix_4	The word's suffix, if it is one of the top 50 most common suffixes of the given length. Only added to words that are at least 2 characters longer than the suffix length.	String
pos	Part of Speech tag, as annotated with spaCy.	String
dep_	Incoming dependency arc as annotated using spaCy.	String

Table 6.10.1: Features employed for the per-token regressors. All the continuous (float) features are normalised for each citation context and all string features are binarised, i.e. they become one feature for each possible value the feature can take.

For the linguistic features we use spaCy.[3] The POS tags use the OntoNotes 5 tagset.[4] and the dependency arcs use the ClearLabels scheme. [5]

We generate our training data by automatically annotating it with the ideal per-term boost scores. Our training set is formed of 10,000 annotated contexts for PMC and 4,836 for AAC. The test data for both of them are the 1,000 annotated contexts corresponding to the same evaluation queries used in Chapter 3. We employ extrinsic evaluation: we measure the actual performance of the generated queries on the held-out test set of queries employed in the experiments in Chapter 3.

We combine the ML approaches with our baselines by applying stopword removal to the output of the regressor. That is, no matter what the predicted weight may be for the term, if it is in the list of stopwords it is assigned a weight of 0.

[3] Using spaCy version 2.0.2 (https://spacy.io/usage/) and the large English model *en_core_web_lg-2.0.0* (https://spacy.io/models/en#en_core_web_lg)

[4] https://catalog.ldc.upenn.edu/LDC2013T19

[5] https://github.com/clir/clearnlp-guidelines/blob/master/md/specifications/dependency_labels.md

6.10.2 Results

The results of the evaluation we can see in Table 6.10.2 tell a clear story about methods for the AAC corpus and a more nuanced one for PMC. We can see that Baseline 2 outperforms Baseline 1 on both corpora, so removing collection-specific stopwords clearly has a useful effect. Yet this is where the similarities end. For AAC, we can see that Linear Regression outperforms this simple baseline, and in turn an Extra Trees Regressor and finally the Multi Layer Perceptron continue to increase the scores. For PMC however, the top performing method remains Baseline 2, and none of the trained models improves on it. This is a clear indication of the differences between the two domains, where the relevant portions of context to extract can in the case of PMC be by default the whole paragraph.

	AAC		**PMC**	
System	**NDCG**	**Top-1**	**NDCG**	**Top-1**
Baseline 1	0.366	0.160	0.347	0.210
Baseline 2	0.373	0.167	**0.349**	0.214
Linear Regression	0.383	0.175	0.340	0.195
Extra Trees Regressor	0.391	0.187	0.317	0.178
Multi Layer Perceptron	**0.407**	0.191	0.344	0.204

Table 6.10.2: Results of the query extraction systems tested. We can see all regression methods outperforming the baseline for AAC, with the MLP being superior. For PMC, the highest scoring method cannot even match the baseline.

The training of Random Forest Regressors enables us to recover the relative importance of each feature employed, which we find very informative. Figures 6.10.1 and 6.10.2 show plots of the importance of the top 40 features for AAC and PMC respectively. Simple heuristics like the length of the term in characters, the normalised distance to the citation token and the term frequency in the context are shown to be important features, as well as the TF-IDF score as computed from the Lucene index. From these graphs we can also see that while the per-sentence AZ and CoreSC tags seem to make some contributions to the regressor, they are nowhere as important as the more simple features.

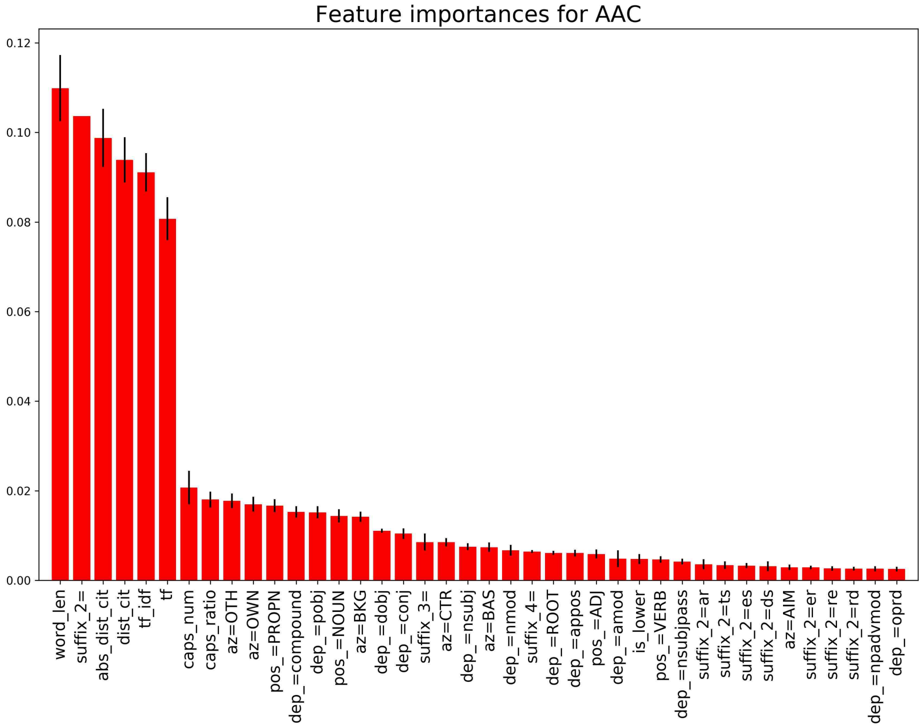


Figure 6.10.1: Top 25 features and their importance for the AAC training set.

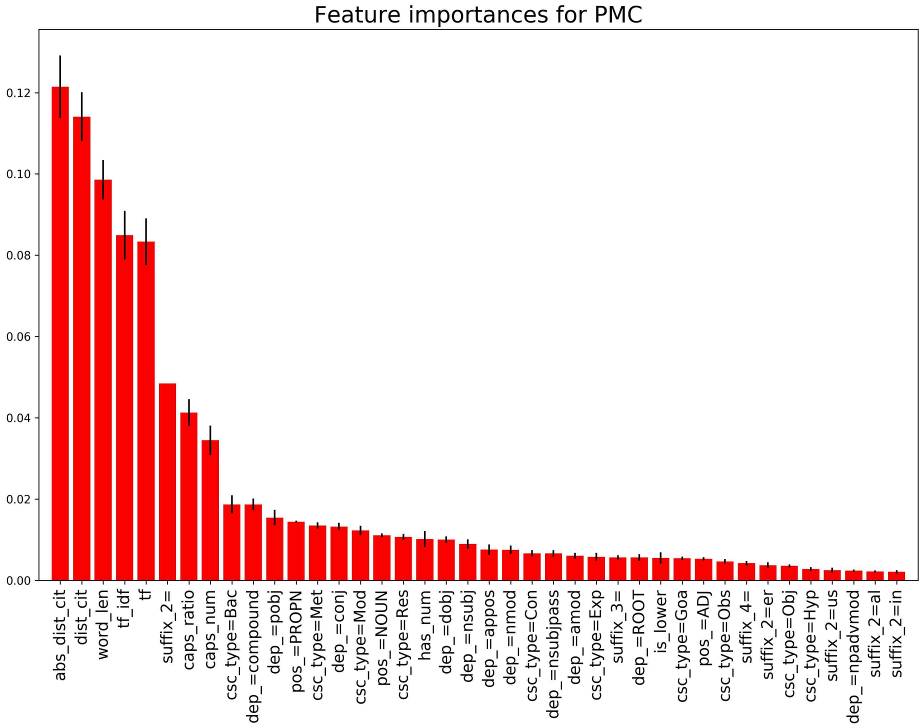


Figure 6.10.2: Top 25 features and their importance for the PMC training set.

We have seen that a relatively simple approach based on surface features can meaningfully improve on our baselines in these experiments for the domain of AAC. Conversely, we see the opposite for PMC, where a simple baseline is superior to any of our trained models.

This shows that different domains will benefit from different techniques, such as those applied in BioNLP, as we mentioned in Section 6.9.3 and we will further discuss in the next chapter in Future Work.

6.11 Future work

There are plenty of areas we could improve on in our approach to keyword boosting. One example is the technique we use to set the boost values in Section 6.6: it is far from optimal. We could set the boost values in a way that truly maximised the evaluation score. This data, annotated with a stronger relevance signal, may help improve the results of the supervised machine learning models.

Another area is stopwords: as we mentioned above, the selection of stopwords could be further tuned to the domain, and even made relative to the context.

Domain-specific approaches could be of help. The work we reviewed in Section 6.2 which employed manually defined cue phrases for citation scope detection could offer inspiration for this.

However, we believe the most important improvements to this task can come from better machine learning models for this task, which take into account the whole context of a citation instead of each individual token. We are particularly optimistic about recurrent or convolutional deep neural networks that represent terms as word embeddings, which we propose in Future Work in Section 7.3.3.

6.12 Conclusion

In this chapter, we have explored whether a more fine-grained approach to query extraction yields better results, and the answer is yes. We have compared the performance of a human annotator with a simple baseline, and shown the the annotator beats (PMC) or performs comparably (AAC) to the baseline when the index contains only the document text, but not so when the text from incoming link contexts is added.

Let us revisit the hypotheses we set out to explore in this chapter:

- H4-1: We can increase the evaluation score of a query by selecting a subset of the available keywords from an insertion context and assigning a specific weight to each of them.
- H4-2: A supervised machine learning algorithm can learn the right selection and weighting of terms using linguistic and morphological features in the citation's immediate context.
- H4-3: The annotated data for training this algorithm can be automatically generated, using our existing implicitly annotated data and the indexed document collections.

How did we fare? We examined different methods for annotating the keywords that should be included in the query and their relative weights, in order to maximise the relevance of recommendations. These do lead to improvements in scores, which confirms our hypotheses H4-1 and H4-3. We then showed that linguistic and surface features, including scientific discourse annotation features (AZ and CoreSC) can inform a model to improve on the baseline approach for one of the two domains, AAC. This means that our H4-2 is domain-dependent.

A common theme throughout this thesis reappears: treating different scientific domains as the same textual domain is suboptimal. AAC and PMC, in spite of being both corpora of 'scientific articles', are for our practical considerations textual domains with important differences, as we can see in the different evaluation results in the many experiments so far. It is clear that every domain will benefit from a different set of approaches to text processing and query extraction, and a one-size-fits-all approach is bound to be overall less performant. This is an important learning that should inform any attempt to build a production CCR system.

Chapter 7

Conclusion and future work

> “Coming back to where you started is not the same as never leaving."
>
> Terry Pratchett

> “Success is not final, failure is not fatal: it is the courage to continue that counts."
>
> Winston Churchill

7.1 Summary

In this thesis we have explored a formulation of Contextual Citation Recommendation as an Information Retrieval task. Our research was from the beginning motivated by practical aims: to contribute to the development of productisable, usable systems that would be helpful to researchers and academics. Ultimately the outcome of this thesis is not a deployed CCR system as we originally envisioned, but a deeper understanding of the issues to consider.

A parallel motivation to our research was to explore the usefulness of scientific discourse annotation schemes like Argumentative Zoning and Core Scientific Contexts to this task, as a shallow approach to semantics and a first step in the direction of offering contextually relevant literature suggestions that attend to the rhetorical function of the citing text. We envisioned that being able to compare methods or results with previous approaches would be helpful to a researcher, and combining these schemes with an information retrieval approach would be a useful first approach.

Our findings here are mixed. On the one hand, we have found consistent relations between types of citing sentences and cited sentences, using both schemes and in both domains, which confirms that explicitly modelling the semantics of scientific discourse is a worthwhile endeavour. On the other hand, we have not found a way of exploiting these findings to improve on the scores in our evaluation task, which was our original aim. We do however believe this proves their potential for use in other applications, such as scientometrics or multi-document summarisation.

From the beginning, we set out to explore two different scientific domains, one we were familiar with (computational linguistics), and an unfamiliar one which receives plenty of attention from the NLP community (biomedical science), expecting to find differences between them. In this we have been successful, and we have demonstrated that the best performing document representation methods and citing patterns are domain specific. Not only this, we have also shown this difference in the effectiveness of a trained classifier at recovering the importance of a term to a citation. While our initial hypothesis that this was feasible proved correct for computational linguistics, it proved wrong in biomedical science. The take-away message is that domain-tailored strategies will be required for any productised CCR system and a one-size-fits-all approach may not be sufficient.

Our early experiments aimed to find these citing patterns between different types of sentences, but in our later ones we turned our attention to the citing context and how to extract a better performing query. Previous annotation of sentences relevant to a cited document provided us with the starting point: we confirmed that using only sentences that were manually selected as relevant to a citation led to higher evaluation scores than using symmetric windows of sentences.

We then started looking at how to extract a more fine-grained query. We confirmed that for our task, requiring that the query matches a keyphrase does not improve evaluation scores over simply setting the right boost or weight to each term in the query. Next, we considered the contribution of stopwords, and our findings came as a surprise. Our expectation was that the dampening of the IDF component of TF-IDF would make their contribution minimal, but in practice, we find that what one considers to be a stopword can notably alter the results.

For the final part of Chapter 6, of *keyword annotation* and *keyword extraction*, again we encountered unexpected situations. Our hypothesis was that if we selected, from a citation context, the minimal set of keywords that would maximise the evaluation score for that context, we would obtain the set of keywords that most uniquely identified the cited paper. We expected that this set of keywords would somehow capture the essence of the paper that makes it relevant to the citation context. It turns out this is not so, and the keywords that maximise the score of a given paper to a context are just oddities: content words that are frequent in that paper (or in other papers citing it) but infrequent over the collection, or even frequent non-content words that are more frequent than usual in the paper (or in the contexts citing it). This made us change our approach from our original aim of identifying keywords to simply treating every term as a potential keyword and aiming instead to find the right weight to set to each. This had implications for our original research aims with machine learning approaches: applying recurrent deep neural networks to this task needed to be left for future work. Formulating this task as sequence regression instead of sequence labelling needs a rethinking of the standard architectures employed for RNNs. One option is to convert it into a classification problem by binning the boost scores to apply, but this requires further research into how to create this classification, based on the distribution of boost values in the generated annotated data.

In retrospect, we underestimated the differences between the two domains in terms of our NLP approaches. Any work which tries to generalise to yet more corpora would be well advised to do some initial data profiling. As we undertook finer-grained extraction of the query in the final chapter, it became evident that an analysis of domain-specific stopwords and preprocessing techniques, particularly tokenisation, would have been a useful first step.

Let us recap our hypotheses, as we defined them in the opening to this thesis:

- H1: The rhetorical or argumentative function of sentences in an academic document can be classified into a number of discrete classes using shallow semantics.

- H2: The classification of a citation context via H1 helps to identify relevant content in candidate reference papers in a corpus.

- H3: The identification of fine-grained relevant content via H2 improves the ranking of papers that are relevant with respect to a given citation context.

- H4: The application of linguistically motivated natural language processing methods to the selection and weighting of terms for a query improves the query's performance.

In terms of these definitions, H1 (we can automatically classify sentences using shallow semantics) we knew to be true from previous work, H2 (these classes help us identify relevant context) proved correct in our experiments, but H3 (this helps improve the ranking of papers) we could never prove. Finally, H4 (linguistically motivated NLP methods lead to better query extraction) is broadly proven, if also shown to be domain-specific.

7.2 Findings

This thesis presents original work on applying scientific discourse annotation to the task of recommending relevant scientific literature to the author of a draft paper.

	Corpus	Document representation	Context extraction	Similarity
Chapter 3 (Duma and Klein, 2014)		Title + abstract Full text Inlink contexts (many parameters)	Window (10, 20, 30, 50, 100) Sentences (1only, [1,2] up, [1,2]down)	Lucene ClassicSimilarity (Cosine distance + length-averaged tf-idf + coord)
Chapter 5 part I (Duma et al., 2016a)	~1 million (PubMed Central OAS) ~22k (ACL Anthology)	Full text: Index by sentence class (CoreSC & AZ)	Sentences (3; 1 up, 1 down)	Lucene ClassicSimilarity+ per-class weighting based on class of citing sentence
Chapter 5 part II (Duma et al., 2016b)		Inlink contexts: Index by sentence class (automatically labelled CoreSC & AZ)	Sentences (3; 1 up, 1 down)	Lucene ClassicSimilarity+ tf-idf + per-class weighting based on class of citing sentence

Table 7.2.1: A grid comparison of our current work using the same categories as the previous work presented in Chapter 2: the corpora and the number of documents employed, and the methods used for document representation, context extaction and similarity metrics.

We conclude with a summary of findings of our work:

- First, in Chapter 3 we explored different methods for selecting what spans of text of a scientific document to index and how to select the span from which to extract a query in a draft document in order to obtain more relevant results. This is to our knowledge the most exhaustive cross-corpus examination of parameters for extracting text from the target document and the anchor text of its incoming citations, as well as to generate the query that will retrieve it. Table 7.2.1 offers an overview of our different sets of experiments.

- Our results show that different domains cite differently: what is true of computational linguistics (a document is better described by the summaries made of it in other papers) does not hold for biomedical science. To our knowledge, no previous work on CCR specifically examines the differences between scientific domains, instead by default conflating them all by using cross-domain corpora.

- Then, we built on our baseline approaches to explore the application of scientific discourse annotation to this task. This entirely novel approach uses the class of a sentence in the draft document, as automatically classified, to determine what weights should be assigned to different types of sentences in target documents in order to increase the scores using our evaluation method, citation resolution. In order to find these weights, we apply a hill climbing approach that required novel optimisation solutions.

- We applied two separate scientific discourse schemes, Argumentative Zoning and Core Scientific Concepts, that were tailored to different domains, to two large corpora in these two domains. We find consistent patterns between types of citing sentences and types of cited sentences, which are encouraging as to the application of argumentation schemes to the task of contextual citation recommendation. One example is that sentences of type Conclusion in cited biomedical papers are consistently cited by other sentences of type Conclusion or Background in citing articles.

- We then applied this new approach to the anchor text of a document's citations and again found consistent citing patterns, which points to promising future work. At the same time, these results combined suggest that scientific discourse annotation schemes could find use in the field of scientometrics.

- In Chapter 6, we started by exploring the performance of a human annotator in selecting the right keywords to retrieve a previously cited paper. First we evaluated context extraction and query generation using a corpus of citation contexts annotated with the sentences relevant to a citation. Our experiments show that extracting keywords from the sentences a human annotator selected as relevant performed better than extracting all keywords within the two sentences window. Then we performed manual annotation of our own test data, but this time at the keyword level. Again we saw that a human annotator can match or outperform a stronger baseline based on removing collection-specific stopwords. However, we also saw that this applies when what is indexed per document is the full text of it (internal methods). When we use information external to the document itself, the human annotator can no longer outperform the baseline, which shows that for the task of CCR, selecting the relevant sentences in a citation's context is a less promising approach.

- We investigated two ways of improving query extraction: keyphrases and stopwords. We employed standard approaches to keyphrase extraction and demonstrated that adding phrases to be matched to a query is of little to benefit to the CCR task: what really counts in our formulation given our choice of similarity function and query type is the boosting / weighting of the terms. Our experiments with stopwords show their contribution to the evaluation scores, and also how important their careful selection and removal is.

- Perhaps a main contribution is showing that linguistic, morphological, structural and statistical features of the words in a text can be used as predictors of the ideal boost to assign to a given term in a query. We contribute a novel approach to annotation of this ideal per-term boost, based on finding the weights that will maximise the evaluation score for that context from the results of the evaluation query.

- We show that simple per-token regressors outperform the baselines of query generation for CCR in the AAC corpus. Crucially, we also show that the training data for these supervised models can be automatically generated without any human input beyond the implicit information in the original publications. This proves that at least in some domains, the relative importance of a term to a citation can be estimated purely from these features. For biomedical science, this approach fails to beat a simple stopword removal baseline, which on the other hand shows that any approach to CCR will need to be custom tailored to each domain it operates in.

7.3 Future work

7.3.1 Natural Language Processing

It is common practice in information retrieval applications to preprocess the text to be indexed, as well as the text of the query. For example, for languages like English with inflected word forms, it is common practice to lemmatise or stem terms in order to mitigate term sparsity and index all terms that are almost identical in meaning as a single term (Manning et al., 2008). In the approaches we presented in this thesis, we do not explicitly address this. We indirectly mitigate the problem of very related words being indexed separately to some degree by employing mixed document representations, i.e. concatenating the full document text and the text of incoming link contexts. However, this issue should be specifically addressed in future work.

Other standard preprocessing techniques that should help for this task involve: 1. preprocessing the text we index for a given document to annotate synonyms and 2. applying query expansion. Query expansion would align with the approach of *keyword generation* that we introduced in Chapter 6, where the query would contain terms not found in the original document but that we know are correlated or are synonyms.

However, we see that much higher gains can come from going beyond terms, keywords and keyphrases and applying Information Extraction and Text Mining techniques to uniquely identify every entity mentioned in the text, as well as statements about the methodology and findings of the research.

Named Entity Recognition (NER) is one of the most commonly researched tasks in the automatic processing of biomedical scientific articles (Goulart et al., 2011). The recognised entities are commonly disambiguated and linked to large databases of genes, proteins, chemical compounds, etc. Rich annotated resources have been created for BioNLP tasks, such as the GENIA corpus[1], where MEDLINE abstracts were manually annotated with POS tags, chunk tags and named entity tags, and more recently biomedical events (Nawaz et al., 2010; Liakata et al., 2012b; Thompson et al., 2011) which has led to the training of performant taggers[2]. These biomedical events are “template-like, structured representations of pieces of knowledge contained within sentences” (Liakata et al., 2012b). They represent data that can be connected to databases of genes, proteins and chemical compounds and to ontologies of interactions, and therefore extracted as discrete data and connected to other similar data. Another interesting approach in the biomedical domain is clause annotation (Waard et al., 2009), which proposes a taxonomy of discourse segment types and sits between sentences and events in level of granularity. Examples of segment types include “fact” (knowledge accepted to be true), “hypothesis” (a proposed idea, not supported by evidence) and “problem” (unresolved, contradictory or unclear issue).

[1] http://www.geniaproject.org/genia-corpus

[2] e.g. http://www.nactem.ac.uk/GENIA/tagger/

Not every domain can currently benefit from these sophisticated approaches available for BioNLP, but work exists which applies similar methods to other domains of science. One example of this is the work of Gábor et al. (2016), who define a number of semantic relations such as "affects", "used_for", "composed_of", "propose", "yields" and "datasource". The novelty is that instead of manually defining ontological and lexical resources for annotating entities in the text, they derived these automatically using a framework for automatic domain-independent term extraction (Buitelaar et al., 2013). They annotate a few hundred abstracts from the ACL Anthology with this scheme but propose it would generalise to other domains. Although the extent to which they could be applied to other domains remains to be studied, we believe that a conceptually similar approach could be applied to the CCR task and to extracting more clearly structured information for retrieval.

7.3.2 Scientific Discourse Annotation

As we have pointed out earlier in this thesis, we have made use of the best classifiers we could obtain for AZ and CoreSC. However, it is clear that their accuracy leaves room for improvement and at present they introduce significant amounts of noise. Better classifiers is one way to go, but a great deal of expert work is required for annotating a corpus with these semantic categories (zones) so the amount of training data is necessarily limited, which in turn limits the potential accuracy and coverage.

Work has been done towards fixing this, for example using weak supervision (Guo et al., 2011) or active learning (Guo et al., 2013) to deal with the sparsity of the annotated data. Perhaps these approaches could be combined with less brittle neural approaches to yield more performant classifiers. However, a more interesting question is whether we should be treating this as a classification task in the first place.

It is an attractive simplification to categorise the contribution of a sentence to a document as one of several discrete classes. However, a softer, more nuanced, multidimensional way of capturing semantics at a sub-sentence level is much more powerful and could significantly contribute to the CCR task. Compositional neural representations of meaning based on distributional similarity of words offer the way forward here.

7.3.3 Deep neural networks

Much has changed in the field since the beginning of this research project. In these intervening years, we have seen neural approaches progressively outperforming traditional approaches in all areas of Artificial Intelligence, including Natural Language Processing (e.g. part-of-speech tagging, named entity recognition, parsing, machine translation) (Goodfellow et al., 2016). Particularly, symbolic approaches to semantics, based on expert-built resources with little coverage have given way to distributional approaches where the meaning of any span of text can be automatically learned and encoded in a latent space of meaning.

A family of models that has been particularly fruitful for these tasks is the Recurrent Neural Network (RNN), particularly when employing Long-Short Term Memory (LSTM) cells (Hochreiter and Schmidhuber, 1997). While these approaches tend to lack interpretability, they are now the foundation of the the state-of-the-art in many NLP tasks. For sequence labelling tasks, these methods are often combined with others, like for example Conditional Random Fields (Huang et al., 2015b), which contribute to capturing sequential dependencies between items.

It is now standard to represent words as vectors in a high-dimensional embedding space, and not as discrete elements. The line of work that popularised this approach is known as word2vec (Mikolov et al., 2013). Since word2vec took the world by storm, a number of competing approaches to multi-dimensional word representation have been proposed. The current state of the art has been pushed by two recent approaches to this. One is ELMo word representations (Peters et al., 2018): "deep contextualised word representations that model both (1) complex characteristics of word use (e.g., syntax and semantics), and (2) how these uses vary across linguistic contexts (i.e., to model polysemy)". Another is ULMFiT: Universal Language Model Fine-tuning for Text Classification (Howard and Ruder, 2018), a transfer learning method to adapting embedding word representations. Both claim very significant reductions in error rates on tasks such as coreference resolution, named entity recognition, semantic role labeling and very importantly on the Stanford Question Answering Dataset (SQuAD, Rajpurkar et al. (2018)).

We envision that either a recurrent or convolutional neural network using these word representation would significantly outperform the simple baseline token-based models for keyword extraction and boosting we presented in Chapter 6, particularly as the boost scores to apply to the terms are relative to each other. However, beyond words, these approaches allow for compositionality of sentences and longer spans of text.

In Chapter 1, we motivated our choice of using existing keyword-based IR backends based on their maturity and scalability. Indeed, at the time of writing we are unaware of any open source frameworks available of the level of functionality and maturity of classical keyword-based IR frameworks such as Elasticsearch.

While backends that enable search over multidimensional representations of words are now becoming more common, these solutions are either bespoke or proprietary. However, it seems clear to us that our CCR task can greatly benefit from neural approaches to semantic modelling.

7.3.4 Evaluation

Given a context c, and a document collection D, we cannot assume that the author of c knows the whole contents of the document collection well enough to cite all those that are relevant. So, while we can treat the human relevance judgments (existing citations in published papers) as gold-standard *true positives*, we cannot assume they are the full set of true positives, so in future work we would explore methods for expanding the set of relevant papers to each citation placeholder and treat this task as lightly supervised.

Ultimately, our evaluation method, which has almost exclusively been used in the literature, remains a very rough approximation of our ultimate task. User studies would be an important next step, but the final evaluation would need to be based on usage metrics of an existing system. It is likely that at this point the task would need to concern itself with specific use cases in order to keep adding value for its users.

7.3.5 Scientific publishing and Open Access

Although the global push for Open Access publishing has seen significant wins over the past few years, we are still far from a world where all published science is accessible to anyone. This remains a significant obstacle to building a practical CCR system: only a large corporation of the size, resources and connections of Alphabet, Microsoft or Apple could at present access and index a wide range of scientific publications, as significant funds and connections are required to collate the paywalled information from publishers.

Like all others, the field of scientific publishing is constantly evolving, and there is much discussion currently not just about the business model of academic publishing and the push for open access publications, but also about the format in which those papers are published[3]. Perhaps the initiative gathering the most strength is FORCE11, a "community of scholars, librarians, archivists, publishers and research funders" coming together to "bring about a change in modern scholarly communications". Their manifesto[4] points out that existing formats used in scientific publication "needlessly limit, inhibit and undermine effective knowledge transfer" so they aim to "rethink the unit and form of the scholarly publication". We strongly believe in this mission and envisage a future for scientific publishing beyond the academic paper, where the paper is published together with the data collected in structured machine readable formats with rich metadata, that could greatly contribute to CCR.

7.4 Final words

We remain convinced that practical Contextual Citation Recommendation systems would greatly assist researchers in their work by, in the first instance, facilitating the discovery of relevant literature at every stage of research.

The ultimate aim remains to augment human cognitive capabilities. Our knowledge of the world is now mediated by the ability of search engines (Google) and virtual assistants (Alexa, Google Assistant, Siri) to immediately answer our questions; the memory of our life events is now externalised to the cloud in the photos we capture and the messages we send; and our spatial knowledge is augmented by GPS-assisted maps and navigation in our pocket. In the same way, the knowledge and understanding of all published science should be available to us at the moment that we need it, presented to us in a contextually useful way.

There is no better time to make this happen than now. In this work we have only explored some directions this research could follow, but everything remains still to be done. However, the technology needed to make it happen is finally reaching the right stage of maturity, and a combination of new Information Extraction and Text Mining techniques, Open Access publishing and a push for new publishing formats will be essential. The deep learning revolution in NLP paves the way to a deeper understanding of the text, and so to ways of better satisfying the information need of the academic author.

[3] https://www.theatlantic.com/science/archive/2018/04/the-scientific-paper-is-obsolete/556676

[4] https://www.force11.org/about/manifesto

Appendix A

Data preprocessing

Normalising and de-noising textual data is complex and very time consuming. A lot of effort went into first obtaining and then cleaning the machine readable XML-tagged documents, particularly the ACL Anthology Corpus. A large number of tasks are needed to connect documents to each other using citations. A significant amount of noise can be introduced by each of these following tasks, and the way we mitigate it is by applying a large amount of heuristics in the shape of Regular Expressions (RegEx).

- *Sentence splitting*. Far from a solved task, splitting text into sentence units is problematic. Spans of text like "et al." can break sentences, which often also happens at page boundaries or when extraneous tags appear in the text, such as footnotes. An example heuristic is that if a sentence ends in ' al.' and the next sentence in the document starts with a lowercase letter, we join the two together and update all references to sentence and citation numbers.

- *Incorporating formatting*. An academic paper is not just free flowing text. All kinds of footnotes, tables, figures and captions need to be dealt with, but also bullet points and inline titles. The conversion of these in the AAC corpus is inconsistent and required significant preprocessing.

- *Recognising a citation in text*. There are vary many citation styles in use, and although a given domain tends towards a given citation style, it is far from homogeneous. Citations of the type of *(Author, 1999)* require a double approach: recognising the pattern using RegEx and building a gazeteer made of the possible renditions of that reference in text, which in turns requires clean reference metadata. There are many possible patterns using this citation style, particularly in multi-citation. For example, a citation may mention a method and in the parentheses that follow offer an acronym for it, followed by a comma, and the author and year. Given that often sentences are badly split because of page breaks or extraneous elements conflated with the text (e.g. table captions), false positives and primarily false negatives are often found. In theory the AAC corpus has already been annotated with all of this information. In practice, we find that many citations are not recognised and many references in the bibliography are broken, so it was necessary to apply these steps again using specifically targeted RegEx and heuristics. On the other hand, citations of type [1], which on first approach may look straightforward, also incur in ambiguity and particularly false positives. Some studies in PMC report numbers surrounding them with square brackets, so [223] will not be a link to reference number 223, but a quantity.

- *Removing inline author names.* In ACL author names are typically cited inline (e.g. "McKeown et al. show that..."). We use the same gazetteer approach and substitute all inline mentions with a placeholder ("__author"), which gets tokenized as a single token.

- *Matching the citation with its corresponding reference in the paper's bibliography.* Some citations that were identified as such in running text could not be matched because their corresponding reference was broken, often hidden at the end of the string representing the "title" of another paper.

- *Matching a reference in document d_1 to the corresponding paper d_2 in the collection D.* This one depends on having clear metadata about all documents in the collection. Thankfully the ACL Anthology Network (AAN) provides this, so all that is needed is a simplification of the strings, removing all punctuation and normalising spaces. Then an exact string matching of the title performs very satisfactorily.

- *Detecting self-citations.* To determine whether an author is self-citing, we convert the list of authors of the citing paper and the cited paper into a list of plain strings, taking the author's given name followed by the family name, and do exact string matching. This is an approximate method which we have not evaluated except anecdotally.

Poor quality OCR complicates all the above tasks. As an example, 'T1' will become 'Ti', beyond more obvious ones, like the letter 'l' becoming the number '1' and viceversa. Converting equations to text consistently leaves stranded tokens such as *j* and *h* and *i* that appear in running text.

On top of this, the AAC corpus comes with formatting XML in the running text, including <i> and <b> tags. Often these tags are broken, i.e. do not have a closing one. No manual data cleaning was performed, so while less, there is still some noise in the data.

Appendix B

Final trained weights

Query type ↓	AIM	BAS	BKG	CTR	OTH	OWN	TXT
AIM	1	1	1	1	1	1	1
BAS	0	0	1	0	1	7	0
BKG	0	0	1	0	0	4	0
CTR	0	0	1	0	1	7	0
OTH	1	0	0.5	0	1	4	0
OWN	0	0	1	0	1	7	0
TXT	1	1	1	1	1	1	1

Table B.0.1: Trained weights for Queries B for AAC internal methods.

Query type ↓	ilc_AIM	ilc_BAS	ilc_BKG	ilc_CTR	ilc_OTH	ilc_OWN	ilc_TXT
AIM	1	1	1	1	1	1	1
BAS	0.5	0	0	0	0.5	7	0
BKG	0	0	0.5	0	1	2	0
CTR	3.5	0	0	0	0	7	0
OTH	0	0	0	0	0	7	0
OWN	0	0	0	0	0	7	0
TXT	1	1	1	1	1	1	1

Table B.0.2: Trained weights for Queries B for AAC external methods.

Query text ↓	ilc_Bac	ilc_Con	ilc_Exp	ilc_Goa	ilc_Hyp	ilc_Met	ilc_Mod	ilc_Mot	ilc_Obj	ilc_Obs	ilc_Res
Bac	2	0	0	0	0	1	0	0	0	0	1
Con	4.5	0	0	0	0	1	0	0	0	0	0
Exp	1	1	1	1	1	1	1	1	1	1	1
Goa	7	1	0	1	0	1	0	0	1	0	0
Hyp	7	1	0	0	0	1	1	0	0	0	1
Met	1	0.5	0	0	0	1	0	0	0	0	1
Mod	1	1	1	1	1	1	1	1	1	1	1
Mot	1	1	1	1	1	1	1	1	1	1	1
Obj	1	0	0	0	0	1	0	0	0	0	1
Obs	1	1	1	1	1	1	1	1	1	1	1
Res	2	1	0	0	0	0.5	0	0	0	0	1

Table B.0.4: Trained weights for Queries B for PMC external methods.

Query type ↓	Bac	Con	Exp	Goa	Hyp	Met	Mod	Mot	Obj	Obs	Res
Bac	1	1	1	0	0	1	1	0	0	0.5	1
Con	1	1	1	0	0	1	0.5	0	0	0	0
Exp	1	1	1	1	1	1	1	1	1	1	1
Goa	1	0	0	0	0	1	0	0	0	0	0
Hyp	1	1	1	1	1	1	1	1	1	1	1
Met	2	0	0	0	0	0.5	0.5	0	0	0	0.5
Mod	1	1	1	1	1	1	1	1	1	1	1
Mot	1	1	1	1	1	1	1	1	1	1	1
Obj	7	0	1	0.5	0	1	0	0	0	1	2
Obs	1	1	1	1	1	1	1	1	1	1	4
Res	1	0	0	0	0	0	0	0	0	0.5	1

Table B.0.3: Trained weights for Queries B for PMC internal methods.

Appendix C

Domain-specific stopwords

AAC				PMC			
Term	Score %	Match %	Boost %	Term	Score %	Match %	Boost %
in	5.45	4.07	0.99	in	5.61	2.71	0.69
is	2.90	2.17	0.78	is	1.34	0.70	0.58
for	2.85	2.16	0.93	for	1.19	0.60	0.50
we	2.00	1.54	0.68	that	0.83	0.33	0.37
as	1.54	1.18	0.83	are	0.61	0.41	0.46
that	1.32	0.99	0.67	on	0.57	0.24	0.35
on	1.14	0.88	0.70	as	0.56	0.28	0.38
are	1.02	0.79	0.58	this	0.50	0.33	0.53
this	0.91	0.73	0.67	was	0.42	0.19	0.30
based	0.68	0.56	0.57	be	0.40	0.21	0.37
be	0.56	0.41	0.44	have	0.31	0.14	0.31
which	0.56	0.41	0.58	were	0.28	0.14	0.17
using	0.51	0.39	0.55	also	0.27	0.17	0.48
used	0.44	0.35	0.48	at	0.26	0.18	0.35
our	0.44	0.33	0.36	been	0.23	0.11	0.30
can	0.34	0.26	0.37	has	0.23	0.16	0.40
have	0.34	0.26	0.42	can	0.20	0.14	0.30
has	0.33	0.25	0.43	it	0.18	0.11	0.29
use	0.31	0.24	0.44	these	0.17	0.08	0.21
it	0.31	0.22	0.30	based	0.16	0.06	0.09
been	0.29	0.21	0.38	used	0.13	0.08	0.20
such	0.27	0.22	0.40	using	0.11	0.06	0.15

these	0.26	0.20	0.35	more	0.11	0.07	0.17
set	0.24	0.19	0.23	found	0.09	0.03	0.10
each	0.23	0.20	0.31	during	0.08	0.06	0.17
two	0.22	0.17	0.31	may	0.07	0.03	0.09
was	0.21	0.15	0.25	than	0.06	0.05	0.13
also	0.21	0.16	0.33	but	0.06	0.03	0.14
other	0.20	0.17	0.33	all	0.06	0.02	0.07
one	0.20	0.15	0.26	only	0.06	0.05	0.16
more	0.15	0.12	0.22	most	0.06	0.04	0.14
different	0.15	0.11	0.21	they	0.05	0.02	0.06
between	0.14	0.12	0.22	each	0.05	0.04	0.09
some	0.14	0.10	0.18	where	0.05	0.03	0.09
at	0.14	0.10	0.19	both	0.04	0.03	0.11
were	0.13	0.10	0.16	al	0.04	0.01	0.03
all	0.13	0.10	0.19	et	0.04	0.01	0.03
into	0.13	0.10	0.21	when	0.04	0.02	0.07
only	0.13	0.09	0.17	well	0.04	0.02	0.10
most	0.13	0.09	0.20	there	0.04	0.02	0.06
where	0.12	0.09	0.21	into	0.03	0.02	0.07
their	0.11	0.09	0.19	those	0.03	0.01	0.04
but	0.11	0.09	0.18	had	0.03	0.02	0.07
they	0.10	0.08	0.17	some	0.03	0.02	0.08
both	0.09	0.08	0.19	within	0.03	0.02	0.06
however	0.09	0.07	0.18	will	0.03	0.02	0.04
first	0.09	0.06	0.15	thus	0.03	0.02	0.10
its	0.09	0.07	0.17	after	0.03	0.02	0.06
than	0.09	0.07	0.13	showed	0.03	0.01	0.02
then	0.08	0.06	0.15	shown	0.03	0.01	0.05
same	0.08	0.06	0.14	due	0.03	0.02	0.07
many	0.08	0.06	0.14	under	0.02	0.01	0.06
given	0.07	0.05	0.12	being	0.02	0.02	0.12
there	0.07	0.05	0.11	would	0.02	0.00	0.02
when	0.07	0.05	0.12	known	0.02	0.01	0.05
well	0.07	0.05	0.13	over	0.02	0.01	0.04
possible	0.05	0.04	0.08	while	0.02	0.01	0.03

how	0.04	0.03	0.08
them	0.04	0.03	0.09
three	0.04	0.03	0.08
will	0.04	0.03	0.07
may	0.04	0.03	0.05
if	0.04	0.03	0.05
so	0.04	0.03	0.08
following	0.04	0.03	0.08
new	0.03	0.03	0.07
any	0.03	0.02	0.06
would	0.03	0.02	0.05
about	0.02	0.02	0.07
do	0.02	0.02	0.04
no	0.02	0.02	0.05
TOTAL PCT	29.63	22.54	20.35

could	0.02	0.01	0.03
no	0.02	0.01	0.03
among	0.02	0.01	0.03
even	0.02	0.01	0.05
very	0.02	0.00	0.01
either	0.01	0.01	0.04
we	0.01	0.01	0.02
about	0.01	0.01	0.03
since	0.01	0.01	0.02
made	0.01	0.01	0.05
highly	0.01	0.01	0.03
show	0.01	0.01	0.04
might	0.01	0.00	0.01
did	0.01	0.00	0.02
up	0.01	0.00	0.01
before	0.01	0.00	0.01
does	0.01	0.00	0.01
still	0.01	0.00	0.02
above	0.01	0.00	0.02
form	0.01	0.00	0.02
if	0.01	0.00	0.02
per	0.00	0.00	0.01
do	0.00	0.00	0.01
least	0.00	0.00	0.02
so	0.00	0.00	0.01
given	0.00	0.00	0.00
out	0.00	0.00	0.00
should	0.00	0.00	0.00
then	0.00	0.00	0.02
them	0.00	0.00	0.00
It	0.00	0.00	0.01
any	0.00	0.00	0.00
here	0.00	0.00	0.01
TOTAL PCT	16.36	8.45	11.18

Appendix D

Keyword annotation task: manual annotation guidelines

D.1 Purpose

Picture yourself as the author of a scientific paper. In academic writing, it is essential to support the claims you make, either with evidence or by citing other scientific articles that have already been published. The purpose of this annotation exercise is to research what keywords found in the text of a draft academic paper are more helpful for finding the most appropriate references from an existing collection of documents, using a search engine.

You will be provided with a number of citation contexts, which are paragraphs of text extracted from scientific articles. These paragraphs originally had citations to other articles in them, which for the purposes of this annotation have been replaced with placeholders. These placeholders are of two types:

- **[CITATION HERE]:** the citation we are interested in finding, based on the text around it
- **[other cit]:** another citation that was present in the original text but which we do not concern ourselves with.

You will also see other placeholders that read **__author**. In this location, the name of the author of a paper has been anonymised and you should ignore this token. Keep in mind that several **__author** tokens in one context can refer to different author names, or to the same one: this is not known.

D.2 Task

The task can be summarised as: if, using a search engine (e.g. Google Scholar), you are trying to find the original paper cited where **[CITATION HERE]** appears, which words from the text would you select?

Think only of the document contents: *which words in the citation's context are likely to appear in the cited paper*?

We are exploring documents in two different domains: computational linguistics (labeled **AAC**) and biomedical science (labeled **PMC**). These domains have their important differences, which you should be aware of, so to familiarise yourself with these, please skim read through both files first.

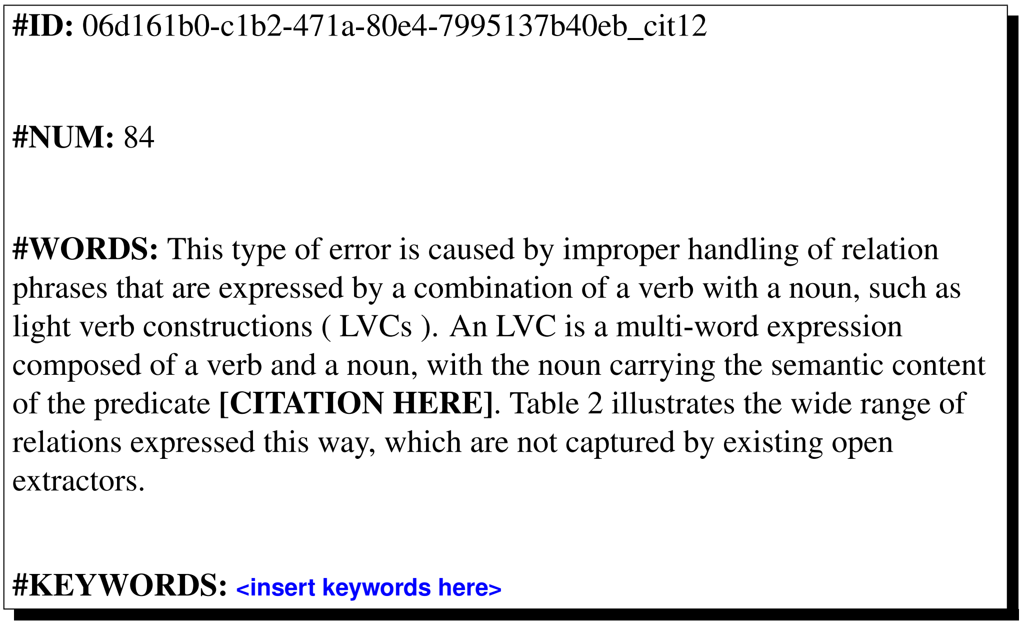
You will receive one file to annotate for each domain, and each contains a total of 100 different paragraphs (citation contexts) to annotate: **keywords_aac.txt** and **keywords_pmc.txt**.

Please save these files as **keywords_aac_annotated.txt** and **keywords_pmc_annotated.txt** and save your changes in them, as documented below.

D.3 Examples

Each file contains 100 contexts. Figure D.3.1 presents one example context from the AAC set that you will not need to annotate and is only provided for illustration. Please avoid modifying the text on the lines that start with **#ID**, **#NUM**, and **#WORDS**. Simply copy the words or span of text from the line starting with #WORDS and paste them after **#KEYWORDS**: leaving a space between "#KEYWORDS" and the text you paste. Please make sure to only copy and paste full words, not parts of them.

In this example, without knowing anything about the subject matter or which paper was cited, we can glean that the citation that is missing where [CITATION HERE] appears is related to "light verb constructions".



#ID: 06d161b0-c1b2-471a-80e4-7995137b40eb_cit12

#NUM: 84

#WORDS: This type of error is caused by improper handling of relation phrases that are expressed by a combination of a verb with a noun, such as light verb constructions (LVCs). An LVC is a multi-word expression composed of a verb and a noun, with the noun carrying the semantic content of the predicate **[CITATION HERE]**. Table 2 illustrates the wide range of relations expressed this way, which are not captured by existing open extractors.

#KEYWORDS: <insert keywords here>

Figure D.3.1: Example context from AAC.

One example of a full set of keywords we could extract to reflect this would be: "light verb constructions LVCs LVC multi-word expression verb noun semantic content predicate". Note that we select both "LVCs" and "LVC", as these are different words. The word "noun" appears twice, but selecting it once is sufficient. We can also select it several times, as this does not matter. We would simply copy and paste this list of words after #KEYWORDS:

> **#KEYWORDS:** light verb constructions LVCs LVC multi-word expression verb noun semantic content predicate

Another way to do this is to simply copy and paste the whole relevant span of text, for example:

> **#KEYWORDS:** light verb constructions (LVCs). An LVC is a multi-word expression, verb and a noun, with the noun carrying the semantic content of the predicate

See how we have omitted the span of text "*composed of a*" and copied and pasted what came before and after that. Note that this is merely an example of what we could annotate, and not an authoritative annotation. For example, we may decide that the words "relation phrases" found earlier in the text are also relevant to the citation. What exactly to include is up to you to decide as the annotator.

The AAC and PMC domains differ substantially in vocabulary and style, as can be seen in this example from PMC (again, not appearing in the corpus to annotate):

> **#ID:** 3244bfe6-9f77-4628-9cdf-24e446405183_cit36
>
> **#NUM:** 203
>
> **#WORDS:** As for HAM/TSP, there were many reports dealing with HAM/TSP treatment, although they included a limited number of patients and need further studies. Such examples are listed : safety and efficacy of a humanized monoclonal anti-IL15Rβ antibody (CD122) __author [other cit] potential use of BNZ-gamma peptide (selectively blocking binding and downstream signaling of IL-2, IL-9 and IL-15) as a new drug __author [other cit] clinical study using Infliximab (anti-TNF-α monoclonal antibody) __author [CITATION HERE] efficacy of Methotrexate __author [other cit] and efficacy of Fampridine (a selective neuronal potassium channel blocker) __author [other cit]. One of the problems in these reports is the lack of standard criteria for treatment efficacy, which makes it difficult to compare the results from the different clinical research teams.
>
> **#KEYWORDS:**

Here we see a number of papers cited, all around the same theme: "HAM/TSP treatment". The citation we are interested in seems to be about "Infliximab", so one potential set of keywords to extract would be:

#KEYWORDS: HAM/TSP treatment clinical study using Infliximab (anti-TNF-α monoclonal antibody)

D.4 Instructions

- It is recommended that you first skim read through each of the two files before starting the annotation task, to familiarise yourself with each textual domain and acquire an awareness of the differences between them.

- For each citation context (paragraph):

  - **INCLUDE:** all the words as you think are likely to be present in the paper that was originally cited.

  - **EXCLUDE:** words that you think are unrelated to the missing citation and unlikely to appear in that paper, and therefore would add less relevant search results.
  - **WHEN IN DOUBT:** err on the side of selecting more keywords rather than fewer.

- For consistency, please make sure to copy/paste from the text rather than type. Only copy full words, not parts of them. In the cases where words are hyphenated, e.g. "*multi-word*" in the example above, or "*anti-TNF-α*" in the PMC text, count this as a single word and copy it including the hyphen.

- For ease of annotation, feel free to copy/paste any length of text directly from *#WORDS*, with or without punctuation in it. Stopwords (grammar words) such as "*for*", "*and*" and "*to*" will be automatically filtered out later on, as well as all punctuation such as commas, parentheses and full stops, so you need not worry about including them or not.

- All the keywords you select will be used as part of a single query. Don't be afraid to select a larger query: the more specific the query, the more likely it will be to find the originally cited paper. But keep in mind the rules above and avoid including what you think is not relevant.

- Keywords only count once: there is no harm in adding a single word several times, but it will not count more than once either.

- Be aware of editorial conventions: in the PMC corpus, when you see ", *[CITATION HERE]*", the citation refers to the text that precedes the comma.

- Pay attention to the function of the citation. Sometimes a citation backs up a specific claim, for example:
  "The authors studied whether CMR improves outcomes prediction after contemporaneous echocardiography in a prospective study of 444 patients clinically referred for CMR ***[CITATION HERE].****"*

  - Here, it is likely that the full span of text "CMR improves outcomes prediction after contemporaneous echocardiography in a prospective study of 444 patients clinically referred for CMR" is relevant to finding the originally cited paper

  - Citing expressions like “The authors studied” or “According to a study by” should never be included as keywords to extract.

- Other times, a citation simply acknowledges work that has defined or popularised a technique or approach, for example:
  *“The raw sentences in all the training and test documents were preprocessed using MXPOST **[CITATION HERE]** and the MST dependency parser [other cit] following __author et al. (2010a).”*

  - Here, the only intuitively relevant part of text seems to be the unique name “MXPOST” which refers to a particular system that the cited paper presents and describes. This paper will be very uniquely identifiable in the whole collection just through this one word.

- Consider the other citation in this paragraph:
  *“The raw sentences in all the training and test documents were preprocessed using MXPOST [other cit] and the MST dependency parser **[CITATION HERE]** following __author et al. (2010a).”*

  - For this other citation, we have a similar situation where “MST dependency parser” seems to be the only relevant part of the text to extract. But again, this is only an example and not authoritative. What to ultimately extract is your decision.

- Once you have chosen the keywords, please have a second read of the paragraph and the keywords you selected! It is easy to miss out on relevant content.

Bibliography

ABU-JBARA, A. AND RADEV, D., 2011. Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 500–509. Association for Computational Linguistics.

ALEX, B. AND BURNS, J., 2014. Estimating and rating the quality of optically character recognised text. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, 97–102. ACM.

ANGROSH, M.; CRANEFIELD, S.; AND STANGER, N., 2013. Context identification of sentences in research articles: Towards developing intelligent tools for the research community. *Natural Language Engineering*, 19, 04 (2013), 481–515.

ANTELMAN, K., 2004. Do open-access articles have a greater research impact? *College & research libraries*, 65, 5 (2004), 372–382.

ATHAR, A. AND TEUFEL, S., 2012. Context-enhanced citation sentiment detection. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 597–601. Association for Computational Linguistics.

BEEL, J.; BREITINGER, C.; LANGER, S.; LOMMATZSCH, A.; AND GIPP, B., 2016a. Towards reproducibility in recommender-systems research. *User Modeling and User-Adapted Interaction*, 26, 1 (Mar 2016), 69–101. doi:10.1007/s11257-016-9174-x. `https://doi.org/10.1007/s11257-016-9174-x`.

BEEL, J.; GIPP, B.; LANGER, S.; AND BREITINGER, C., 2016b. Paper recommender systems: A literature survey. *International Journal on Digital Libraries*, 17, 4 (2016), 305–338.

BIAŁECKI, A.; MUIR, R.; AND INGERSOLL, G., 2012. Apache Lucene 4. *Open Source Information Retrieval*, 1 (2012), 17.

BIRD, S.; DALE, R.; DORR, B. J.; GIBSON, B. R.; JOSEPH, M.; KAN, M.-Y.; LEE, D.; POWLEY, B.; RADEV, D. R.; AND TAN, Y. F., 2008. The ACL Anthology Reference Corpus: A reference dataset for bibliographic research in computational linguistics. In *LREC*.

BJÖRK, B.-C.; LAAKSO, M.; WELLING, P.; AND PAETAU, P., 2014. Anatomy of green open access. *Journal of the Association for Information Science and Technology*, 65, 2 (2014), 237–250.

BLEI, D. M., 2012. Probabilistic topic models. *Communications of the ACM*, 55, 4 (2012), 77–84.

BLEI, D. M.; NG, A. Y.; AND JORDAN, M. I., 2003. Latent dirichlet allocation. *The Journal of machine Learning research*, 3 (2003), 993–1022.

BRADSHAW, S., 2003. Reference directed indexing: Redeeming relevance for subject search in citation indexes. In *International Conference on Theory and Practice of Digital Libraries*, 499–510. Springer.

BRIN, S. AND PAGE, L., 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30, 1 (1998), 107–117.

BUBLITZ, W., 2011. Cohesion and coherence. *Discursive Pragmatics, John Benjamins Publishing Company, Amsterdam/Philadelphia*, (2011), 37–49.

BUITELAAR, P.; BORDEA, G.; AND POLAJNAR, T., 2013. Domain-independent term extraction through domain modelling. In *The 10th international conference on terminology and artificial intelligence (TIA 2013), Paris, France*. 10th International Conference on Terminology and Artificial Intelligence.

CONSTANTIN, A.; PETTIFER, S.; AND VORONKOV, A., 2013. Pdfx: fully-automated pdf-to-xml conversion of scientific literature. In *Proceedings of the 2013 ACM symposium on Document engineering*, 177–180. ACM.

CONTRACTOR, D.; GUO, Y.; AND KORHONEN, A., 2012. Using argumentative zones for extractive summarization of scientific articles. *Proceedings of COLING 2012*, (2012), 663–678.

DAVISON, B. D., 2000. Topical locality in the web. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 272–279. ACM.

DÉJEAN, H. AND MEUNIER, J.-L., 2006. A system for converting PDF documents into structured XML format. In *International Workshop on Document Analysis Systems*, 129–140. Springer.

DUMA, D. AND KLEIN, E., 2014. Citation resolution: A method for evaluating context-based citation recommendation systems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

DUMA, D.; LIAKATA, M.; CLARE, A.; RAVENSCROFT, J.; AND KLEIN, E., 2016a. Applying Core Scientific Concepts to context-based citation recommendation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC-2016), 23-28 May 2016, Portoroz (Slovenia).*

DUMA, D.; LIAKATA, M.; CLARE, A.; RAVENSCROFT, J.; AND KLEIN, E., 2016b. Rhetorical classification of anchor text for citation recommendation. *D-Lib Magazine*, 22, 9/10 (2016).

DUMA, D.; SUTTON, C.; AND KLEIN, E., 2016c. Context matters: Towards extracting a citation's context using linguistic features. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*, 201–202. ACM.

EYSENBACH, G., 2006. Citation advantage of open access articles. *PLoS biology*, 4, 5 (2006), e157.

FISCHER, G., 2001. User modeling in human–computer interaction. *User modeling and user-adapted interaction*, 11, 1-2 (2001), 65–86.

FRANK, E.; PAYNTER, G. W.; WITTEN, I. H.; GUTWIN, C.; AND NEVILL-MANNING, C. G., 1999. Domain-specific keyphrase extraction. In *16th International joint conference on artificial intelligence (IJCAI 99)*, vol. 2, 668–673. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

GÁBOR, K.; ZARGAYOUNA, H.; TELLIER, I.; BUSCALDI, D.; AND CHARNOIS, T., 2016. A typology of semantic relations dedicated to scientific literature analysis. In *International Workshop on Semantic, Analytics, Visualization*, 26–32. Springer.

GARFIELD, E. ET AL., 1965. Can citation indexing be automated. In *Statistical association methods for mechanized documentation, symposium proceedings*, vol. 269, 189–192. National Bureau of Standards, Miscellaneous Publication 269, Washington, DC.

GARZONE, M. AND MERCER, R. E., 2000. Towards an automated citation classifier. In *Conference of the Canadian Society for Computational Studies of Intelligence*, 337–346. Springer.

GAY, C. W.; KAYAALP, M.; AND ARONSON, A. R., 2005. Semi-automatic indexing of full text biomedical articles. In *AMIA Annual Symposium Proceedings*, vol. 2005, 271. American Medical Informatics Association.

GIPP, B. AND BEEL, J., 2009. Citation proximity analysis (CPA): a new approach for identifying related work based on co-citation analysis. In *ISSI09: 12th International Conference on Scientometrics and Informetrics*, 571–575.

GOODFELLOW, I.; BENGIO, Y.; AND COURVILLE, A., 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

GORMLEY, C. AND TONG, Z., 2015. *Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine*. " O'Reilly Media, Inc.".

GOULART, R. R. V.; DE LIMA, V. L. S.; AND XAVIER, C. C., 2011. A systematic review of named entity recognition in biomedical texts. *Journal of the Brazilian Computer Society*, 17, 2 (2011), 103–116.

GUO, Y.; KORHONEN, A.; LIAKATA, M.; KAROLINSKA, I. S.; SUN, L.; AND STENIUS, U., 2010. Identifying the information structure of scientific abstracts: an investigation of three different schemes. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, 99–107. Association for Computational Linguistics.

GUO, Y.; KORHONEN, A.; SILINS, I.; AND STENIUS, U., 2011. Weakly supervised learning of information structure of scientific abstracts - is it accurate enough to benefit real-world tasks in biomedicine? *Bioinformatics*, 27, 22 (2011), 3179–3185.

GUO, Y.; SILINS, I.; STENIUS, U.; AND KORHONEN, A., 2013. Active learning-based information structure analysis of full scientific articles and two applications for biomedical literature review. *Bioinformatics*, 29, 11 (2013), 1440–1447.

HACOHEN-KERNER, Y., 2003. Automatic extraction of keywords from abstracts. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 843–849. Springer.

HALL, D. W. AND PESENTI, J., 2017. Growing the artificial intelligence industry in the UK. *Independent review for the Department for Digital, Culture, Media and Sport/Department for Business, Energy and Industrial Strategy, https://www. gov. uk/government/publications/growing-the-artificial-intelligence-industry-in-the-uk*, (2017).

HASAN, K. S. AND NG, V., 2014. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 1262–1273.

HE, J.; NIE, J.-Y.; LU, Y.; AND ZHAO, W. X., 2012. Position-aligned translation model for citation recommendation. In *String Processing and Information Retrieval*, 251–263. Springer.

HE, Q.; PEI, J.; KIFER, D.; MITRA, P.; AND GILES, L., 2010. Context-aware citation recommendation. In *Proceedings of the 19th international conference on World wide web*, 421–430. ACM.

HIROHATA, K.; OKAZAKI, N.; ANANIADOU, S.; AND ISHIZUKA, M., 2008. Identifying sections in scientific abstracts using conditional random fields. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

HOCHREITER, S. AND SCHMIDHUBER, J., 1997. Long short-term memory. *Neural computation*, 9, 8 (1997), 1735–1780.

HOFMANN, T., 1999. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, 289–296. Morgan Kaufmann Publishers Inc.

HOWARD, J. AND RUDER, S., 2018. Fine-tuned language models for text classification. *arXiv preprint arXiv:1801.06146*, (2018).

HUANG, W.; KATARIA, S.; CARAGEA, C.; MITRA, P.; GILES, C. L.; AND ROKACH, L., 2012. Recommending citations: translating papers into references.

In *Proceedings of the 21st ACM international conference on Information and knowledge management*, 1910–1914. ACM.

HUANG, W.; WU, Z.; LIANG, C.; MITRA, P.; AND GILES, C. L., 2015a. A neural probabilistic model for context based citation recommendation. In *In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

HUANG, W.; WU, Z.; MITRA, P.; AND GILES, C. L., 2014. Refseer: A citation recommendation system. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, 371–374. IEEE Press.

HUANG, Z.; XU, W.; AND YU, K., 2015b. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*, (2015).

HULTH, A., 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, 216–223. Association for Computational Linguistics.

HYLAND, K., 2009. *Academic discourse: English in a global context*. Bloomsbury Publishing.

JHA, R.; JBARA, A.-A.; QAZVINIAN, V.; AND RADEV, D. R., 2017. NLP-driven citation analysis for scientometrics. *Natural Language Engineering*, 23, 1 (2017), 93–130.

JIMENO-YEPES, A. J.; PLAZA, L.; MORK, J. G.; ARONSON, A. R.; AND DÍAZ, A., 2013. MeSH indexing based on automatically generated summaries. *BMC bioinformatics*, 14, 1 (2013), 208.

JURAFSKY, D., 2000. *Speech & language processing*. Pearson Education India.

JUSTESON, J. S. AND KATZ, S. M., 1995. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural language engineering*, 1, 1 (1995), 9–27.

KANG, I.-S. AND KIM, B.-K., 2012. Characteristics of citation scopes: a preliminary study to detect citing sentences. In *Computer Applications for Database, Education, and Ubiquitous Computing*, 80–85. Springer.

KAPLAN, D.; IIDA, R.; AND TOKUNAGA, T., 2009. Automatic extraction of citation contexts for research paper summarization: A coreference-chain based approach.

In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, 88–95. Association for Computational Linguistics.

KATARIA, S.; MITRA, P.; AND BHATIA, S., 2010. Utilizing context in generative Bayesian models for linked corpus. In *AAAI*, vol. 10, 1.

KIM, S. N. AND KAN, M.-Y., 2009. Re-examining automatic keyphrase extraction approaches in scientific articles. In *Proceedings of the workshop on multiword expressions: Identification, interpretation, disambiguation and applications*, 9–16. Association for Computational Linguistics.

KIM, Y. AND WEBBER, B., 2006. Implicit reference to citations : A study of astronomy papers. In *ERPANET*.

KLEINBERG, J. M., 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46, 5 (1999), 604–632.

KONSTAN, J. A. AND RIEDL, J., 2012. Recommender systems: from algorithms to user experience. *User modeling and user-adapted interaction*, 22, 1-2 (2012), 101–123.

KRUSCHWITZ, U., 2005. *Intelligent document retrieval: exploiting markup structure*, vol. 17. Springer Science & Business Media.

LANDAUER, T. K., 2006. *Latent semantic analysis*. Wiley Online Library.

LARSEN, P. AND VON INS, M., 2010. The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index. *Scientometrics*, 84, 3 (2010), 575–603.

LEWIS, D. D. AND SPÄRCK JONES, K., 1996. Natural language processing for information retrieval. *Communications of the ACM*, 39, 1 (1996), 92–101.

LIAKATA, M.; SAHA, S.; DOBNIK, S.; BATCHELOR, C.; AND REBHOLZ-SCHUHMANN, D., 2012a. Automatic recognition of conceptualization zones in scientific articles and two life science applications. *Bioinformatics*, 28, 7 (2012), 991–1000.

LIAKATA, M.; TEUFEL, S.; SIDDHARTHAN, A.; AND BATCHELOR, C. R., 2010. Corpora for the conceptualisation and zoning of scientific papers. In *LREC*.

LIAKATA, M.; THOMPSON, P.; DE WAARD, A.; NAWAZ, R.; MAAT, H. P.; AND ANANIADOU, S., 2012b. A three-way perspective on scientific discourse annotation for knowledge extraction. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, 37–46. Association for Computational Linguistics.

LIU, T.-Y. ET AL., 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3, 3 (2009), 225–331.

LIU, X.; YU, Y.; GUO, C.; SUN, Y.; AND GAO, L., 2014. Full-text based context-rich heterogeneous network mining approach for citation recommendation. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, 361–370. IEEE Press.

LOPEZ, P. AND ROMARY, L., 2010. HUMB: Automatic key term extraction from scientific articles in GROBID. In *Proceedings of the 5th international workshop on semantic evaluation*, 248–251. Association for Computational Linguistics.

LU, Y.; HE, J.; SHAN, D.; AND YAN, H., 2011. Recommending citations with translation model. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11 (Glasgow, Scotland, UK, 2011), 2017–2020. ACM, New York, NY, USA. doi:10.1145/2063576.2063879. http://doi.acm.org/10.1145/2063576.2063879.

MANNING, C. D.; RAGHAVAN, P.; AND SCHÜTZE, H., 2008. *Introduction to information retrieval*, vol. 1. Cambridge University Press.

MEDELYAN, O.; FRANK, E.; AND WITTEN, I. H., 2009. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, 1318–1327. Association for Computational Linguistics.

MENG, R.; ZHAO, S.; HAN, S.; HE, D.; BRUSILOVSKY, P.; AND CHI, Y., 2017. Deep keyphrase generation. *arXiv preprint arXiv:1704.06879*, (2017).

MERITY, S.; MURPHY, T.; AND CURRAN, J. R., 2009. Accurate argumentative zoning with maximum entropy models. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, 19–26. Association for Computational Linguistics.

MIHALCEA, R. AND TARAU, P., 2004. TextRank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; AND DEAN, J., 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, (2013).

MIZUTA, Y. AND COLLIER, N., 2004. 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*, 29–35. Association for Computational Linguistics.

MIZUTA, Y.; KORHONEN, A.; MULLEN, T.; AND COLLIER, N., 2006. Zone analysis in biology articles as a basis for information extraction. *International journal of medical informatics*, 75, 6 (2006), 468–487.

NAKOV, P. I.; SCHWARTZ, A. S.; AND HEARST, M., 2004. Citances: Citation sentences for semantic analysis of bioscience text. In *Proceedings of the SIGIR04 workshop on Search and Discovery in Bioinformatics*, 81–88.

NALLAPATI, R. M.; AHMED, A.; XING, E. P.; AND COHEN, W. W., 2008. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 542–550. ACM.

NANBA, H.; KANDO, N.; AND OKUMURA, M., 2011. Classification of research papers using citation links and citation types: Towards automatic review article generation. *Advances in Classification Research Online*, 11, 1 (2011), 117–134.

NASTASE, V. AND HITSCHLER, J., 2018. Correction of OCR word segmentation errors in articles from the ACL collection through neural machine translation methods. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

NAWAZ, R.; THOMPSON, P.; MCNAUGHT, J.; AND ANANIADOU, S., 2010. Meta-knowledge annotation of bio-events. In *LREC*.

NGUYEN, T. D. AND KAN, M.-Y., 2007. Keyphrase extraction in scientific publications. In *International conference on Asian digital libraries*, 317–326. Springer.

PAGE, L.; BRIN, S.; MOTWANI, R.; AND WINOGRAD, T., 1999. The pagerank citation ranking: Bringing order to the web. (1999).

PETERS, M. E.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K.; AND ZETTLEMOYER, L., 2018. Deep contextualized word representations. In *Proc. of NAACL*.

POLLARD, T. D., 2007. Regulation of actin filament assembly by arp2/3 complex and formins. *Annu. Rev. Biophys. Biomol. Struct.*, 36 (2007), 451–477.

QAZVINIAN, V. AND RADEV, D. R., 2010. Identifying non-explicit citing sentences for citation-based summarization. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, 555–564. Association for Computational Linguistics.

RADEV, D. R.; JOSEPH, M. T.; GIBSON, B.; AND MUTHUKRISHNAN, P., 2009a. A bibliometric and network analysis of the field of computational linguistics. *Journal of the American Society for Information Science and Technology*, (2009).

RADEV, D. R.; MUTHUKRISHNAN, P.; AND QAZVINIAN, V., 2009b. The acl anthology network corpus. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, 54–61. Association for Computational Linguistics.

RAJPURKAR, P.; JIA, R.; AND LIANG, P., 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, (2018).

RAVENSCROFT, J.; LIAKATA, M.; AND CLARE, A., 2013. Partridge: An effective system for the automatic cassification of the types of academic papers. In *Research and Development in Intelligent Systems XXX*, 351–358. Springer.

RAVENSCROFT, J.; OELLRICH, A.; SAHA, S.; AND LIAKATA, M., 2016. Multi-label annotation in scientific articles-the multi-label cancer risk assessment corpus. In *LREC*.

RITCHIE, A., 2009. Citation context analysis for information retrieval. Technical report, University of Cambridge Computer Laboratory.

RITCHIE, A.; ROBERTSON, S.; AND TEUFEL, S., 2008. Comparing citation contexts for information retrieval. In *Proceedings of the 17th ACM conference on Information and knowledge management*, 213–222. ACM.

RITCHIE, A.; TEUFEL, S.; AND ROBERTSON, S., 2006a. Creating a test collection for citation-based IR experiments. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 391–398. Association for Computational Linguistics.

RITCHIE, A.; TEUFEL, S.; AND ROBERTSON, S., 2006b. How to find better index terms through citations. In *Proceedings of the workshop on how can computational linguistics improve information retrieval?*, 25–32. Association for Computational Linguistics.

ROBERTS, R. J., 2001. PubMed Central: The GenBank of the published literature. (2001).

ROBERTSON, S.; ZARAGOZA, H.; ET AL., 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3, 4 (2009), 333–389.

SANDERSON, M., 2010. *Test collection based evaluation of information retrieval systems*. Now Publishers Inc.

SAWILOWSKY, S. S., 2009. New effect size rules of thumb. (2009).

SCHÄFER, U. AND KASTERKA, U., 2010. Scientific authoring support: A tool to navigate in typed citation graphs. In *Proceedings of the NAACL HLT 2010 workshop on computational linguistics and writing: Writing processes and authoring aids*, 7–14. Association for Computational Linguistics.

SHOTTON, D., 2010. Cito, the citation typing ontology. *Journal of biomedical semantics*, 1, Suppl 1 (2010), S6.

SILVA, C. AND RIBEIRO, B., 2003. The importance of stop word removal on recall values in text categorization. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, vol. 3, 1661–1666. IEEE.

SMALL, H., 1973. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the Association for Information Science and Technology*, 24, 4 (1973), 265–269.

SODERLAND, S., 1999. Learning information extraction rules for semi-structured and free text. *Machine learning*, 34, 1-3 (1999), 233–272.

STROHMAN, T.; METZLER, D.; TURTLE, H.; AND CROFT, W. B., 2005. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, vol. 2, 2–6.

SWALES, J., 1986. Citation analysis and discourse analysis. *Applied linguistics*, 7, 1 (1986), 39–56.

TANG, J. AND ZHANG, J., 2009. A discriminative approach to topic-based citation recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 572–579. Springer.

TEUFEL, S., 2000. *Argumentative zoning: Information extraction from scientific text*. Ph.D. thesis, University of Edinburgh.

TEUFEL, S., 2010. *The structure of scientific articles: Applications to citation indexing and summarization*. Center for the Study of Language and Information.

TEUFEL, S.; CARLETTA, J.; AND MOENS, M., 1999. An annotation scheme for discourse-level argumentation in research articles. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, 110–117. Association for Computational Linguistics.

TEUFEL, S. AND MOENS, M., 2002. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, 28, 4 (2002), 409–445.

TEUFEL, S.; SIDDHARTHAN, A.; AND BATCHELOR, C., 2009. Towards discipline-independent argumentative zoning: evidence from chemistry and computational linguistics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, 1493–1502. Association for Computational Linguistics.

TEUFEL, S.; SIDDHARTHAN, A.; AND TIDHAR, D., 2006. Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 103–110. Association for Computational Linguistics.

THOMPSON, P.; NAWAZ, R.; MCNAUGHT, J.; AND ANANIADOU, S., 2011. Enriching a biomedical event corpus with meta-knowledge annotation. In *BMC Bioinformatics*.

TURNEY, P. D., 1999. Learning to extract keyphrases from text. *arXiv preprint cs/0212013*, (1999).

TURTLE, H.; HEGDE, Y.; AND ROWE, S., 2012. Yet another comparison of Lucene and Indri performance. In *SIGIR 2012 Workshop on Open Source Information Retrieval*, 64–67.

VALENZUELA, M.; HA, V.; AND ETZIONI, O., 2015. Identifying meaningful citations. In *AAAI Workshop: Scholarly Big Data*.

WAARD, A.; BUITELAAR, P.; AND EIGNER, T., 2009. Identifying the epistemic value of discourse segments in biology texts (project abstract). In *IWCS*.

WANG, Y.; WANG, L.; LI, Y.; HE, D.; CHEN, W.; AND LIU, T.-Y., 2013. A theoretical analysis of ndcg ranking measures. In *Proceedings of the 26th Annual Conference on Learning Theory (COLT 2013)*.

WHITE, H. D., 2004. Citation analysis and discourse analysis revisited. *Applied linguistics*, 25, 1 (2004), 89–116.

WILBUR, W. J. AND SIROTKIN, K., 1992. The automatic identification of stop words. *Journal of information science*, 18, 1 (1992), 45–55.