Natural Language Processing and Information Retrieval Methods for Intellectual Property Analysis

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart zur Erlangung der Würde eines Doktors der Philosophie (Dr. phil.) genehmigte Abhandlung

Vorgelegt von

Charles Jochim

aus Bloomington, Indiana (USA)

Hauptberichter:	Prof. Dr. Hinrich Schütze
Mitberichter:	Prof. Dr. Thomas Ertl
Mitberichter:	Prof. Dr. Leo Wanner

Tag der mündlichen Prüfung: 16. Dezember 2013

Institut für Maschinelle Sprachverarbeitung
der Universität Stuttgart

2014

Abstract

More intellectual property information is generated now than ever before. The accumulation of intellectual property data, further complicated by this continued increase in production, makes it imperative to develop better methods for archiving and more importantly for accessing this information. Information retrieval (IR) is a standard technique used for efficiently accessing information in such large collections. The most prominent example comprising a vast amount of data is the World Wide Web, where current search engines already satisfy user queries by immediately providing an accurate list of relevant documents. However, IR for intellectual property is neither as fast nor as accurate as what we expect from an Internet search engine.

In this thesis, we explore how to improve information access in intellectual property collections by combining previously mentioned IR techniques with advanced natural language processing (NLP) techniques. The information in intellectual property is encoded in text (i.e., *language*), and we expect that by adding better language processing to IR we can better understand and access the data. NLP is a quite varied field encompassing a number of solutions for improving the understanding of language input. We concentrate more specifically on the NLP tasks of statistical machine translation, information extraction, named entity recognition (NER), sentiment analysis, relation extraction, and text classification.

Searching for intellectual property, specifically patents, is a difficult retrieval task where standard IR techniques have had only moderate success. The difficulty of this task only increases when presented with multilingual collections as is the case with patents. We present an approach for improving retrieval performance on a multilingual patent collection by using machine translation (an active research area in NLP) to translate patent queries before concatenating these parallel translations into a multilingual query.

Even after retrieving an intellectual property document however, we still face the problem of extracting the relevant information needed. We would like to improve our understanding of the complex intellectual property data by uncovering latent information in the text. We do this by identifying citations in a collection of scientific literature and classifying them by their citation function. This classification is successfully carried out by exploiting some characteristics of the citation text, including features extracted via sentiment analysis, NER, and relation extraction. By assigning labels to citations we can better understand the relationships between intellectual property documents, which can be valuable information for IR or other applications.

Zusammenfassung

Die Menge an Texten, die geistiges Eigentum beschreiben, wächst stetig. Um diese Masse an Informationen überschaubar zu machen, ist es notwendig, bessere Methoden zu entwickeln, um den Zugriff darauf zu vereinfachen. Information Retrieval (IR) ist eine Standardtechnik, um effizient Informationen aus großen Datenbanken abzurufen. Die wohl bekannteste Informationsquelle, deren Größe die effiziente Verarbeitung erschwert, ist das World Wide Web (WWW). Hierfür wurden Suchmaschinen entwickelt, die für von Benutzern gestellte Suchanfragen Listen relevanter Dokumente erstellen. IR für geistiges Eigentum ist jedoch im Vergleich langsamer und ungenauer als wir es von Suchmaschinen im WWW gewohnt sind.

Diese Dissertation befasst sich damit, wie mit einer Kombination von Methoden aus dem Information Retrieval und der natürlichen Sprachverarbeitung (Natural Language Processing, kurz NLP) der Zugang zu Textsammlungen geistigen Eigentums verbessert werden kann. Diese Kombination ist vielversprechend, da Informationen über geistiges Eigentum wie bereits erwähnt in Texten festgehalten werden (d.h., es handelt sich um natürliche *Sprache*). NLP ist ein komplexer Forschungsbereich, der ein breites Spektrum an Ansätzen bietet, um die Bedeutung sprachlicher Daten automatisch zu analysieren. Die in dieser Dissertation beschriebene Arbeit befasst sich mit statistischer maschineller Übersetzung, Informationsextraktion (genauer mit Named Entity Recognition (NER), Sentiment-Analyse und Relationsextraktion) und Textklassifikation.

Automatisches Suchen in Textsammlungen geistigen Eigentums, insbesondere Patentsuche, stellt eine besondere Herausforderung dar. Bisherige Ansätze unter Verwendung von IR-Standardtechniken waren daher nur wenig erfolgreich. Zusätzliche Schwierigkeiten treten auf, wenn mehrsprachige Textsammlungen durchsucht werden sollen, wie es in der Patentsuche oft der Fall ist. In dieser Arbeit wird ein Verfahren vorgestellt, um die Suche in einer mehrsprachigen Textsammlung von Patenten zu verbessern. Dies wird durch die Verwendung von maschineller Übersetzung erzielt, die auf die Suchanfragen angewendet wird, indem aus mehreren übersetzten Suchanfragen eine mehrsprachige Suchanfrage erstellt wird.

Die Extraktion von relevanten Informationen aus einem Dokument ist ein weiteres Problem, das dem Suchvorgang folgt. Um dieses Problem zu lösen, ist es notwendig, implizite Informationen in den Daten zu erkennen, um komplexe Zusammenhänge besser verstehen zu können. In dieser Arbeit wird dies durch die automatische Identifikation und Klassifikation von Zitaten in einer Textsammlung wissenschaftlicher Fachliteratur erreicht. Der vorgestellte Ansatz kombiniert dazu verschiedene Merkmale der Texte, die

unter Anderem durch Sentiment-Analyse, NER und Relationsextraktion automatisch erkannt werden. Durch die Klassifikation von Zitaten werden Zusammenhänge zwischen den Dokumenten ersichtlich, die für die Verbesserung von IR-Systeme und andere Anwendungen genutzt werden können können.

Acknowledgments

First, I would like to thank Markus Dickinson and Sandra Kübler for getting me started in computational linguistics.

I would like to thank everyone at IMS. The institute is full of people that have helped me along the way in one way or another, and I regret that I did not give back as much as I received. Among those at the IMS, I need to particularly thank Sabine Dieterle for making life easier for me in and out of IMS. Thanks to Hassan Sajjad for putting up with me for the first two years and for the engaging conversations about NLP and everything else. I am very grateful to Christina Lioma, who helped me work through my early papers and pushed me to think much more critically about the work I was doing. I would like to thank Alex Fraser for reading drafts of papers, discussing research problems, and giving advice (even if I was not one of his advisees). I am lucky to have had Hamid Kobdani, Florian Laws, and Lukas Michelbacher finish ahead me so that I could lean on them for guidance on how to finish the thesis, what it should look like, and how to survive the whole process in general. Thanks to the members of various reading groups (Thomas Müller, Anders Björkelund, Wolfgang Seeker, Alessandra Zarcone, Jason Utt, among others that I am sure I am forgetting) who put up with my naive questions. I am also grateful to have been thrown in with the Sentiment Analysis group so that I could benefit from all of their expertise: thanks to Wiltrud Kessler, Andrea Glaser, and Khalid Al Khatib.

I was also lucky to work with people at the Institute for Visualization and Interactive Systems (VIS). I enjoyed all the people I met and worked with there, in particular Florian Heimerl and Harald Bosch.

Two people that fall into a lot of the categories above, without whom completion of this thesis would have been much less certain, are Steffen Koch and Christian Scheible. I have run out of ways of thanking them over the years (not to mention the fact that I ran out of ways of saying "thank you" two paragraphs ago).

I would like to thank Leo Wanner for being on my thesis committee and for useful suggestions to improve this thesis. I am grateful that Thomas Ertl was on my committee and I am indebted to him for the position I had in the project with VIS.

Hinrich Schütze made all of the work in this thesis possible and gave me the opportunity to work with all the fantastic people mentioned above. I am lucky to have had him as my advisor and grateful for everything that he has done for me.

I am grateful to the DFG for funding this thesis under SPP 1335 *Scalable Visual Analytics*.

Finally, I could not have done this without the support of my family. My parents, sister, and grandfather were always a source of encouragement. I would not have finished this without the unwavering support from Elisabetta (in the face of my constant doubt). I am so thankful to her, Daniel, and Josephine for putting up with me and giving me something more to think about than F_1 scores or statistical significance.

Contents

List of Figures

List of Tables

List of Abbreviations

ACL	Association for Computational Linguistics
CLEF	Conference and Labs of the Evaluation Forum
CLIR	cross-language IR
EPO	European Patent Office
IP	intellectual property
IPC	international patent classification
IR	Information Retrieval
IRF	Information Retrieval Facility
LM	language modeling
MAP	mean average precision
MaxEnt	maximum entropy
MI	mutual information
MT	machine translation
NER	Named Entity Recognition
NLP	Natural Language Processing
P10	precision at 10
POS	part of speech
PRES	patent retrieval evaluation score
SMT	statistical machine translation
USPTO	United States Patent and Trademark Office
VIS	Institute for Visualization and Interactive Systems

Chapter 1

Introduction

1.1 Motivation

The term *information explosion* has been used to describe the ever-increasing creation and distribution of information over the last half-century (Nakagawa et al., 2008). The concept is often associated with the proliferation of the World Wide Web but it is not only there that we have seen this dramatic growth in information in recent decades. Processing, storing, and accessing the data we have produced until now is already a challenge, but as we continue to generate new information at ever-increasing rates, the development of scalable data processing techniques is even more imperative. With massive amounts of data available, we need tools capable of handling that data. Information retrieval (IR) techniques have so far been very successful in processing and accessing the data available on the Web so that user queries are immediately served with an accurate list of the most relevant documents. Other search domains present a different set of challenges for information retrieval. The Web, for example, has a huge contributor base and user base making solutions like the PageRank algorithm or user feedback mechanisms like click-through data highly effective so that the documents most likely to be relevant are returned first. These circumstances are not found in many other search scenarios, e.g., searching for a book in a library catalog or for a patent in a patent database. For more difficult search tasks, we should explore

other techniques for strengthening IR. In these cases, lacking click-through metadata or an extensive network of links, we must rely on extracting as much information out of the text as possible. For this reason, in this thesis we also rely on natural language processing (NLP) techniques. The field of NLP is quite varied, encompassing a number of different approaches to automatically analyze language in an effort to improve the understanding of it. A number of NLP techniques are already used in information retrieval, e.g., lemmatization or language modeling; we plan to extend the use of NLP in IR with other more complex approaches such as named entity recognition (NER), relation extraction, text classification, and machine translation. We are interested in applying these techniques, in particular, to access the information in patent and scientific literature collections. These data collections have many characteristics in common and present similar challenges for information access, ones that differ from web collections for example.

A patent, which is a right awarded to an inventor to protect their intellectual property in exchange for the public disclosure of the invention, should be novel and therefore unique. Patents are granted by a number of state patent offices throughout the world, e.g., United States Patent and Trademark Office (USPTO), European Patent Office (EPO), or Japan Patent Office (JPO). In the last few decades in particular, patent application numbers have seen significant growth as patents have become more ubiquitous around the world to protect intellectual property. Certain technological areas have even seen exponential growth. For example, Figure 1.1 shows the sharp rise in "computer technology" patents that have been granted in the last decade. Growth varies among patent offices as well; Figure 1.2 shows the steady growth in patent applications to the Chinese patent office, SIPO, and in 2011, for the first time, it accepted more patent applications than the USPTO. [1]

Scientific literature has seen similar increases. The earliest scientific writing of course dates back much further than the 1950s, but it is then that the study of scientific literature began in earnest with Eugene Garfield (1955) and the Institute for Scientific Information (ISI). The work led by Garfield

[1] http://www.wipo.int/export/sites/www/freepublications/en/intproperty/941/wipo_pub_941_2012.pdf

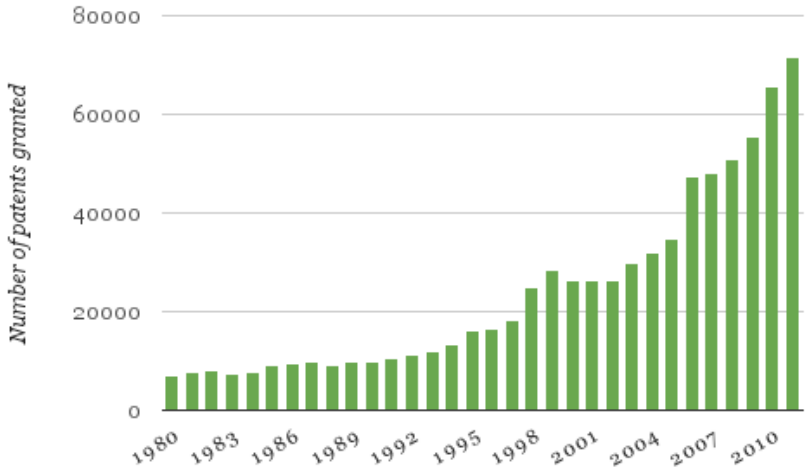

Figure 1.1: Computer technology patents granted worldwide. Source: WIPO statistics database. Last updated: March 2013.

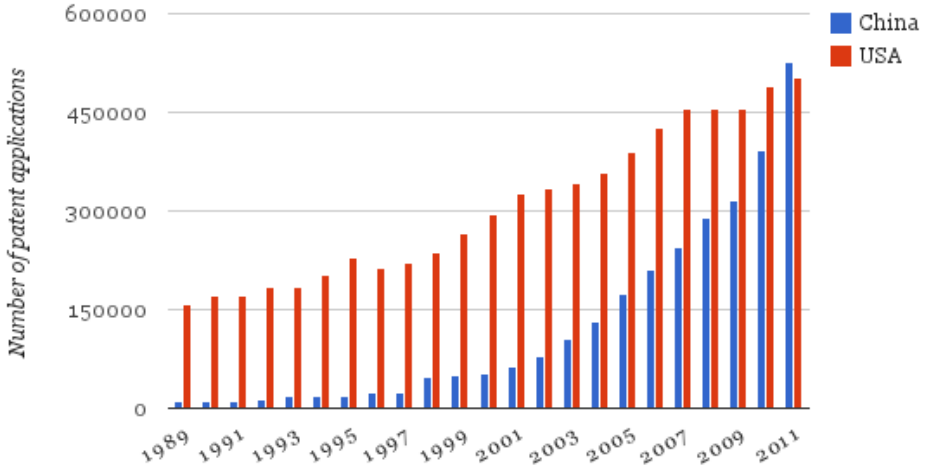

Figure 1.2: Total patent applications from China and the United States. Source: WIPO statistics database. Last updated: March 2013.

and others in information science can give us some indication of how scientific literature has grown in the last half-century. The initial ISI Science Citation Index (SCI) began with an index of 613 journals.[2] Now the *Web of Science*, which includes the SCI, comprises over 12,000 journals and 148,000 conference, workshop, and symposia proceedings totaling over 46 million records.[3]

With this information overload, new and better techniques must be developed to improve our access to this information. Information retrieval has already been very successful in doing this, in particular for the World Wide Web, but different data collections lead to different challenges for accessing the information. Of course ideal searching solutions differ for different mediums (e.g., image search vs. video search vs. text search), but even the strategies for accessing relevant text information in a web collection differ greatly from those for accessing a relevant patent or scientific article. Additional tools are needed to more efficiently gain access to this data. To this end, we propose combining the strength of information retrieval with natural language processing. NLP is already widely used to strengthen IR and in this thesis we present further evidence that some of the various techniques that fall under the umbrella of NLP can be used to improve information retrieval and information access.

Intellectual Property. We focus on IR and NLP techniques for *intellectual property*, specifically patent and scientific literature collections. These data collections share a number of characteristics that are important for our work.

First, the data contained in these collections is a mix of text, metadata, and images. We are primarily concerned with handling the text portions of the documents but valuable information is also contained in the images, figures, diagrams, and tables and in the various forms of metadata. We consider that the metadata in both patents and scientific literature can be explicit or implicit. Both contain explicit descriptive metadata, e.g., attribution of the

[2] `http://thomsonreuters.com/products_services/science/free/essays/50_years_citation_indexing/`

[3] The Web of Science is included, with other scientific databases and citation indexes, in the *Web of Knowledge*.

author and institution for scientific literature, or inventor and assignee for patents; or contain explicit structural metadata, e.g., sections in scientific literature or description and claim fields in patents. Explicit bibliographic metadata is also present in both patents and scientific articles. However, there is a wealth of additional information to be extracted from these documents. In both cases, bibliographic metadata can be enhanced by inspecting the textual context of a citation – something we consider to be implicit metadata. This implicit metadata can also be found in additional structural layers at finer levels of granularity than sections or numbered claims. Argumentative Zoning (Teufel, 1999) has been used to show the rhetorical structure of scientific literature for example. Patent descriptions also have a certain structure even if it is not always well-defined. For example, a figure is immediately followed by a detailed description of that figure (as might be expected). Some other patent structure is recommended but not strictly enforced. According to the regulations under the Patent Cooperation Treaty, the description should include the following: technical field, background art, disclosure of invention, description of drawings, best mode(s) for carrying out the invention, industrial applicability.[4] However, this is not an absolute requirement, and so a subset of these parts may appear in a given patent, with or without the appropriate subheadings. NLP, specifically information extraction, is commonly used in this situation to identify the various forms of metadata. Techniques like NER, relationship extraction, keyword extraction, or summarization can be applied to extract this metadata.

Second, we are dealing with highly technical content, which lends itself to the use of highly technical language. This results in a larger, highly-specific lexicon and often more complex syntactic constructions. In the case of patents, this is sometimes referred to as *patentese*, and while scientific literature does not have to be written in such a way as to withstand legal scrutiny, it is still written to withstand the scrutiny of peers. The language in both must be written in a clear unambiguous way: in the case of patents this serves to clarify, quite specifically, exactly what is the invention that must be protected; and in the case of scientific literature, the work must

[4] `http://www.wipo.int/pct/en/texts/rules/r5.htm#_5_1`

be understood by the academic community so that it could be replicated if desired and so that the correct conclusions are being drawn from the hypothesis, methodology, and results presented. IR models designed for *ad hoc* retrieval can be used effectively for many retrieval tasks, but for complex language like that found in patents and scientific literature they have proved to be somewhat lacking. Therefore, including the more complex models of language available in NLP is crucial to improvement over IR alone.

Third, these collections are quite sparse in the sense that usually very few documents will satisfy the information need of the users. This contrasts with queries on the Web, for example, where often times many documents satisfy the user's needs (albeit some better than others). Patent *prior art* or *infringement* searches can even boil down to the search for a single relevant patent document. If you consider that each patent should be a unique invention – one requirement for patentability is that the invention is novel – then there should be no redundancy in a patent collection (which again differs from the Web). Publishable scientific literature should also generally present some novel idea and so again we can assume that each article in a collection contains some information that cannot be found elsewhere in the collection. The sparsity of the collections is tied to, and further complicated by, the fact that our objective is to obtain high recall. Patent search is often the first example given for a recall-oriented search task (Magdy, 2012). In the case of prior art search, for example, if a relevant document exists but is not found, the applicant is open to an infringement lawsuit. Identifying relevant documents in scientific literature may be equally important (although likely not as costly). The expectation when submitting a scientific article for publication is that the relevant literature has been reviewed by the author and the most pertinent (and often the most recent) of that literature is summarized in some sort of "related work" section. The combination of the sparsity of relevant documents in patent and scientific literature collections and the importance of recovering those rare documents, makes IR for patents and scientific literature a particularly difficult task. Here too, it should be evident that this is a logical application for NLP. A number of solutions for improving recall rely on linguistic manipulations: stemming or lemmatization; expanding queries

with synonyms; or expanding result sets with clusters of semantically similar documents. Because intellectual property text is so complex, it is critical to have a more detailed and accurate analysis of the language so that we can find these very specific documents.

Strengthening IR with NLP. Standard IR techniques have already been applied to patent and scientific literature collections with varying success. Many approaches rely on a standard inverted index of terms or leverage the citation information (e.g., graph-based co-citation approaches). Because of the facts that (i) the novel information is encoded in "natural language" and (ii) making sense of the document using only a weighted term vector is difficult, we argue for including more powerful text data mining and NLP techniques into the information access process.

There is already significant overlap of methods used in both NLP and IR and we will not summarize them all here. However, some prominent examples are:

- preprocessing techniques used when building an IR index are the same as those used for NLP, e.g., tokenization or lemmatization;
- language modeling, an active area of NLP research, forms the basis of a popular probabilistic IR approach;
- NLP and IR can even be found in a number of the services already offered by leading search engines (e.g., question answering ("How old is Vanna White?") or the directions generated from "how do I get from tulsa to tucson?"; see Figure 1.3).

Despite this overlap the use of NLP in IR is still limited, primarily using shallow NLP techniques (Brants, 2003; Schütze, 2010).

In the data collections that we are working with, the information contained can be presented in figures, tables, images, and metadata, but more than anything it is found in the text. Because the language in these documents is rather complex we need natural language processing tools to gain

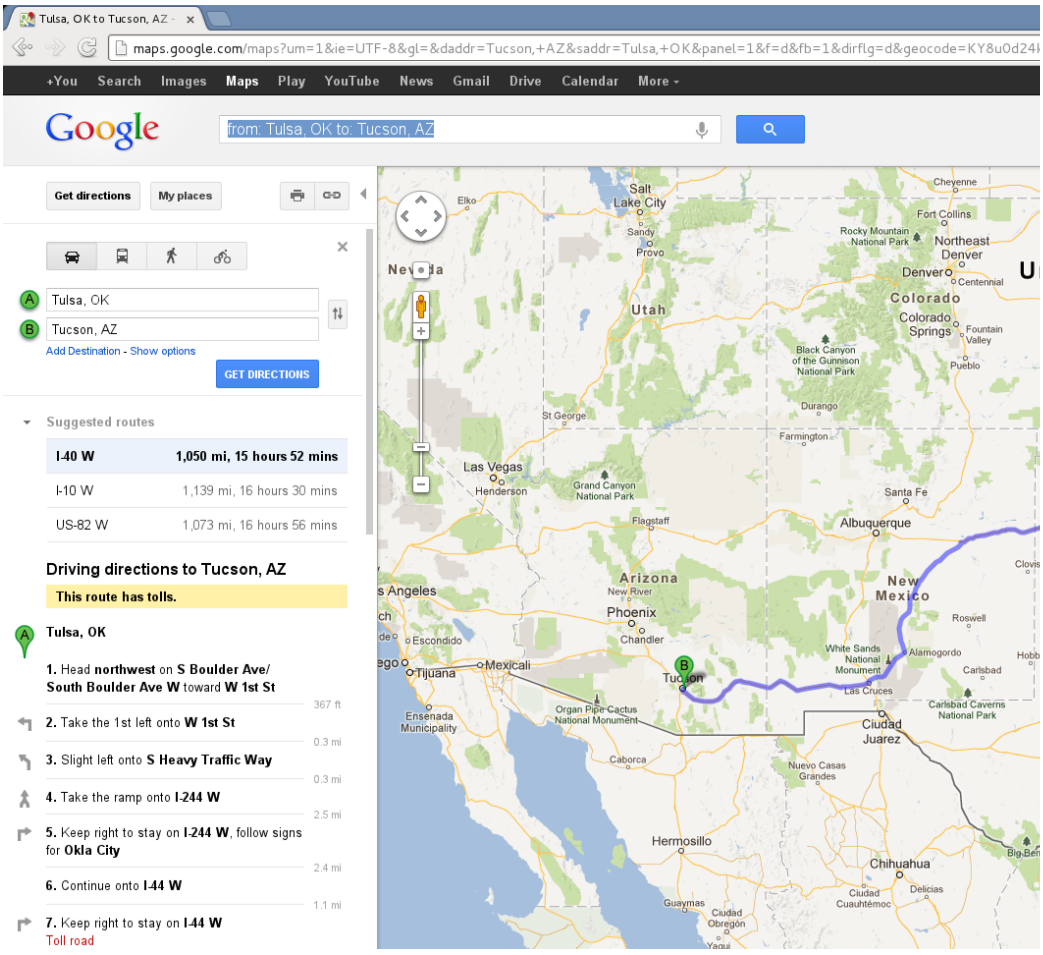

Figure 1.3: First result for query, “how do I get from tulsa to tucson?”

access to the data that is encoded in natural language. For example, standard shallow NLP techniques like part-of-speech tagging can be applied in a number of ways to improve information retrieval (see Lioma, 2008). If we use deeper NLP techniques we can make better sense of both the query and the indexed document to more accurately represent the user's information need.

Because language is used to express the information need and also represent the information in the collection, we can leverage NLP at different stages of the retrieval process. In this thesis, we will see examples of different ways that NLP techniques can be applied at these various stages. More specifically, NLP may be applied to the document or to the query. If it is applied to the document then the output of the NLP system can be included in the index, e.g., standard applications like indexing lemmatized word forms or perhaps indexing NER output to distinguish "New York" in "New York City" from "New York Mets". Applying NLP to the query can be difficult because of constraints on the query response time and also deep NLP techniques are often of limited use on a list of different keywords, i.e., not natural language. Some of the preprocessing mentioned above should be done both when indexing and querying (e.g., so that a lemmatized query matches a lemmatized document). In other cases, there is a choice whether to perform the NLP task on the queries or on the documents. In an example pertinent to this thesis, cross-language IR, one can translate an entire collection and index the translated documents or only the query can be translated at query time. Translating the entire collection (assuming the collection is small enough for this to be reasonable) is appealing because it can be done once, offline, and then indexed, avoiding translation with each submitted query. The translation would likely be more accurate as well, as you would have more context and (presumably) well-formed sentences which lends itself to better machine translation (whether it be phrase-based, n-gram-based, or hierarchical). On the other hand, for research purposes this solution is not as practical, because after making changes to the machine translation you then need to translate and index the entire collection again; if the collection is very large this solution is not feasible. Handling ambiguities in the queries can also be handled nicely when translating at query time; multiple possible translations can be

included in the query (optionally with the probability of their translation if available). See Oard and Diekema (1998) for a detailed discussion of the advantages and disadvantages of document translation and query translation. We choose to apply machine translation to the queries directly (see Chapter 2 and Chapter 3) to avoid translating a very large collection and to more easily test different translation alternatives.

In Chapters 2 and 3 we apply NLP (e.g., machine translation) to the queries, while in Chapter 4 we use NLP techniques directly on the documents. We use text mining and classification approaches to enrich the document's metadata (which can subsequently be indexed).

The fact that IR and NLP are related – and that they even overlap – is widely accepted (e.g., the two major textbooks in NLP, Jurafsky and Martin (2000) and Manning and Schütze (1999), both include chapters on IR), but the role that NLP can play in IR is still an open question. There already exists a large body of literature in IR which relates to NLP topics, just as there is NLP literature relating to IR.[5] In this thesis we too investigate how the two disciplines can be combined, specifically for analyzing and accessing intellectual property information.

Putting it all together. The work in this thesis was undertaken within the framework of a project on scalable intellectual property analysis. The task of accessing the information encoded in intellectual property is a difficult one; to tackle this problem we employ different tools that have been successful in other data analysis tasks: natural language processing, information retrieval, and visual analytics. Within the larger project, we aim to unlock the information in intellectual property collections by combining these three tools. In this thesis, we focus specifically on NLP and IR.

Our experiments are conducted on two intellectual property subareas: patents and scientific literature. The first may be of wider interest because of the importance to industry and research and development, while the second is of increasing interest in academic institutions (as a part of larger compre-

[5]To further this point, the call for papers at SIGIR 2013 includes "NLP for IR" while the call for papers at ACL 2013 includes "Information Retrieval".

hensive methods for evaluating the output of individual personnel, research groups, or departments). The latter also has the advantage that we are better equipped to evaluate the effectiveness of our own solutions.

Throughout the thesis we explore ways of combining IR and NLP, e.g., by using statistical machine translation in an IR system, or by extracting metadata using text classification for later indexing in a retrieval system. This has all been done in parallel with visual analytic research to take advantage of the strengths of both textual and visual analytics. The visual analytic innovations are beyond the scope of this thesis, but we do illustrate in a later appendix how we have used these two complementary techniques so far, and propose further possibilities for their collaboration.

1.2 Contributions

The goal of our work, culminating in this thesis, was to improve the data access in highly technical text collections. This has been done by exploring techniques that are used in the fields of IR and NLP. The specific contributions of this thesis are the following:

- We use a machine translation system for multilingual IR that has been trained directly on the IR collection. To the best of our knowledge, we were the first to use the parallel portions of a multilingual IR collection to build translation dictionaries. We subsequently used those dictionaries to translate queries for multilingual IR. This contribution is covered in (Jochim et al., 2010) and (Jochim et al., 2011).

- We show that translating and expanding queries in a multilingual IR setting improves recall measures for patent retrieval (a recall-oriented task). We compare different translation solutions and conclude that using a patent-specific translation dictionary, built directly from the parallel portions of the patents, yields superior results than a generic translation dictionary. This contribution is covered in (Jochim et al., 2010).

- We confirm that using phrase translations selectively can improve patent retrieval. Our comparison of term and phrase translations for multilingual queries shows that term translations may be more consistent but that phrase translations, while volatile, can lead to better results. We also look into the various effects of term and phrase translation on French and German language queries. This contribution is covered in (Jochim et al., 2011).

- We show that additional *descriptive* metadata like citations can easily be extracted from scientific literature and that we can automatically classify the citations' function. This classification makes use of the lexical features in a citation's local context and relation information extracted from that context (e.g., the relation between the citing and the cited articles). See Chapter 4 for a complete description of features. Automatically labeling citation function can be used for summarization, information retrieval, bibliometric and information science measures, among other applications. This contribution is covered in (Jochim and Schütze, 2012).

- We show, with the help of our visualization partners, that the combination of IR, NLP and visual analytics can improve information access, in particular for difficult collections such as patents and scientific literature, where data is encoded primarily in text but also as graphic data and metadata. This contribution is touched on in (Jochim et al., 2010) and (Heimerl et al., 2012a).

1.3 Structure

This thesis contains five chapters and two appendices. We include here a short introduction to each:

Chapter 2 begins our investigation into multilingual patent retrieval. Our goal is to compare the effects of using more or less complete translation dictionaries and to improve *prior art* search on a multilingual patent collection using query translation and expansion with these dictionaries. We

build our own translation dictionary from the parallel patent collection corpus and test its translations against those of a domain-free dictionary. We use a standard multilingual patent collection for conducting our experiments and show that the domain-specific translation dictionary that we compile outperforms the domain-free dictionary and also performs well in conjunction with traditional query expansion techniques.

Chapter 3 continues our work on translating and expanding patent queries by using phrase translation along with term translation. The objective is still to improve prior art search results, in particular for recall. We again build a patent-specific translation dictionary using the parallel portions of the patent collection, however, in this chapter we build a phrase dictionary and incorporate phrase translations in the information retrieval system. We use the same standard multilingual patent collection and run experiments to compare queries using term translation to queries that also use phrase translation. We show that phrase translation is more volatile than term translation, but that for some queries it has potential to improve results. We also show that term and phrase translation differently help German and French queries.

Chapter 4 presents our work in classifying citations in scientific literature. In previous chapters we focused on using NLP techniques to improve querying an intellectual property collection; in this chapter we use NLP to extract latent metadata that can potentially be used in an IR system, among other applications. This chapter introduces the utility of *citation function*, and the importance of and difficulty in automatically classifying citation function. We present a detailed investigation of features for citation classification. Finally, our results confirm that this is a difficult task but that with further investigation of useful features improved classification accuracy is attainable.

Chapter 5 summarizes the contributions of this thesis and presents some of the possible future research directions that can be pursued to extend this work.

Appendix A gives a short introduction to visualization and visual analytics for IR and NLP, and presents some of the work in this direction from collaboration with the Institute for Visualization and Interactive Sys-

tems (VIS) at the University of Stuttgart. The work in this thesis is part of the Schwerpunktprogramme (SPP) 1335, Scalable Visual Analytics: Interactive Visual Analysis Systems of Complex Information Spaces, funded by the Deutsche Forschungsgemeinschaft (DFG). This SPP project encourages the application of visual analytics to text data. Consequently, the use of visualization is a common thread that runs throughout the thesis, but the research questions posed in this thesis are not directed at visualization.

Appendix B is a supplement to Chapter 4. It includes the annotation guidelines used for annotating our citation corpus along with some additional information related to cue phrase features, i.e., we provide the list of automatically extracted cue phrases using mutual information.

Chapter 2

Building a Dictionary for Patent Query Translation

In this chapter we begin our exploration of intellectual property, specifically patents, using IR and NLP. Patent IR is a difficult retrieval task in particular when dealing with patents in multiple languages. By leveraging NLP techniques, e.g., machine translation, we hope to improve results on a multilingual patent collection. This chapter is organized as follows. We start by motivating the use of machine translation techniques for multilingual patent retrieval in Section 2.1. We continue in Section 2.2 by presenting the methodology of our proposed query translation approach. Section 2.3 describes and discusses the experimental evaluation of our approach. Section 2.4 describes some related work in patent IR, cross-lingual and multilingual IR, and the use of query translation and query expansion in IR. Finally, we summarize the chapter in Section 2.5 and lead in to Chapter 3 where this approach is extended and we explore the challenges of using phrase translations.

2.1 Motivation

Patent IR, also referred to as *patent retrieval* or *patent search*, is a specialized branch of IR that aims to support patent professionals in retrieving patents that satisfy their information needs and search criteria (Tait, 2008).

Patent retrieval is generally considered to be a difficult task (Tait, 2008, 2009). The vocabulary used in patents – which has been referred to as *patentese* (Atkinson, 2008) – is one source of difficulty because it often contains highly specialized or technical words not found in everyday language. The structure of patents can also lead to difficulties; patents are structured documents that contain several different fields, such as *description*, *claims*, or *prior-art*. The text in these fields is built over time, may not necessarily be in logical sequence (Atkinson, 2008), and can be partially translated into one or more different languages (e.g., English, French, and German, in the case of patents from the European Patent Office (EPO)). Additional difficulty in patent retrieval stems from the frequently intentional obfuscation of content by patent writers who wish to make their patents difficult to retrieve. This exacerbates the retrieval problem and can throw off robust standard IR approaches and systems (Azzopardi et al., 2010). Overall, the above difficulties mean that processing patent text is an open and challenging problem.

A common scenario in patent retrieval is *prior-art retrieval*, which is performed by patent searchers to determine the novelty of a new invention (Tiwana and Horowitz, 2009). One difficulty in this scenario is that patent searchers require comprehensive knowledge of all related and relevant patents. Overlooking a single valid patent could lead to detrimental and very expensive implications, such as infringement and litigation. In practice, this means that *recall* is very important for prior-art retrieval (Magdy and Jones, 2010). In addition, the increasingly large amounts of patent data available for retrieval, combined with the frequent and deliberate obfuscation of patent content, create a need for increased *precision* in retrieval.

In this work, we ask whether we can improve the precision and recall of patent retrieval, and more specifically of prior-art retrieval, by query translation. We reason that, since patents are partially translated into one or more languages, a collection of patents can be seen as a multilingual corpus that contains multiple languages across documents (e.g., some patents are written in French, others in English, and still others in German) and also within documents (e.g., a patent originally written in English can contain sections which are translated into French and German). Given such a multilingual

patent collection, we propose to expand queries using translations of the original query terms. Our goal is to create multilingual queries, in line with the multilingual patents available for retrieval. Our intuition is that within a multilingual collection, queries in more than one language may be useful for retrieval.

Although the term *cross-language IR* (CLIR) often subsumes all IR tasks involving more than one language, we will use cross-lingual IR to describe instances where the query is in one language and the collection is in a second language and use multilingual IR to describe any remaining retrieval scenario involving two or more languages, e.g., retrieval using a multilingual patent collection. In the work we present here, we choose to expand queries with translated terms resulting in multilingual queries, as opposed to replacing the original query terms with their respective translations (cf. cross-lingual IR). This type of query building can also be seen as a form of query expansion, because the queries are expanded with their respective translations.

We tackle query translation using a dictionary-based approach, where query term translations are fetched from a translation dictionary. We expect that the more accurate the translation, the better the retrieval performance. Our hypothesis is that a domain-specific translation dictionary on patents will give more accurate translations and hence better retrieval performance than a general domain-free translation dictionary, because the former will have better coverage of patent domains than the latter. However, maintaining a domain-specific patent translation dictionary is neither trivial nor always feasible: dictionary coverage is affected by the various different and dynamically changing patent subdomains, where even coining entirely novel concepts is not unusual. An additional drawback to static dictionaries is their weakness to deal with the ambiguous language often used by patent writers to deliberately obfuscate details of their patents. To address these points, we propose extracting a domain-specific translation dictionary from the patent collection used for retrieval. We do so by taking advantage of the parallel translations existing between parts of patents in the collection. Specifically, we identify such parallel translations, align them, and compute the translation probabilities between terms in the aligned translations. These

translations constitute the entries in our domain-specific patent translation dictionary.

2.2 Methodology

As previously stated, our goal is to increase patent retrieval performance, i.e., recall and precision, by expanding queries with translations in a multilingual search scenario. In this chapter, we focus on using term translations taken from a bilingual dictionary. Specifically, we compare the effectiveness of different translation dictionaries on the multilingual retrieval performance. We have chosen two dictionaries to compare: one, an off-the-shelf, domain-free dictionary; and the other a domain-specific dictionary extracted from a parallel corpus of patents. In describing our methodology, we begin with the extraction of a domain-specific patent translation dictionary from the patent claims (Section 2.2.1); then illustrate how the patent queries are translated and expanded using a translation dictionary (Section 2.2.2); and finally, compare the coverage of the two translation dictionaries (Section 2.2.3).

To evaluate our query translation, we conduct experiments separately with the general domain-free translation dictionary and the domain-specific translation dictionary that we extract from the patent collection. In addition, because our query translation can also be seen as a form of query expansion, we conduct experiments with a standard statistical query expansion technique (Rocchio, 1971).

2.2.1 Extracting a Patent-Specific Translation Dictionary

The retrieval collection we use throughout this work is comprised of patents granted by the EPO. When a patent is granted, the EPO provides manual translations of the patent claims in each of its three official languages, i.e., granted patent's claims appear in English, French and German.[1] We use these

[1]When abbreviated, we use the ISO 639-1 codes, EN, FR, and DE, respectively.

parallel translations of the claims to extract bilingual dictionaries for each language pair.

In order to extract a bilingual translation dictionary from the patent claims, we need to align the parallel translations of the claims, and then estimate translation probabilities for pairs of terms from the source and target language.

Aligning the parallel translations of the patent claims is not straightforward. The patent claims are guaranteed to be aligned for each language but we need aligned terms for our translation. Patent claims are very particular in that they are usually composed of a single sentence; however this single sentence can often be 100-200 words long, with some upwards of 600 words in length. These long sentences are usually composed of long lists of modifiers that "can run for many pages of six-point text" (Atkinson, 2008)). Term alignment is typically done on a sentence basis, but aligning terms in very long sentences becomes prohibitive and so sentences must be broken up into smaller segments that then need to be aligned. Different heuristics may be used to automatically divide large sentences into clauses, e.g., splitting sentences by punctuation. However, punctuation varies between the three languages, and so we choose to split sentences by the XML markup found in the patent documents. Since we now have (potentially non-aligned) clauses instead of actual sentences, we need a sentence aligner that performs well with sentences and clauses as input. We use the freely-available Gargantua sentence aligner[2], which has a reported F_1 measure of 0.98 in sentence alignment (Braune and Fraser, 2010).

Additionally, we conduct a manual evaluation aided by the developer of Gargantua, an expert in sentence alignment. We evaluate Gargantua's accuracy on the patent clauses using 2898 sentences from randomly chosen patents in the German-English parallel patent claims. The sentence alignment returned from Gargantua is manually edited to create a small gold standard for patent clause alignment. In two different evaluations, testing Gargantua against this gold standard has given $F_1 = 0.98$ and 0.99 respectively.

[2] `http://sourceforge.net/projects/gargantua/`

Source-Target Language	dict.cc	PatDict
EN-FR	2,950	521,387
EN-DE	109,961	532,042
FR-EN	2,913	467,176
FR-DE	7,338	466,435
DE-EN	124,596	1,461,929
DE-FR	8,743	1,794,897
$\sum$	256,501	5,243,866
Avg. translations per entry	2.00	9.31
Pct. overlapping terms	22.37%	1.09%

Table 2.1: Statistics of our translation dictionaries: the number of terms in each of the six source-target language pairs, the sum of those six numbers ($\sum$), the number of translations per entry (averaged over all pairs) and the percentage of overlapping terms (i.e., 22.37% of terms in dict.cc are also found in PatDict).

Using the aligned (sub-)sentences we can compute the translation probabilities between pairs of source-target language terms in the aligned patent claims using the freely-available GIZA++ toolkit (Och and Ney, 2003). For each language pair we run GIZA++ in both translation directions, i.e., for languages ℓ_1 and ℓ_2, we run GIZA++ with ℓ_1 as the source language and ℓ_2 and the target, and vice versa. Our GIZA++ training consists of four HMM iterations, five IBM Model 1 iterations, and ends with four iterations of IBM Model 4. GIZA++ produces a number of files useful for building a machine translation system; we are specifically interested in the table of translation candidate terms and their probabilities, which makes up our domain-specific translation dictionary for patents (*PatDict* henceforth). Even though patents encompass a number of subdomains (e.g., the International Patent Classification (IPC) is subdivided into eight sections, A-H[3]), we consider PatDict to be domain-specific to patents, in the sense that it covers solely the patent domain.

[3]Human necessities (A), Performing operations; transporting (B), Chemistry; metal-

2.2.2 Translating Queries with a Bilingual Dictionary

Our initial approach is to translate queries term by term using a bilingual translation dictionary. In fact, we test two dictionaries: we compare the PatDict dictionary described above, to a publicly available, domain-free dictionary, dict.cc.[4] Dict.cc is an online collection of bilingual dictionaries that contains all three of the language pairs found in PatDict. A summary of each of the translation dictionaries is given in Table 2.1.

Given a query Q in its original language (ℓ_1), our aim is to expand it with translations of the original query terms. As shown in Algorithm 1, for each term $t \in Q$ we select a single translation t' from the bilingual dictionary, and we expand the original query with it. We select the single best translation t' from the dictionary, where we define as single best the translation with the highest probability. If t is not covered in the dictionary or if t' is a stopword,[5] no translation takes place. We repeat this for all language combinations. At the end of this process, our new translated and expanded query Q' is the union of the original query terms and their single best available translations.

Algorithm 1 Query translation

```
Q' ← Q
ℓ1 ← LANGUAGE(Q)
for all t ∈ Q do
  for all ℓ2 ∈ {EN,FR,DE} where ℓ1 ≠ ℓ2 do
    t' ← TRANSLATE(t, l1, l2)
    if t' ∉ STOPLIST then
      Q' ← Q' + t'
    end if
  end for
end for
```

We select translations according to the translation probabilities stored in the dictionary. PatDict contains the translation probabilities estimated by GIZA++, but this is not the case for dict.cc. To add translation prob-

lurgy (C), Textiles; paper (D), Fixed constructions (E), Mechanical engineering; lighting; heating; weapons; blasting (F), Physics (G), Electricity (H).

[4] `http://www.dict.cc/`

[5] We use the default stopword lists in Apache Lucene for each of the three languages.

Source-Target Language	dict.cc		PatDict	
	Total	Per Query	Total	Per Query
EN-FR	11,154	56.9	140	0.7
EN-DE	2317	11.8	187	1.0
FR-EN	715	47.7	106	7.1
FR-DE	626	41.7	106	7.1
DE-EN	3595	40.4	247	2.8
DE-FR	4787	53.8	186	2.1
All	23,194	38.7	972	1.6

Table 2.2: Query terms (from the whole query set used in this work, described in section 2.3.1.2) that have no translation in dict.cc and PatDict. For example, there are 626 terms in French queries which have no German translation in dict.cc.

abilities to dict.cc, we first compute the frequency of each word in dict.cc using the respective English, French, and German language corpus.[6] Then we estimate the translation probability $p(f|e)$ using the maximum likelihood estimate (MLE). We use Wikipedia[7] as our corpus because it is more domain-independent, similar to dict.cc.

2.2.3 Dictionary Coverage and Translation Selection

The term coverage and the translation probabilities differ between our domain-specific PatDict and the domain-free dict.cc. PatDict has better coverage as can be seen in Table 2.2 with many fewer query terms without translations. We expect that this should in turn give more complete translations of the queries. However, better coverage alone does not necessarily mean more accurate translation; working with a large number of low probability translations can lower translation accuracy (hence retrieval effectiveness) and increase computational costs (Wang and Oard, 2006). This is why translation proba-

bilities are also needed.

In our description of the translation process in the previous section, we only describe using the single best translation, however, this can easily be extended to handle the n best translations. Alternatively, we could select only those terms that have a sufficiently high translation probability by setting a probability threshold, θ, such that each individual translation candidate must exceed θ. A cumulative probability threshold could also be used, where translations are included (in descending order of probability) and their probabilities summed until this cumulative probability threshold is reached (see Darwish and Oard, 2003). The threshold value could then be determined on the basis of either translation accuracy or retrieval performance. A further alternative, shown to be effective by Darwish and Oard (2003), is to consolidate translation probabilities from various resources, i.e., combining our two translation dictionaries and renormalizing their respective translation probabilities.

Moreover, tokenizing the queries and conducting term-based translation is not the only possible option. An interesting alternative would be to do phrase-based translation, in order to capture any non-compositional semantics in the queries that may be lost in term-based approaches. This alternative is discussed in detail in Chapter 3.

2.3 Experiments

2.3.1 Settings

2.3.1.1 Retrieval dataset

The experiments are conducted using the CLEF-IP 2010 collection (84GB) (Roda et al., 2009), a subset of the Matrixware Research Collection, provided by the IRF.[8] The collection contains 2.7 million EPO patent documents from 1985-2002, covering 1.3 million separate patents in English (69%), French

[6]Dictionary terms not found in the corpus are assigned $freq = 1$.
[7]We use an English Wikipedia dump from 1 February 2009.
[8]`http://www.ir-facility.org/`

(7%) and German (24%). Patents are roughly comprised of textual data, bibliographic metadata, and drawings. In this paper, we ignore metadata fields like inventor, applicant, publication date, and International Patent Classification (IPC), and we use only the text fields for retrieval: title, abstract, description, and claims. Of these four text fields, we draw attention to the abstracts, which we use for queries (described below), and to the claims, which we use for creating the patent translation dictionary (described in Section 2.2.1).

The CLEF-IP 2010 collection contains a training set with 300 queries (or *topics*) and their respective relevance assessments, for the CLEF-IP prior-art retrieval task.[9] Topics for CLEF-IP prior-art retrieval are not provided in the form of predefined keywords and/or phrases, like in other standard test collections, but instead as pointers to a patent file. Hence, an extra processing step is needed to generate queries from the patent documents (described in in the following section). Overall, out of the 300 queries, 196 are English, 89 are German, and 15 are French. Table 2.3 displays the statistics of this collection per language. Overall, the majority of the data available is in English, followed by German, and then French. This means that when analyzing differences in retrieval performance between languages we need to look at several possible factors: both different linguistic properties and different per-language query and document statistics could be the cause. Keep in mind also that relevant patents to a French query can be (and often are) in English and German, and vice versa, e.g., 15 French queries have 202 relevance assessments, of which 79 are English, 74 are French, and 49 are German.

2.3.1.2 Query creation

There are a number of ways that queries have been generated from patent documents (Graf et al., 2009; Xue and Croft, 2009; Magdy et al., 2011; Mahdabi et al., 2011). In this work, we concentrate on creating queries from the abstract of the patent documents. Xue and Croft (2009) showed that in

[9]The set of relevance assessments for testing was not yet available when these experiments were conducted, which is why results come from the training set.

	EN		FR		DE	
topics	196	(65.3%)	15	(5.0%)	89	(29.7%)
qrels	2551	(72.2%)	177	(5.0%)	771	(21.8%)
documents	1,839,915	(69.0%)	189,218	(7.1%)	639,124	(24.0%)

Table 2.3: CLEF-IP 2010 collection statistics by original language. Topics are queries, qrels are the patents relevant to those topics, and documents are patents to index and search.

patents from the United States Patent and Trademark Office (USPTO) the *brief description* is the best-performing single field from which to generate queries. European patents from the EPO do not have this field so we use the closest equivalent, the abstract. Given the abstract of a patent, we extract queries in two different ways: (i) using the entire abstract, minus stopwords (***abstract queries*** henceforth), and (ii) using the top k weighted terms from the abstract (***weighted queries*** henceforth). For the latter set of queries, we use *tf-idf* (Spärck Jones, 1972) to measure term weight, and set $k = 20$. As a result, weighted queries are much shorter than abstract queries. The average length of the abstract queries is 46.30 terms for English, 45.08 terms for German, and 40.13 terms for French, i.e., roughly double the size of the weighted queries. This, combined with the fact that abstract query terms are not weighted (i.e., selected according to their salience), means that we expect the abstract queries to contain more noise (i.e., off-topic terms) than the weighted queries. Note that for the weighted queries the weights are only used to filter terms and have no effect on the ranking.

2.3.1.3 Plan of experiments

We use Apache Lucene[10] to index the collection without omitting stopwords or using any stemming. For retrieval, we use Lucene's standard implementation of the tf-idf retrieval model, and we perform a standard TREC[11] eval-

[10] `http://lucene.apache.org/`
[11] `http://trec.nist.gov/`

uation of the top 1000 returned documents, using the standard measures of mean average precision (MAP), precision at 10 (P10), and recall at 1000 (Recall). We also include the Patent Retrieval Evaluation Score (PRES) (Magdy and Jones, 2010), which has been designed specifically for recall-oriented retrieval tasks.

We organize our experiments as follows.

(i) The **baseline** uses a monolingual query. For example, an English query is used to search all patents in English, and also the portions (i.e., claims) of German or French patents that have been translated to English.

(ii) We run two **query translation** experiments using separately the two different dictionaries: dict.cc and PatDict. We refer to these runs as QT_D and QT_P respectively.

(iii) Because our query translation is also a form of query expansion, in the sense that we expand the original queries with their translations, we also conduct a run with standard statistical **query expansion**. We use Rocchio's query expansion (Rocchio, 1971), as is implemented in Lucene[12] (Rubens, 2006), to expand the queries with the top t most pertinent terms from the top d most relevant documents. We tune t and d as follows: t = 10,20,30,...,100 and d = 1,2,3,4,5,8,10,15,20, separately for MAP, P10, recall, and separately for abstract queries and weighted queries. The best performance is uniformly achieved with $d = 1$, but optimal t values vary as follows:

- for MAP, $t = 40$ always;
- for P10, $t = 60$ for abstract queries and $t = 30$ for weighted queries;
- for recall, $t = 40$ always;

[12] `http://lucene-qe.sourceforge.net/`

The Rocchio formula also includes two weighting parameters α and β, which we keep at default values ($\alpha = 1, \beta = 0.75$ (Manning et al., 2008)). We refer to this query expansion run as QE.

(iv) Finally, we combine Rocchio's **query expansion with query translation**. Specifically, we first expand the original (monolingual) query using Rocchio's query expansion, and then translate all terms in the Rocchio-expanded query using a translation dictionary. As before, we tune t and d; their optimal values are identical to those reported in (iii) above. We refer to these runs as QE+QT_D and QE+QT_P respectively.

In total, we conduct six experiments: *baseline,* QT_D*,* QT_P*,* QE*,* $QE + QT_D$*,* $QE + QT_P$.

2.3.2 Results

2.3.2.1 Analysis by original query language

Tables 2.4 and 2.5 display MAP, P10, recall, and PRES for our experiments, grouped by the language of the original query. Table 2.4 has results for abstract queries and Table 2.5 for weighted queries.

Query translation does not consistently overperform or underperform with respect to the baseline across all experiments and metrics, but focusing only on translations with PatDict (QT_P) and measuring recall we do see consistent improvement. Comparing the domain-free (dict.cc) and domain-specific (PatDict) dictionaries used for translation, we observe that PatDict leads to higher recall and PRES, but does not have consistently higher MAP or P10 scores across languages. Since prior-art search heavily relies on recall, the domain-specific dictionary might be a better choice.

Effect of noisy terms. Comparing Table 2.4 to 2.5, we observe that the baseline weighted queries outperform the baseline abstract queries for all metrics (e.g., 0.04704 vs. 0.03841 MAP, 0.06455 vs. 0.05351 P10, 0.33474 vs. 0.29381 recall, and 0.26527 vs. 0.23118 PRES; these are all significant

		DE	EN	FR	All
MAP	baseline	0.03144	0.04101	0.04588	0.03841
	QT_D	0.02902*	0.04082	0.04923	0.03773
	QT_P	0.02803	0.04283	0.04389	0.03848
	QE	0.03829*	**0.04750***	0.05644	0.04520*
	QE+QT_D	0.03829*	0.04749*	0.05644	0.04520*
	QE+QT_P	**0.03831***	**0.04750***	**0.05645**	**0.04522***
P10	baseline	0.04494	0.05590	**0.07333**	0.05351
	QT_D	0.04157	0.05641	**0.07333**	0.05284
	QT_P	0.03708	0.05744	0.06667	0.05184
	QE	**0.05281**	**0.06205**	**0.07333**	**0.05987**
	QE+QT_D	**0.05281**	**0.06205**	**0.07333**	**0.05987**
	QE+QT_P	**0.05281**	**0.06205**	**0.07333**	**0.05987**
Recall	baseline	0.18048	0.34151	0.34625	0.29381
	QT_D	0.17755	0.33925	0.33958	0.29114
	QT_P	**0.24122***	0.35179	**0.36627**	0.31960*
	QE	0.23029*	**0.36157***	0.35145	0.32199*
	QE+QT_D	0.23029*	0.36135*	0.35145	0.32184*
	QE+QT_P	0.23283*	0.36135*	0.35145	**0.32260***
PRES	baseline	0.14792	0.26463	0.29029	0.23118
	QT_D	0.14602	0.26360	0.28495	0.22967
	QT_P	**0.18046**	**0.27277**	0.30243	**0.24678**
	QE	0.17271	0.26872	0.30973	0.24220
	QE+QT_D	0.17296	0.26873	0.30972	0.24228
	QE+QT_P	0.17371	0.26884	**0.30978**	0.24258

Table 2.4: **Results for abstract queries by original query language.** We use queries consisting of the abstract without stopwords. *baseline*: monolingual query. QT_D, QT_P: query translation with dict.cc or PatDict. QE: query expansion. Best scores marked bold. * marks statistical significance with respect to *baseline* at $p < .05$ using the approximate randomization test (Smucker et al., 2007).

		DE	EN	FR	All
MAP	baseline	**0.04081**	0.04968	0.04971	**0.04704**
	QT_D	0.03615*	0.04886	0.04278	0.04477*
	QT_P	0.02313*	**0.05154**	0.04209	0.04261*
	QE	0.03777	0.04899	0.05637	0.04602
	QE+QT_D	0.03777	0.04899	0.05637	0.04602
	QE+QT_P	0.03783	0.04898	**0.05642**	0.04604
P10	baseline	**0.06067**	0.06513	**0.08000**	**0.06455**
	QT_D	0.05506	0.06256	**0.08000**	0.06120*
	QT_P	0.04157	**0.06769**	0.06000	0.05953
	QE	0.05169	0.05487	0.06667	0.05452*
	QE+QT_D	0.05169	0.05487	0.06667	0.05452*
	QE+QT_P	0.05169	0.05487	0.06667	0.05452*
Recall	baseline	0.24090	0.37800	0.32910	0.33474
	QT_D	0.21222*	0.37668	0.32688	0.32523*
	QT_P	**0.25137**	**0.38608**	0.34834	**0.34409**
	QE	0.23284	0.34218*	**0.37085**	0.31107*
	QE+QT_D	0.23284	0.34218*	**0.37085**	0.31107*
	QE+QT_P	0.23294	0.34218*	**0.37085**	0.31110*
PRES	baseline	**0.19358**	0.29555	0.29701	0.26527
	QT_D	0.16634	0.29323	0.26520	0.25406
	QT_P	0.18674	**0.30505**	0.26669	**0.26791**
	QE	0.18852	0.25712	0.30538	0.23912
	QE+QT_D	0.18877	0.25713	0.30530	0.23919
	QE+QT_P	0.19008	0.25726	**0.30638**	0.23973

Table 2.5: **Results for weighted queries by original query language.** We use queries consisting of the top weighted terms from the abstract. *baseline*: monolingual query. QT_D, QT_P: query translation with dict.cc or PatDict. QE: query expansion. Best scores marked bold. * marks statistical significance with respect to *baseline* at $p < .05$ using the approximate randomization test.

differences[13]). The abstract queries seem to contain more noise, which hurts overall retrieval performance. This affects query translation, as potentially noisy terms are translated and become translated noise. Often, such potentially noisy terms consist of commonly occurring terms, which are more likely to be covered in the dictionary, than other salient but more technical terms (this is particularly true for dict.cc). In this case, such terms may have higher translation probabilities simply because of their increased frequency of (co-)occurrence in the translation resources. Overall, we do not see this effect (of introducing more potentially noisy terms) with query expansion because query expansion chooses weighted terms and effectively ignores less significant terms. In general, we see little improvement in QE (with and without translation) from abstract queries to weighted queries, and in fact, scores decrease for some metrics (e.g., P10 All; MAP German and French). The relative decrease in query expansion's effectiveness using weighted queries is apparent in the differences between QE and the baseline. QE MAP increases 17.7% (0.03841 to 0.04520) over the baseline for abstract queries (Table 2.4), while decreasing by 2.2% for the weighted queries (Table 2.5); the same is true for P10 which increases by 11.9% (0.03841 to 0.05987) with abstract queries and decreases by 15.5% for weighted queries.

Language morphology. Looking at retrieval performance by language, perhaps the most consistent result is that German retrieval does worse than English and French. German has the lowest baseline scores of the three languages and precision in particular seems to drop further when adding translation to German queries. We expect both the low retrieval performance and negative impact of translation to be due in part to the more complicated morphology of German. For French queries, like the German ones, translation using PatDict has a negative effect on precision. This contrasts with the PatDict translations from English where the opposite is true. The French and German performance is probably caused by the insufficient leverage that QT_P has available when many potential translations cannot be matched be-

[13] $p < .05$. The significance tests throughout the thesis use the approximate randomization test (Smucker et al., 2007).

cause of morphological and compounding variations. This may be aggravated by the fact that no stemming is done in our current retrieval system and that our dictionary lookup does not account for morphology either. It may be that the prevalence of specific compounds in German (a characteristic of complex texts, especially technical texts like patents) is making the translation task harder, which is a well-known problem (Popovic et al., 2006; Stymne, 2008). Overall, the more complex morphology of both German and French likely accounts for some of the problems with translation, meaning that more sophisticated handling of morphology might improve translation accuracy, and hence retrieval performance.

Language coverage. As MAP and precision drop when adding QT_P translations for French and German queries, recall increases. So despite the difficulties due to morphology, some new relevant documents are returned using translations. The predominance of English in the CLEF-IP collection might also contribute to this effect. Monolingual German (or French) queries may highly rank relevant German (or French) patent documents, missing the relevant English ones. If English terms are added to the queries the rank of the relevant German (or French) documents drops, decreasing precision, however, new relevant English documents are retrieved, improving recall. Consider, for example, German queries, where 49% of the relevant documents are German, 47% English, and only 3% French. The monolingual baseline weighted query is likely to return primarily German documents (the 243 returned documents have an average rank of 200.9). For German weighted queries using the PatDict translation (QT_P), recall increases (now a total of 254 returned documents) but the ranking of many of the relevant German documents slips (average rank of 249.5).

Effect of "patentese". Because queries are taken from patent abstracts, we must consider the differences between separate text fields in patents. In particular, for example, given a monolingual German abstract query with a relevant document in English, the German *abstract* query terms should match the *claims* terms in the German-translated claims (e.g., from a granted En-

glish patent) if the relevant document is to be returned. While the abstract is written for a more general audience, the claims are written in "patentese," the very formulaic legal language used in patents to unambiguously describe an invention, and withstand scrutiny during patent prosecution. If the language use differs greatly between abstracts and claims, then retrieving relevant documents in another language becomes even more difficult.

For a relevant document in English (with German-translated claims), the different language usage between abstract and claims might make retrieval difficult for a German query while English queries can access the English abstract, title, and description, in addition to the claims. The large majority of patent documents and queries are in English (see Table 2.3). Likewise, 72.2% of the relevance assessments are in English, which should make it easier for the English queries. In fact, this is one reason why English recall is higher than French and German for weighted queries (Table 2.5).

We expect our approaches using translation to mitigate this problem, especially when using PatDict translations where only 1.6 words per query are not translated (Table 2.2). Another possible way, particularly for the monolingual case, would be to change how the query is generated, using terms from the entire document.

Translation selection. Note that in these experiments we only use the single most probable translation from the dictionary. This can be problematic as many words are ambiguous, and by limiting the translations to only one, other possible correct translations will be missed. Of course there are other translation methods that allow contextualization, e.g., returning the top n translations, or using phrase-based translations. The former we reserve for future work as a straightforward extension of the approach described in this chapter, while the latter, phrase-based translation, is explored in depth in the next chapter.

2.3.2.2 Analysis by query difficulty

In order to further understand our results, we look at retrieval performance per query, and group queries based on the recall of the baseline. For this

Query difficulty	Baseline recall
very hard (hard++) queries	0%
hard queries	1% – 49%
easy queries	50% – 99%
very easy (easy++) queries	100%

Table 2.6: Definition of query difficulty based on the recall of the monolingual baseline.

analysis, we assume that the lower the recall of the monolingual baseline, the more difficult it will be to improve retrieval performance using either query translation or query expansion. Based on this assumption, we define four groups of *query difficulty* as shown in Table 2.6[14] Evaluating query difficulty in this way has been done before (using measures other than recall), see for example the TREC 2009 Million Query Track (Carterette et al., 2009).

We observe different trends in groups of different query difficulty, which are found in Tables 2.7 and 2.8 and discussed below.

Very hard queries. For the *very hard* queries, query translation improves performance, but only when using the domain-specific dictionary (QT_P). Although this improvement is modest, it does highlight the potential of using query translations. Specifically, we note a possible advantage in query translation over query expansion: If the original query returns no relevant documents, query expansion will not add meaningful terms, except by accident; translation has a better chance of improving performance in this case because it can add relevant French or German translated terms.

It could be argued that using query translations in this context provides no new information, and that they just repeat what was in the original query. However, we expect translations to act as synonyms or like other query expansion methods. Using translations for expansion might gain access to otherwise

[14]The distribution of queries over these categories varies with the baseline results and can therefore be found in the column headers of Tables 2.7 and 2.8.

	Qry. distrib.	**hard++** 23.7%	**hard** 48.5%	**easy** 25.1%	**easy++** 2.7%	**All** 100%
MAP	baseline	0.00000	0.02361	0.09294	0.13616	0.03841
	QT_D	0.00000	0.02237	0.09246	0.13789	0.03773
	QT_P	**0.00059***	0.02611	0.08710	0.14288	0.03848
	QE	0.00026*	0.02696	0.10771*	0.18873	0.04520*
	QE+QT_D	0.00026*	0.02696	0.10770*	0.18873	0.04520*
	QE+QT_P	0.00027*	**0.02698**	**0.10772***	**0.18882**	**0.04522***
P10	baseline	0.00000	0.04207	0.12000	0.11250	0.05351
	QT_D	0.00000	0.04069	0.12000	0.11250	0.05284
	QT_P	0.00000	**0.04345**	0.11067	0.11250	0.05184
	QE	0.00000	**0.04345**	**0.14000**	**0.13750**	**0.05987**
	QE+QT_D	0.00000	**0.04345**	**0.14000**	**0.13750**	**0.05987**
	QE+QT_P	0.00000	**0.04345**	**0.14000**	**0.13750**	**0.05987**
Recall	baseline	0.00000	0.24677	**0.58759**	**1.00000**	0.29381
	QT_D	0.00000	0.24143	0.58724	**1.00000**	0.29114
	QT_P	**0.06422***	0.27363*	0.57958	0.98214	0.31960*
	QE	0.04589*	0.28719*	0.58171	0.96825	0.32199*
	QE+QT_D	0.04589*	0.28689*	0.58171	0.96825	0.32184*
	QE+QT_P	0.04612*	**0.28833***	0.58171	0.96825	**0.32260***
PRES	baseline	0.00000	0.18525	**0.46815**	0.89376	0.23118
	QT_D	0.00000	0.18223	0.46759	0.89747	0.22967
	QT_P	**0.04386**	0.19822	0.46492	0.88287	**0.24678**
	QE	0.02114	0.20498	0.45247	0.90740	0.24220
	QE+QT_D	0.02113	0.20514	0.45249	0.90741	0.24228
	QE+QT_P	0.02128	**0.20557**	0.45269	**0.90748**	0.24258

Table 2.7: **Results for abstract queries by query difficulty.** Query difficulty is estimated from baseline recall rate. *very hard* and *very easy* are given as "hard++" and "easy++". The percentages in column headings show the distribution of queries by difficulty. *baseline*: monolingual query. QT_D, QT_P: query translation with dict.cc or PatDict. QE: query expansion. Best scores marked bold. * marks statistical significance with respect to *baseline* at $p < .05$ using the approximate randomization test.

		hard++	**hard**	**easy**	**easy++**	**All**
	Qry. distrib.	19.4%	48.8%	28.4%	3.3%	100%
MAP	baseline	0.00000	0.02800	**0.09696**	0.17368	**0.04704**
	QT_D	0.00000	0.02596*	0.09222*	0.17585	0.04477*
	QT_P	**0.00103***	0.02681	0.08358*	0.16629	0.04261*
	QE	0.00010*	0.02864	0.08932	0.19800	0.04602
	QE+QT_D	0.00010*	0.02864	0.08933	0.19802	0.04602
	QE+QT_P	0.00010*	**0.02866**	0.08934	**0.19805**	0.04604
P10	baseline	0.00000	0.04795	**0.13059**	0.12000	**0.06455**
	QT_D	0.00000	0.04384*	0.12588	0.12000	0.06120*
	QT_P	**0.00172**	**0.04932**	0.10706*	**0.14000**	0.05953
	QE	0.00000	0.04041	0.10588	**0.14000**	0.05452*
	QE+QT_D	0.00000	0.04041	0.10588	**0.14000**	0.05452*
	QE+QT_P	0.00000	0.04041	0.10588	**0.14000**	0.05452*
Recall	baseline	0.00000	0.25537	**0.62121**	**1.00000**	0.33474
	QT_D	0.00000	0.23685*	0.61957	**1.00000**	0.32523*
	QT_P	**0.07817***	**0.26040**	0.59687	0.95962	**0.34409**
	QE	0.03780*	0.24742	0.55216*	0.77619	0.31107*
	QE+QT_D	0.03780*	0.24742	0.55216*	0.77619	0.31107*
	QE+QT_P	0.03694*	0.24782	0.55216*	0.77619	0.31110*
PRES	baseline	0.00000	0.19070	**0.50552**	**0.85047**	0.26527
	QT_D	0.00000	0.17479	0.49367	0.84813	0.25406
	QT_P	**0.05228**	**0.19247**	0.47890	0.82652	**0.26791**
	QE	0.01506	0.18340	0.43266	0.70707	0.23912
	QE+QT_D	0.01506	0.18356	0.43265	0.70714	0.23919
	QE+QT_P	0.01585	0.18397	0.43327	0.70715	0.23973

Table 2.8: **Results for weighted queries by query difficulty.** Query difficulty is estimated from baseline recall rate. *very hard* and *very easy* are given as "hard++" and "easy++". The percentages in column headings show the distribution of queries by difficulty. *baseline*: monolingual query. QT_D, QT_P: query translation with dict.cc or PatDict. QE: query expansion. Best scores marked bold. * marks statistical significance with respect to *baseline* at $p < .05$ using the approximate randomization test.

irretrievable patents. For example, a French patent query with no French relevance assessments[15] may not retrieve any relevant patent in the monolingual baseline, and get little help expanding the query with more French terms (i.e., QE), but retrieve relevant patents after expanding the query with English. This is the case, for example, for two French abstract queries, where the baseline and QE return no relevant documents but QT_P does.

Generally, query expansion is more likely to perform better when given *good* queries, where by *good* we mean queries containing more topical terms and fewer noisy terms. We can see that this is also true for patents.

Very easy queries. MAP and P10 for *very easy* queries consistently benefit from using query expansion, with and without query translation. Recall behaves in the opposite way with decreases from 1.0 to 0.96825 for abstract queries and 1.0 to 0.77619 for weighted queries. However, we believe that this is partially due to estimation bias: *very easy* queries are defined as those that get 1.0 on the baseline and this number is very likely to decrease when comparing to other runs. The larger drop from 1.0 to 0.77619 which occurs with query expansion for weighted queries could be due to *topic drift* in the expanded queries, which can reduce precision and recall. This is potentially a big advantage for query translation, as it is not affected by a similar problem.

The use of the PRES measure strengthens the argument of estimation bias and topic drift. For abstract queries, the PRES results are better for all QE and QE+QT runs, supporting the claim of estimation bias. PRES decreases along with recall for the weighted queries, suggesting possible topic drift.

Query translation and query expansion. If we focus on just the hard queries (*hard* and *very hard*), we see that either QT_p or QE+QT_p always performs best (with the exception of P10 for abstract queries where none of the methods find a relevant document in the top 10 for any query). Overall, either QT_p or QE+QT_p performs best in most cases, which is a trend we also

[15]Three of the 15 French query patents have no French relevance assessments (i.e., only English and/or German).

saw in the analysis by language (Table 2.4). In general, our collective results from these experiments show that query translation and query expansion can be used as complementary techniques without any detrimental effects to retrieval performance.

Finally, a note on the evaluation metrics applied to this patent collection. The CLEF-IP collection, and the NTCIR test collections before it, use the topic patent's citations to automatically collect relevance assessments, instead of human relevance assessments (see Graf and Azzopardi, 2008, for details). Even though the patent citations do indicate relevant documents, it may be that they do not indicate all documents which humans would assess as relevant. It could be the case that the system is returning highly relevant documents which do not show up in the list of citations. With true human-generated relevance assessments, the evaluation numbers from our experiments would very likely be higher.

2.4 Related Work

The last decade has seen an increase in scientific interest in patent retrieval (Lupu et al., 2011; Lupu, 2012), the challenges of which has long attracted NLP approaches (Osborn et al., 1997; Larkey, 1999). Throughout this thesis we look at NLP and IR techniques for intellectual property collections; in this chapter and the following, we specifically use statistical word alignment from NLP to translate patent queries, i.e., we focus on patent multilinguality. There are two large IR evaluation forums that deal with multilingual patent retrieval. The CLEF Initiative[16] has sponsored an Intellectual Property (IP) track since 2009 with various tasks related to patent retrieval, notably the prior art search task for finding relevant prior art patent documents in a multilingual patent collection. Similarly, NTCIR[17] has had separate workshops for cross-lingual and multilingual IR and patent retrieval since 2002; the four most recent meetings also included a patent translation task (Goto et al., 2011).

[16] http://www.clef-initiative.eu/
[17] http://research.nii.ac.jp/ntcir/

Much of the work in patent IR has aimed to improve query creation and has dealt less with developing patent-specific IR models. This underscores the difficulty of building meaningful queries from a patent document as is practice for prior-art search. In early editions of the CLEF-IP track, standard techniques were used for selecting query terms like document frequency (Graf et al., 2009) or inverse document frequency (idf) (Toucedo and Losada, 2010). Xue and Croft (2009) filtered query terms instead by tf-idf and then compared weighting the terms by term frequency (tf), tf-idf, or equal weighting, and found that tf weighting led to the best results for MAP and Recall@100. Lopez and Romary (2009) on the other hand employed a more involved technique that modeled the approach of a real patent searcher – a technique that led to the best results in CLEF-IP 2009 (Roda et al., 2009). Magdy et al. (2011) later compared these two approaches ("simple" and "sophisticated"), and concluded that they are statistically indistinguishable, although "sophisticated" did perform better. Summarization would be another viable approach for filtering valid query terms. Bouayad-Agha et al. (2009) illustrate how patent claims can be summarized, simplifying the patent language while retaining grammaticality. Such a system would allow us to reduce the patent to a feasible query size without restricting the query to the abstract or description fields.

A common approach for improving queries in IR is to use query expansion (Rocchio, 1971; Salton and Buckley, 1990), which has also been explored in patent retrieval. Bashir and Rauber (2009, 2010) conclude that using pseudo-relevance feedback (PRF) in particular benefits retrievability in patent IR. They originally show that using a novel cluster-based PRF approach performs best for patent retrieval (Bashir and Rauber, 2009). In later work (Bashir and Rauber, 2010), they incorporate PRF into a language modeling (LM) approach that also improves retrievability results on a USPTO collection. Comparing their results to others who have used query expansion in patent IR is difficult, as they measure performance by *retrievability* (Azzopardi and Vinay, 2008) on a different patent corpus (in contrast to using TREC measures on a standard collection). Magdy and Jones (2011a) on the other hand conclude that PRF does not help patent retrieval *effectiveness* (cf. patent re-

trievability), evaluating MAP and PRES on a standard CLEF-IP collection. They try other expansion techniques as well, including one using translations that is similar to what we describe in this chapter, but they find that while precision might improve, recall does not. Mahdabi and Crestani (2012) also show that the typical application of PRF does not help in patent retrieval, but that it can help for some queries. They use a regression model to identify those queries where PRF may help retrieval, which returns significantly better results (MAP and PRES). Later they extend this approach by expanding queries with select noun phrases (Mahdabi et al., 2012). Ganguly et al. (2011) also employ PRF, but use it to reduce queries instead of expanding them. Relevant documents are returned for full patent queries and then those queries are stripped of the non-relevant information that is not found in the returned relevant documents. This approach proves to be effective with improvement in MAP and PRES scores. Despite the extensive use of query expansion described above, with much of it being conducted on the multilingual CLEF-IP collection, query expansion for patent IR has only been applied to monolingual retrieval tasks.

Ballesteros and Croft (1997) did test query expansion in a cross-lingual setting, however, the well-known dataset sensitivity of query expansion often led to instability. Lavrenko et al. (2002) used relevance models in a LM framework for cross-lingual IR. They argue that this approach avoids the tuning usually required for traditional query expansion while still obtaining competitive results.

Our approach to query expansion is to make use of query translation. Translating queries has been studied for some time (Dunning and Davis, 1992) and it is typically realized using translation dictionaries, machine translation (MT) systems, parallel corpora or combinations of these (see Kraaij et al. (2003); Oard et al. (2008) for an overview). Mainstream approaches to multilingual IR aim to maximize translation accuracy in order to improve retrieval performance (Kraaij et al., 2003), however more recent approaches have also focused on improving retrieval performance using "approximate" rather than accurate translations (Gao et al., 2010; Wang and Oard, 2006). Specifically, Gao et al. (2010) present a system for cross-lingual query sug-

gestion reliant on web query logs. Queries are not literally translated, but instead a multilingual web query log is used to find target queries similar to the original source queries. Their system relies on word translations derived from the Europarl corpus, as well as co-occurrence statistics, and click-through information from a web query log to estimate the similarity between queries cross-lingually. This method outperforms MT-based and dictionary-based query translation. However, it would be difficult to use a similar method with patent retrieval because of the lack of query log data for patent retrieval. Although, in principle, query logs and click-through data are available from the web, in practice, collecting this information from patent searchers would prove difficult. Even releasing what and how one is searching can possibly be a liability for patent professionals.

A second approach to depart from exact query translation, from Wang and Oard (2006), considers translation as a problem of *meaning matching*. Bidirectional term alignments are extracted from Europarl (for English-French) and English-Chinese parallel news corpora, the terms of which are then augmented with WordNet synset information. This method performs well, but it would be infeasible to apply it directly to the patent domain as there are no resources like WordNet for patents. This approach is still similar to our own though, in that translations are used like synonyms in building queries. Wang and Oard also only use term translations and speculate that they might benefit from also using phrases.

As mentioned above, Magdy and Jones (2011a) use MT to expand queries for patent retrieval in English. They further explore the effect of MT by building MT systems using varying amounts of parallel data (Magdy and Jones, 2011b). They show that retrieval results are not significantly hurt by using the MT system trained on less data. It would be interesting to try this in future work.

Finally, several studies use word alignment algorithms from statistical MT to extract dictionaries from corpora (Lefever et al., 2009; Sun et al., 2000), or analyze multilingual collections with the goal of improving retrieval (Diaz and Metzler, 2007; Franz et al., 2001; Kraaij et al., 2003; Nie et al., 1999; Yang et al., 1998). However, most of these approaches use statistical

word alignment to extract a multilingual dictionary not directly from the retrieval collection, but on some external collection. Our approach differs because we extract the translation dictionary directly from the patent retrieval collection, something that had not been done before to the best of our knowledge. We believe that this is a promising approach because dictionaries are highly domain-dependent and the better the correspondence between the dictionary's domain and the collection's domain, the more improvement in retrieval performance we would expect.

2.5 Summary

In this chapter we start to explore the multilingual aspect of patent retrieval. In particular we focus on the impact of using different dictionaries for term translation. Starting with a collection of partially translated patents, we study the effect of query translation on retrieval performance. Specifically, we expand monolingual patent queries with their translations, using both a domain-specific patent dictionary that we extract from the patent collection, and a general domain-free dictionary. The experimental evaluation on a standard CLEF-IP dataset has mixed results, but some conclusions can still be drawn. The patent-specific translation dictionary generally outperforms the domain-free dictionary, although neither show great improvement compared to standard statistical monolingual query expansion. Query expansion with translation does not significantly improve over standard query expansion but the two may still be complementary and there appears to be no detriment to their combination. In general, our results show greater improvement when the source language is English, as opposed to French, and even more so German, a finding partly due to the effect of the complex German and French morphology upon translation accuracy, but also partly due to the prevalence of English in the collection (69% of the original language). Finally, a thorough analysis by query difficulty revealed that cases where standard query expansion fails (e.g., zero recall) can still benefit from query translation.

Chapter 3

Phrase Translation in Patent Retrieval

In this chapter, we extend our investigation of the potential of machine translation to help patent information retrieval. Previously (Chapter 2), we compared a domain-free translation dictionary to one compiled from the patents themselves. Here we build on the latter approach and train a complete statistical machine translation system instead of building a dictionary using only word alignments. In this chapter we focus only on the translation from German and French because, as we briefly touch on in the previous chapter, with such a high number of relevant documents for English, translation from English has a more limited opportunity to improve results (and a greater chance of hurting them).

The chapter is organized as follows. We begin in Section 3.1 by motivating the use of phrase translation for multilingual patent queries. In Section 3.2 we revisit the methodology for translating patent queries and detail the extension to phrase translation. Section 3.3 includes the experimental evaluation of our approach with a detailed discussion of the results. In Section 3.4 we review some of the relevant work related to phrase translation and patent IR. Finally, in Section 3.5, we summarize our work on query translation and expansion.

3.1 Motivation

As detailed in the previous chapter, patent IR is a very challenging retrieval task, made difficult in part by the complex technical and legal language used in patents, *patentese* (Atkinson, 2008). Patent IR is a recall-oriented task that requires an exhaustive search, as overlooking a single relevant patent could be very costly if that patent has been infringed upon. To meet the demands for high retrieval effectiveness, in particular for a multilingual patent collection like that of the European Patent Office (EPO), we propose using multilingual queries to access the multilingual collection. In particular in this chapter we propose a query translation and expansion approach that uses phrase-based statistical machine translation (SMT).

As in Chapter 2, given monolingual patent queries we plan to build multilingual queries using query translations. However, now we are interested in the effects of using phrase translations along with term translations. We present two alternatives for this approach: (i) term-by-term translations and (ii) translations that can also involve phrases. For brevity, we simply refer to these two approaches as *term translation* and *phrase translation* – even though phrase translations are in reality a superset of term translations, i.e., for phrase translation there are four different types of translations that can occur in the query:

1. term to term
2. term to phrase
3. phrase to term
4. phrase to phrase

Our query translation is realized using a domain-specific machine translation system trained on patents, i.e., we compile a translation dictionary of terms and phrases specifically for the patent domain. We train the SMT system directly on the parallel translations found in patents from the patent collection used for retrieval. More precisely, we identify the parallel translations, align them, and compute the translation probabilities between terms

and phrases in the aligned translations. These translations constitute the entries in our domain-specific patent translation dictionary.

Our approach differs from other work (e.g., Franz et al., 2001; Kraaij et al., 2003; Wang and Oard, 2006) in that we derive a bilingual term and phrase dictionary from the retrieval collection itself – that is, we do not derive the dictionary from unrelated parallel corpora. This aspect of our work is important because it is difficult to obtain good translation coverage when using a generic dictionary or parallel text from a different corpus (cf. Chapter 2 or Franz et al. (2001)).

To evaluate our query translation hypothesis, we compare retrieval performance of our multilingual queries to that of monolingual queries, using a competitive retrieval model. We further include runs where translation has been realized with a competitive MT system (Kettunen, 2009), *Google Translate*,[1] so that our translation approach can be compared to a state-of-the-art and freely-available competitive approach. Because our query translation is also a form of query expansion (since queries are expanded with their translations), we conduct additional experiments using pseudo-relevance feedback.

3.2 Methodology

The aim of our approach is to turn monolingual queries into multilingual queries for patent IR. To this end, we extract a translation dictionary of terms and phrases from a parallel patent corpus (Section 3.2.1) . The parallel corpus is comprised of the same patents as in the retrieval collection (see Section 3.3.1 for its description). We then describe how we use the extracted dictionary to translate and expand the original monolingual queries (Section 3.2.2), and specify in which cases we do the translation and expansion (Section 3.2.3).

Our experimental evaluation is done on a standard CLEF-IP (Roda et al., 2009) dataset, focusing on the morphologically difficult cases of German and French queries. We wish to observe whether term or phrase translation is

[1] `http://translate.google.com/`

more beneficial for French of German, and test if our translation approaches are compatible with relevance feedback.

3.2.1 Extracting a Translation Dictionary of Terms and Phrases

Our retrieval collection contains patents from the European Patent Office (EPO), which may include text fields, e.g., *title*, *abstract*, *description*, and *claims*; metadata fields, e.g., *applicant*, *inventor*, *International Patent Classification (IPC)*, and *date published*; and figures and illustrations. The claims field is of particular interest for patent IR, because it contains the legally-binding portion of the patent that may be used later to determine the patent's validity or defend it against infringement (Atkinson, 2008; Azzopardi et al., 2010). In EPO patents, the claims of granted patents have been manually translated into English, French, and German. Therefore, the claims of our patent collection may be seen as a parallel corpus which can be used to extract translation dictionaries specific to the patent domain.

Our method for producing phrase translations is an extension of our previous dictionary extraction approach in Chapter 2. We briefly summarize it again here and then describe more thoroughly the extension to obtain phrase translations.

As before (Section 2.2.1), we extract a parallel corpus from the EPO patent claims. The (sub-)sentences in the claims are aligned using the Gargantua sentence aligner (Braune and Fraser, 2010), and then words are subsequently aligned using the GIZA++ word alignment toolkit (Och and Ney, 2003). In Chapter 2 this word alignment comprised our translation dictionary, however, here we use this word alignment to train an SMT system. We use the phrase-based approach of the Moses[2] SMT system (Koehn et al., 2007). To obtain phrase translations, we use the default GDFA (*grow diagonally final AND*) alignment (Koehn et al., 2003) from GIZA++ to train Moses and produce a bidirectional phrase table. The phrase table includes phrase translations with a source phrase of length m and a target phrase of length

[2] `http://www.statmt.org/moses/`

Languages	# entries	*term* → *term*	*term* → *phr.*	*phr.* → *term*	*phr.* → *phr.*
FR-EN	162,840,175	922,952	1,715,491	3,437,236	156,764,496
FR-DE	157,977,915	1,645,245	3,570,673	10,419,547	142,342,450
DE-EN	116,611,676	1,290,111	4,642,276	2,863,824	107,815,465

Table 3.1: Patent dictionary: Note that the dictionaries are bidirectional, i.e., 1,715,491 French terms can be translated to an English phrase and 1,715,491 English phrases can be translated to a French term.

n, where m and n are between 1-7 inclusive. This means that the phrase table contains term to term translations, term to phrase translations, phrase to term translations, and phrase to phrase translations. The phrase table is also bidirectional, so the size of the German-English dictionary will be the same as the size of the English-German dictionary. The translation probabilities are not symmetric though; e.g., the German word *Gefäß* translates to *vessel* with probability 0.61, but *vessel* translates to *Gefäß* with probability 0.26. Note that the phrases in the phrase table are *statistical* phrases and not *linguistic* phrases (cf. Ballesteros and Croft, 1997). In other words, the phrases we use are just sequences of words of arbitrary length (up to length 7) that do not necessarily constitute a linguistically meaningful unit, while linguistic phrases have a particular meaning that could potentially be found in a dictionary.

Table 3.1 shows the total number of entries for each language pair as well as a summary by the type of translation relation, i.e., term → phrase. Generally, there are many more equivalences involving phrases simply because there are many more phrases than terms in any given language. The highest number of equivalences between a term and a phrase (in either direction) occur for German-English (4,642,276) and French-German (10,419,547) because compounds (which are terms) are frequent in German and they are most often translated as phrases.

original German query	ein tintenstrahl aufzeichnungsmaterial mit einem träger und mindestens einer unteren pigment ...
French term translation	*un jet support avec un support et moins une inférieure pigment ...*
English term translation	*a inkjet recording with a carrier and least a lower pigment ...*

Table 3.2: Example of term translation from German to French and English.

3.2.2 Translating Queries

Given the term and phrase translation dictionary described above, we use two different methods to translate queries; the first is term to term translation (QT_t); the second includes phrase translations (QT_p). In the term translation method, for every term t in a query Q, we identify the best translation t' of the term, and create an extended query Q' by appending t'.[3] For now we consider only the single best translation, i.e., the most probable one according to the bilingual dictionary extracted by Moses. We do this for all possible source-language combinations, resulting in a multilingual query. An example of term translation can be seen in Table 3.2, where the first row is the original German query, and the second and third rows are the French and English term translations. This example shows some of the typical problems that occur in automatic term-by-term translations: some terms are poorly translated (German *aufzeichnungsmaterial* 'recording material' is translated as 'support' by the German-French dictionary) and some terms can only be adequately translated as a phrase (German *aufzeichnungsmaterial* may best be translated as the phrase, 'recording material', in English).

For the phrase translation method, we extract only those phrases in the original query for which we have a translation. Our definition of phrase is the longest *n-gram* (i.e., string of n words) in the dictionary for n=[1-7]. Terms

[3]This is also shown in Algorithm 1 in Chapter 2.

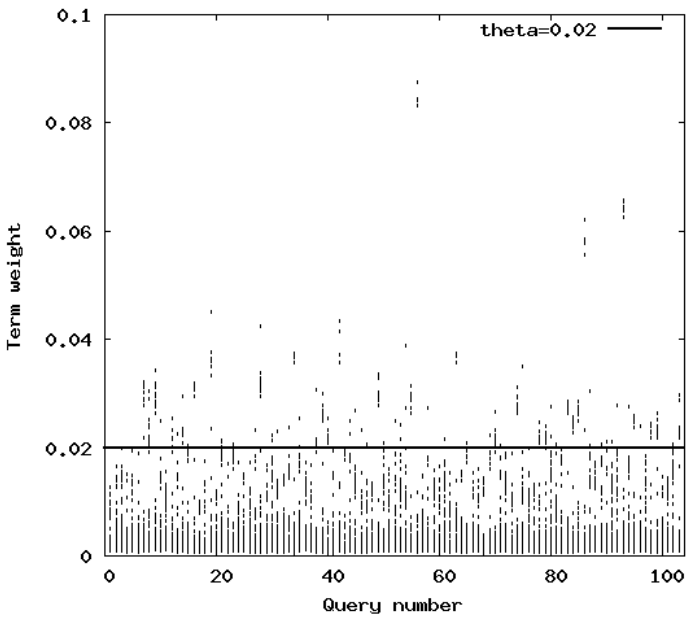

Figure 3.1: Term weights of the monolingual baseline query set. Terms above the threshold, θ, are translated.

that are not present in any phrase are translated as terms in the multilingual query. Since stopwords will be removed by the retrieval system we do not need to translate phrases of only stopwords. So although "of a" would be considered a phrase in our approach, we do not translate it. Taking this one step further, we remove any stopwords at the beginning or end of a phrase, but preserve the stopwords within phrases. So "of the ink jet" simply becomes "ink jet" while "coated with aluminum" remains the same.

3.2.3 Translating Salient Terms

In the previous section, we described our query translation method. We do not apply this to all query terms, but to a selection of the most salient terms in the query. This is motivated by our previous work, which showed the limitation of fully translating whole patent queries (see, for example, the discussion of noisy terms in Section 2.3.2.1). Additionally, in other recent

work in CLIR (e.g., Wang and Oard, 2006; Gao et al., 2010), we have seen that satisfactory retrieval performance can be achieved with approximate translations of queries.

We select the terms that are to be translated as follows. Using our baseline retrieval model (see Section 3.3.1), we look at the term weights assigned to the individual query terms. Figure 3.1 displays this distribution over our whole query-set (also described in Section 3.3.1). By defining a threshold θ for term weights, we can assume that we reasonably separate the most salient query terms from the rest. Hence, we translate only the terms whose weight is greater than θ. The higher the threshold, the fewer terms are translated. For example, for $\theta = 0.02$, 9.1% of query terms are selected for translation.

3.3 Experiments

3.3.1 Settings

For our experiments we investigate the translation and expansion of patent queries from the original monolingual queries to multilingual queries. We focus particularly on the translation from German (to English and French) and French (to English and German) because they have more complex morphology than English, and hence pose more of a problem to multilingual IR than translation from English. It is also the case that only 11.5% of relevance assessments for English queries are German or French patents – so there is less to gain by translating from English.

We use the same collection as in Chapter 2, the CLEF-IP 2010 patent collection (Roda et al., 2009) from the IRF[4] (see Table 3.3 for a review of the size and Table 2.3 on page 39 for details on the distribution by language). The query set on which we conduct our experiments has 104 topics in German and French with relevance assessments. These topics are not TREC-style queries, but full patent documents. We generate queries in a manner similar to the previous chapter by taking the set of unique terms from the abstract

[4] http://www.ir-facility.org/

Size	84 GB
# documents	2,680,604
# terms	9,840,411,560
# unique terms	20,132,873

Table 3.3: Size of CLEF-IP 2010 collection.

as the baseline query. This results in an average query length of 57.6 terms.[5] We choose the abstract due to the success of Xue and Croft (2009) using the *brief summary* field in USPTO patents.

In Chapter 2 we used Apache's Lucene, a robust IR library that proved to be valuable in our collaboration with the University of Stuttgart's Institute for Visualization and Interactive Systems (VIS) (some of that collaboration is covered in Appendix A). However, the primary retrieval model in Lucene is a Boolean model that is then scored with cosine similarity in a vector space model. Instead, for these experiments we index the entire collection using Indri,[6] a state-of-the-art IR system that is widely used for research. By using Indri, we can more easily compare our results to other recent research in patent retrieval.

Our settings for Indri are the following. We indexed the collection without removing stopwords or stemming. For retrieval we use the default Indri retrieval model, a combination of language modeling and inference networks (Metzler and Croft, 2004), using Dirichlet smoothing. We tune Dirichlet's μ parameter in the range $\mu = \{5000, 7500, 10000, 12500, 15000, 17500, 20000\}$. For retrieval, we use standard stoplists for German and French.[7] Our translation approach includes a term weight threshold θ (described in Section 3.2.3), which we tune: $\theta = \{0.016, 0.02, 0.025\}$. Because our query translation approach expands queries with their (partial) translations, we also conduct experiments with pseudo-relevance feedback (PRF), using Indri's default PRF

[5]Note that stopwords will be removed at query time.

[6]`http://www.lemurproject.org/`

[7]accessible from `http://members.unine.ch/jacques.savoy/clef/`

implementation, an adaptation of Lavrenko's relevance models (Lavrenko and Croft, 2001). Note that in Chapter 2 we used Rocchio PRF instead. PRF uses the following parameters: number of documents ($fbDocs$) and number of terms ($fbTerms$). We set $fbDocs = 1$ and $fbTerms = 40$ based on our best results in Chapter 2. Finally, we include a run that uses a state-of-the-art translation system for query translation. We choose Google Translate, which has been shown to lead to the best retrieval results (Kettunen, 2009) among several MT systems. This type of translation differs from our own: we submit the whole query for translation to Google Translate, whereas our approach translates only salient terms/phrases; also, Google Translate is domain-free, while our approach uses a patent translation dictionary extracted from the retrieval collection. We include the Google Translate runs simply to contextualize the results from our approach (which indicate this is a difficult task). We use the standard TREC evaluation measures mean average precision (MAP), precision at 10 (P10), and recall, along with the recall-oriented PRES (Patent Retrieval Evaluation Score) from Magdy and Jones (2010). We tune separately for each evaluation measure.

3.3.1.1 Outline of experiments

Our experiments are set up as follows:

(i) **baseline:** original monolingual query;

(ii) **QT$_t$:** the original query is expanded with term translations of its most salient terms (as described in Section 3.2.1);

(iii) **QT$_p$:** the original query is expanded with phrase translations of its most salient terms. A phrase is translated if it contains at least one salient term. Any salient terms that are not contained in at least one phrase are translated as terms;

(iv) **PRF:** same query as baseline but with PRF;

(v) **QT$_t$+PRF:** same as QT$_t$ but with PRF;

(vi) **QT_p+PRF:** same as QT_p but with PRF;

(vii) **Google MT:** the full abstract query is expanded with its translation using Google Translate.

3.3.2 Results

3.3.2.1 Analysis by language

Table 3.4 summarizes our experimental results for German and French queries. We can see improvement by expanding queries with translation although the improvements are not significant and not consistent across languages or retrieval measures. For German, term translation leads to better precision but both recall measures are higher using phrase translation. We originally expected that term-to-phrase translation would handle German compounds better than term-to-term translation. Indeed, in query 237 (Q#237), *korngrößenverteilung* translates to 'particle size distribution' with phrase translation, and only to 'size' using term translation. For this query, term translation improves MAP by 4% over the baseline, and phrase translation improves MAP by 36% over the baseline. However, there are also cases where German is equally well translated with phrases or terms, e.g., in Q#201, *tintenstrahl* is translated as 'inkjet' or 'ink jet', respectively. The PRF run for German improves over the baseline, but using PRF with translated queries leads to mixed results. P10 and PRES are best using only PRF, while MAP and recall benefit some from both phrase and term translations.

French results contradict the German ones with term translation producing better recall results (recall and PRES) and phrase translation leading to more substantial gains in precision (MAP and P10). For the French PRF runs, term translation with PRF does a little worse than PRF alone (except for recall), while phrase translation shows clear improvement compared to term translation and PRF (except P10), particularly in terms of recall and PRES. One reason that French may benefit more from phrase translation than German is that it does not have as many compounds: concepts that are expressed as phrases in French are often translated as compounds in German.

		DE	FR
MAP	baseline	0.0581	0.0527
	QT_t	**0.0598**	0.0556
	QT_p	0.0577	**0.0614**
	PRF	0.0664	0.0730
	QT_t+PRF	0.0667	0.0719
	QT_p+PRF	**0.0672**	**0.0744**
	Google MT	0.0473	0.0652
P10	baseline	0.0864	0.0667
	QT_t	**0.0875**	0.0867
	QT_p	**0.0875**	**0.1000**
	PRF	**0.0875**	**0.1067**
	QT_t+PRF	0.0841	0.1000
	QT_p+PRF	0.0864	0.1000
	Google MT	0.0659	0.1200
Recall	baseline	0.2518	0.3288
	QT_t	0.2673	**0.3431**
	QT_p	**0.2679**	0.3358
	PRF	0.2694	0.3346
	QT_t+PRF	**0.2810**	0.3378
	QT_p+PRF	0.2785	**0.3771**
	Google MT	0.3276*	0.4021
PRES	baseline	0.2104	0.2844
	QT_t	0.2163	**0.2926**
	QT_p	**0.2209**	0.2914
	PRF	**0.2274**	0.3042
	QT_t+PRF	0.2248	0.3002
	QT_p+PRF	0.2237	**0.3272**
	Google MT	0.2611	0.3297

Table 3.4: German and French results. Best scores in bold. * marks statistical significance with respect to baseline at $p < .05$ using the approximate randomization test (Smucker et al., 2007). QT_t is term translation and QT_p is phrase translation.

The phrase *flux de matière* 'flux of material' in Q#242 gets translated into 'materialströme' in phrase translation, and into 'strom', 'von', and 'material' in term translation.

Adding PRF improves results for all three queries (comparing baseline to PRF, QT_t to QT_t+PRF, etc.) with the exceptions of German P10 with term and phrase translation and French recall with term translation. This is an interesting result, as the usefulness of PRF in patent IR has been debated in the literature (cf. Bashir and Rauber (2009) and Magdy and Jones (2011a)). Previous work focused only on English patents however, and not multilingual patent IR. Using our combined expansion approach with translation and PRF, MAP and recall results are consistently better. This shows that the combination of expansion techniques can help retrieval performance, and that these two different approaches are not incompatible.

Finally, looking at the Google MT run, it is not surprising that it does best on recall-oriented measures, but performs worse in the other measures: the queries translated by Google contain the full patent abstract and its full translation, meaning that they are very lengthy queries with evidently better coverage at the expense of precision.

3.3.2.2 Analysis by query difficulty

In order to further understand our findings we look more closely at performance on a per-query basis. Specifically, we group queries on the basis of their baseline recall, on the assumption that queries of very low baseline recall will be much more difficult to improve (using either PRF or translation), than queries with higher baseline recall. Tables 3.5 and 3.6 present retrieval performance split between three groups of query difficulty: *very hard* (baseline recall = 0%), *hard* (baseline recall = 1%–49%), and *medium* (baseline recall = 50%–100%).[8]

German. We see that for German queries of *medium* difficulty the term and phrase translations perform worse than their baseline and PRF coun-

[8]This is the same as the analysis in Chapter 2 except that we have conflated the two higher recall groups into *medium* (see Table 2.6).

	Qry. distrib.	**hard++** 31.8%	**hard** 48.9%	**medium** 19.3%	**All** 100%
MAP	baseline	0.0000	0.0375	**0.2058**	0.0581
	QT_t	0.0002	**0.0417**	0.2036	**0.0598**
	QT_p	**0.0003**	0.0406	0.1953	0.0577
	PRF	0.0013	0.0525*	**0.2091**	0.0664
	QT_t+PRF	**0.0018***	0.0543*	0.2049	0.0667
	QT_p+PRF	0.0014*	**0.0558**	0.2045	**0.0672**
P10	baseline	0.0000	0.0721	**0.2647**	0.0864
	QT_t	0.0000	0.0767	0.2588	**0.0875**
	QT_p	0.0000	**0.0837**	0.2412	**0.0875**
	PRF	0.0000	0.0860	**0.2353**	**0.0875**
	QT_t+PRF	0.0000	0.0837	0.2235	0.0841
	QT_p+PRF	0.0000	**0.0907**	0.2176	0.0864
Recall	baseline	0.0000	0.2614	**0.6421**	0.2518
	QT_t	**0.0413**	0.2798	0.6078	0.2673
	QT_p	0.0391	**0.2802**	0.6137	**0.2679**
	PRF	0.0461	0.2685	**0.6397**	0.2694
	QT_t+PRF	**0.0957***	**0.2821**	0.5834	**0.2810**
	QT_p+PRF	0.0902*	0.2760	0.5947	0.2785
PRES	baseline	0.0000	0.2029	**0.5761**	0.2104
	QT_t	0.0179	0.2112	0.5560	0.2163
	QT_p	**0.0226**	**0.2189**	0.5526	**0.2209**
	PRF	0.0341	0.2166	**0.5730**	**0.2274**
	QT_t+PRF	0.0541	**0.2272**	0.5000	0.2248
	QT_p+PRF	**0.0545**	0.2243	0.5010	0.2237

Table 3.5: German results by difficulty. The percentages in column headings show the distribution of queries by difficulty. * marks statistical significance $p < .05$ using the approximate randomization test.

terparts, while results for these runs improve for the *hard*, and *very hard* queries. Looking closer, it seems few queries account for most of this variation. For example, in the group of *medium* queries, one query (Q#213) has better recall with term and phrase translation than the baseline, while for two other queries (Q#75 and Q#152) the baseline has better recall. The recall for Q#152 drops substantially with 8 of 10 relevant documents being retrieved in the baseline and none being retrieved with either word or phrase translation. It is unsurprising that all of the relevance assessments for this query are German (hence the settings for this query can be seen as biased). This query alone accounts for most of the decrease in translation results in the *medium* group. A single query can also account for much of the improvement in the *hard* group's translation results. For Q#201 for example, 2 of the 20 relevant patent documents are in German and the baseline query only returns those 2 relevant documents. Adding phrase translations (in particular the addition of the phrase "ink jet") increases the number of relevant documents returned to 15.

We also observe that *medium* difficulty queries tend to have more relevance judgments in their original source language (like Q#152 above), and that *hard* queries tend to have relevance judgments from different languages. To the extent that this is the case, it is understandable that results for *medium* queries worsen with translation: a largely monolingual (say, French) result set has high ranks for relevant documents (which are all French), but for a multilingual result set the ranking of some French relevant documents slips. On the other hand, *hard* queries may improve if the baseline monolingual result set (French) does not match many relevant documents (several English documents with a few French), but with the addition of multilingual documents to the result set, more relevant documents are retrieved. This bias with regard to the percentage of a query's relevant documents that are in the original source language of the query also affects the performance of PRF. PRF chooses terms for expansion from the top ranked documents. These documents are likely to be in the same language as the original query. So an original French query in the baseline will expand the query with French terms. In the cases of QT_t+PRF and QT_p+PRF, the highest ranked result

with QT is often a multilingual patent document and so the multilingual query will have multilingual expansions. With this multilingual patent collection, multilingual query expansion should be more desirable, and in fact QT+PRF outperforms PRF for MAP and recall. In particular, QT+PRF does better than PRF for *hard* and *very hard* queries where it appears there is a larger percentage of relevant documents in a language different than the query.

French. In contrast to the German results, the French MAP and P10 scores improve for term and phrase translation across the *hard* and *medium* groups, and results for *very hard* remain the same. Q#239 is one example where MAP and P10 improve, while recall remains the same. The original French query has recall of 1.0, returning all four of its relevant documents (two of which contain translated claims). Phrase translation still proves to be useful here in improving the relevant documents' ranking (MAP and P10 both improve). For recall, we see the same behavior as with German: recall drops using translations (QT_t and QT_p) in the *medium* group and rises for harder queries. For the *medium* group, only one less document is retrieved using QT_t with respect to the baseline. This single document accounts for the 2.8% relative drop in recall. Note that there are only 15 French queries and an average of 13.5 relevance assessments per query, so one relevant document being added to or dropped from the result set can have a big impact. For the majority of queries however, recall remains the same (discussed below and shown in Table 3.7).

3.3.2.3 Individual Query Evaluation

For a deeper look, Table 3.7 shows how QT_t and QT_p performed against the baseline for individual queries. We counted, for each evaluation measure, the number of queries that performed better than, worse than, or equal to the baseline. All measures for both translation methods, with the exception of one tie (2-2) for QT_t recall, have more queries that exceed the baseline than queries that drop below it. We observe that, for each language/evaluation measure combination, there are more queries improved by phrase transla-

	Qry. distrib.	**hard++** 20.0%	**hard** 46.7%	**medium** 33.3%	**All** 100%
MAP	baseline	0.0000	0.0676	0.0635	0.0527
	QT_t	0.0000	0.0694	0.0695	0.0556
	QT_p	0.0000	**0.0794**	**0.0729**	**0.0614**
	PRF	0.0000	**0.0907**	0.0919	0.0730
	QT_t+PRF	0.0000	0.0894	0.0905	0.0719
	QT_p+PRF	0.0000	0.0852	**0.1038**	**0.0744**
P10	baseline	0.0000	0.0857	0.0800	0.0667
	QT_t	0.0000	**0.1286**	0.0800	0.0867
	QT_p	0.0000	**0.1286**	**0.1200**	**0.1000**
	PRF	0.0000	**0.1143**	**0.1600**	**0.1067**
	QT_t+PRF	0.0000	**0.1143**	0.1400	0.1000
	QT_p+PRF	0.0000	0.1000	**0.1600**	0.1000
Recall	baseline	0.0000	0.2427	**0.6465**	0.3288
	QT_t	0.0000	**0.2864**	0.6283	**0.3431**
	QT_p	**0.0145**	0.2795	0.6073	0.3358
	PRF	0.0145	0.2530	0.6409	0.3346
	QT_t+PRF	0.0145	0.2530	**0.6505**	0.3378
	QT_p+PRF	**0.0290**	**0.3439**	0.6323	**0.3771**
PRES	baseline	0.0000	0.2054	**0.5657**	0.2844
	QT_t	0.0000	**0.2285**	0.5578	**0.2926**
	QT_p	**0.0096**	0.2267	0.5511	0.2914
	PRF	0.0010	0.2367	**0.5806**	0.3042
	QT_t+PRF	0.0113	0.2305	0.5712	0.3002
	QT_p+PRF	**0.0204**	**0.2805**	0.5767	**0.3272**

Table 3.6: French results by difficulty. The percentages in column headings show the distribution of queries by difficulty. * marks statistical significance $p < .05$ using the approximate randomization test.

Eval. meas.	$QT_t >$ *baseline*	$QT_t <$ *baseline*	$QT_t =$ *baseline*	$QT_p >$ *baseline*	$QT_p <$ *baseline*	$QT_p =$ *baseline*
			German			
MAP	19	13	56	33	23	32
P10	5	4	79	7	4	77
recall	16	9	63	18	12	58
			French			
MAP	5	3	7	9	3	3
P10	2	0	13	2	0	13
recall	2	2	11	4	3	8

Table 3.7: German and French performance per query. $QT_t >$ *baseline* indicates that the evaluation measure was greater for term translation than the baseline. $<$ and $=$ indicate less than and equal to, respectively. Other notation as in Table 3.4.

tion than queries improved by term translation (i.e., $33 > 19$ for German MAP), with one tie, 2-2, for French P10. However, in three cases (German MAP, German recall, French recall), there are also more queries where phrase translation does worse than term translation (i.e., $23 > 13$ for German MAP); the other three cases (German P10, French MAP, French P10) are tied. Our interpretation of these results is that phrases have significant potential for improving retrieval results, but they must be carefully selected, otherwise performance will deteriorate. In contrast, term translations are more conservative and less likely to have a negative effect, but at the same time they offer limited improvements. In future work, we could learn which queries benefit more from using phrase translation (see, for example, the regression approach of Mahdabi et al. (2012) for when to apply query expansion).

3.3.2.4 Tuning of μ

Finally, Figures 3.2-3.3 show MAP and P10 scores across the tuning range of μ. The more stable the line of our approach, the less sensitive it is to factors

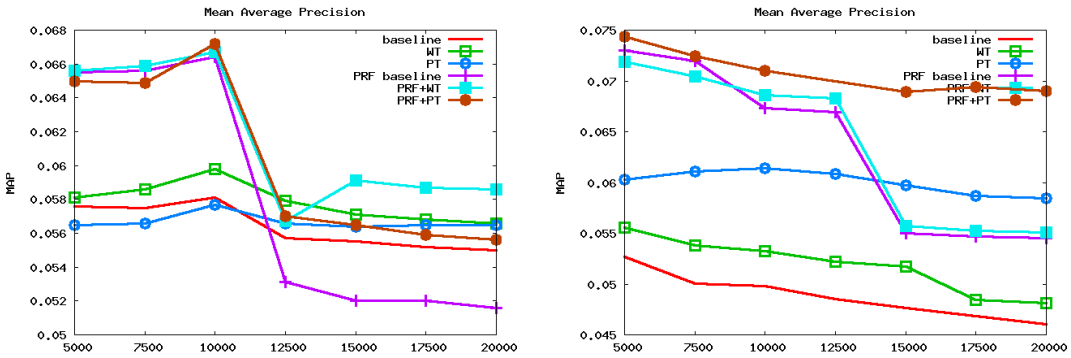

Figure 3.2: Dirichlet prior μ tuning for German (left) and French (right) versus MAP (y axis).

pertaining to variation in document length and collection statistics. For the MAP tuning for both German and French, the results for term and phrase translation are quite stable, while the three runs that use PRF drop (between 10000 and 12500 for German and between 12500 and 15000 for French).

Note that results are less stable for the P10 tuning, although word and phrase translations appear more stable than PRF runs. The German PRF results drop for P10 like they did for MAP.

3.4 Related Work

Most of the discussion of related work in Chapter 2 also applies to this chapter. We will briefly touch again on some of the most relevant of that literature (but for further details see Section 2.4) and introduce some additional work relevant specifically to phrase translation and phrase retrieval.

Generally, in cross-lingual and multilingual IR, translation can be broadly realized using a combination of bilingual dictionaries and/or parallel corpora and/or machine translation (see Kraaij et al., 2003; Oard et al., 2008, for an overview). All three of these resources are covered in this work: we present a way of extracting a bilingual dictionary from a parallel corpus, and we also

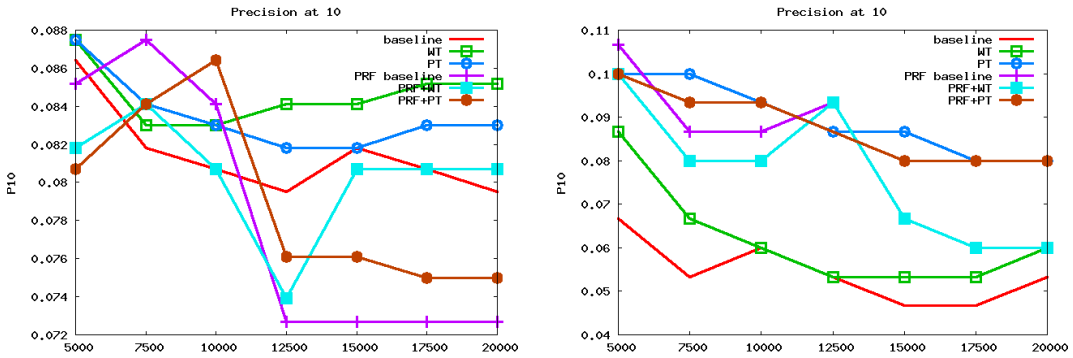

Figure 3.3: Dirichlet prior μ tuning for German (left) and French (right) versus P10 (y axis).

include experiments where translation is realized using Google's competitive MT system, *Google Translate*. Translating queries for cross-lingual and multilingual IR has been widely studied but comparatively little work has been done for multilingual patent IR. Magdy and Jones (2011a,b) do use machine translation to expand queries similarly to what we have done in Chapter 2 and their results also indicate that query translation for patents may help for French and German but less so for English.

Like query translation, query expansion using pseudo-relevance feedback has been heavily used by the larger IR community. A number of studies have applied it to patent retrieval as well, with mixed results. Magdy and Jones (2011a) look at standard PRF and conclude that it did not help patent retrieval (a finding that is also reported in (Mahdabi and Crestani, 2012)), while Bashir and Rauber (2009) claim that PRF does in fact improve patent *retrievability*.

In this chapter we add phrase translation to the previous approach using only term translations from word alignments. We construct (statistical) phrases following the approach of Koehn et al. (2003). Phrase translation has not been thoroughly investigated for patent retrieval. Mahdabi et al. (2012) use select noun phrases for query expansion, but this does not lead to sig-

nificant improvements, and their best results come instead from expanding queries with terms from relevant IPC classes (in conjunction with a classifier that selects which queries should be expanded). Wäschle and Riezler (2012a,b) look closely at machine translation in the patent domain but their work has not yet been applied to patent queries.

An early, well-cited phrase-based cross-lingual IR study was conducted by Ballesteros and Croft (1997), who expanded bilingual dictionaries with phrases and used them effectively in IR. Their definition of a phrase differs from the definition of phrase we use in this work. They used well-defined linguistic phrases coming from the Collins machine-readable translation dictionary, while we define phrases statistically as any string of words, meaning that no grammatical preprocessing is required and there is no dependency on manually compiled dictionaries. One interesting conclusion from their work is that, to improve retrieval, phrase translations must consistently be of high quality. Our results also suggest that this is the case as the phrase translation has the potential to improve query results but its behavior is more volatile than term translation. Their work has also reported positive results using translation and query expansion, however, their study uses older and hence lower baselines (from 1997); the same approach might produce different findings with current baselines.

3.5 Summary

In this chapter we continued our investigation of query translation and expansion for patent retrieval. We looked at two alternative translation methods, term translation and phrase translation. Our experimental evaluation showed good results for both, especially on hard queries. Phrase translation seems to be more beneficial for French than for German because German often uses single-term compounds instead of phrases, thus limiting the potential benefit of phrase to term and phrase to phrase translations.

Chapter 4

Citation Classification

In this chapter we address the task of citation classification and investigate different features used in a faceted classification scheme. Earlier chapters covered data collections that are rich in metadata (but where NLP techniques are still useful for recovering additional information). In the case of scientific literature, much less metadata is explicitly defined (i.e., "marked up"), but there is a great deal of descriptive and structural metadata: descriptive metadata includes the title, author, abstract, and bibliography; and structural metadata refers to the organization of the paper in sections, figures, tables, etc. Leveraging this information may improve search and retrieval capabilities as alluded to in Chapter 1.

Machine learning techniques have already been explored for recovering much of this metadata (Peng and McCallum, 2004; Councill et al., 2008). We focus on the citation metadata with the aim of attaching additional information to the individual citations. In particular, we want to label citations by their *citation function*.

The chapter is structured as follows. We begin in Section 4.1 with an introduction to citation classification and our motivation for why this task is important. Sections 4.2 and 4.3 cover the details of our annotation scheme and corpus, followed by a detailed description of the features used for classification in Section 4.4. Section 4.5 presents the classification experiments we conduct with a discussion of the results in Section 4.6. Section 4.7 discusses

related work. Finally, we close with a summary and an outline of future work in Section 4.8.

4.1 Motivation

Citations are a valuable resource for characterizing scientific publications and their links to each other. They have been exploited for a number of NLP and IR applications, including summarization (Qazvinian and Radev, 2008; Qazvinian et al., 2010), improved indexing and retrieval (Ritchie et al., 2006, 2008), and building integrated research databases (Nanba et al., 2004). Bibliometric measures that quantify the impact of publications (e.g., Moed, 2005) are also based on citations.

Most of this work does not differentiate between uses of citations, e.g., whether a citation is more or less important to the paper or whether the paper's authors support or refute the claims made in the cited work. However, recently a number of research groups have attempted to classify citations with respect to dimensions like importance and relation to cited work (Teufel et al., 2006b; Dong and Schäfer, 2011; Sugiyama et al., 2010; Abu-Jbara and Radev, 2012). By adding such fine-grained information to individual citations, the various applications of citation analysis can be better served; e.g., citations that are foundational to a paper may constitute better summary sentences for the cited paper.

Thus, there are clear potential benefits to fine-grained citation analysis; and a number of case studies have been published that demonstrate this potential (Nanba et al., 2004; Teufel et al., 2006b). However, fine-grained citation analysis has yet to be widely used in applications that access and analyze the scientific literature. In this chapter, we identify some potential reasons for this state of affairs and propose solutions.

The first problem with current fine-grained citation analysis is that prior work has tended to develop custom classification schemes for a particular application. This means that the development cycle for a citation classifier must be started from scratch for each new application. In contrast to this prior work, we base our work on a standard classification scheme for cita-

tions from information science, the classification scheme of Moravcsik and Murugesan (1975) (henceforth MM). We believe it is important to use an annotation scheme that is not bound to automatic citation classification for one particular task such as IR or bibliographic measures. Instead, it should be expressive enough to handle citations across many tasks. The MM scheme comprises four different dimensions or *facets*, which allows us to annotate the quality of the cited work along with its relation to the citing work. This gives the classification flexibility, so that it can be used in different application scenarios; e.g., some facets of the citation are more relevant for IR in digital libraries, while others are more useful in automatic document summarization.

The second reason that fine-grained citation analysis has not seen widespread adoption is that it remains a challenge to accurately and automatically classify citations according to a predefined classification scheme (Teufel et al., 2006b). We address this problem by introducing several novel features designed specifically for use in citation classification. Some of these new features are needed to support the more flexible and generic MM facet classification scheme. In particular, we extract novel features that capture the relationship between the citing paper and the cited paper. Identifying this relationship helps in understanding what motivated an author to reference the cited work. We also investigate how different features perform across the four facets, and how other variables, like the size of the context from which we extract features, affect the classification. We go on to compare different feature sets used for citation classification. In particular we compare different lexical, syntactic, and positional features. We aim to provide an extensive investigation of the comparative utility of features for citation analysis.

The final barrier to widespread adoption of fine-grained citation analysis is the fact that progress in the field has been hampered by the lack of a standard annotated corpus. Although all of the previous work we cover has used corpora of NLP articles for citation analysis experiments, none has tried reusing an existing corpus or annotation scheme. This makes accurately comparing results impossible, which in turn makes it difficult to gauge the advancement of the state of the art. Authors have focused on developing new annotation schemes, but no work has gone into building resources that

allow the research community to evaluate and compare different citation classification methods.

As shown in this chapter, previous results are difficult or impossible to reproduce because existing citation approaches have not been described in sufficient detail and resources created or used for the approach have not been published. To address the lack of reproducible experiments in citation classification, we have created, and made publicly available, a manually-annotated citation corpus. Additional information for replicating these experiments has been made available online[1] and is summarized in the appendices of this thesis (see Appendix B). We hope that this corpus can provide a benchmark for further advances in citation classification.

4.2 Annotation Scheme

In selecting our fine-grained classification scheme, we focused on two criteria. The first criterion is that we should consult the field of research that has the most expertise and the longest research record in developing classification schemes for citations. This field is information science. We have chosen the scheme proposed by Moravcsik and Murugesan (MM) because it adequately represents scientific literature for a broad range of citation classification scenarios. Furthermore, it is a well-established annotation scheme that is widely cited and used inside and outside of the information science community. A number of other classification schemes have been proposed for citation analysis in the last half-century, but we save our discussion of these for the Related Work in Section 4.7 and in this section only focus on the merits of the MM scheme we have chosen.

The second criterion for selecting the scheme was that it should be flexible and adaptable for different citation use cases. The MM scheme achieves this in that it is composed of four *independent* or *orthogonal* facets. For each facet, it assigns a label from a set of two labels. The scheme can be summarized with the four questions they posed:

[1] `http://www.ims.uni-stuttgart.de/~jochimcs/citation-classification/`

1. Is the reference conceptual or operational?
2. Is the reference organic or perfunctory?
3. Is the reference evolutionary or juxtapositional?
4. Is the reference confirmative or negational?

The *conceptual* vs. *operational* facet – **CONC-OP** – asks: "Is this an idea or a tool?," where examples of tools are MRI in brain imaging and part-of-speech (POS) taggers in NLP. The *organic* vs. *perfunctory* facet – **ORG-PERF** – distinguishes those citations that form the underpinnings of the citing work from more cursory citations. The *evolutionary* vs. *juxtapositional* facet – **EVOL-JUX** – highlights the relationship between the citing and cited papers. If the citing paper builds on the cited work, it is EVOL while it is JUX if it presents an alternative to the cited work. Finally, **CONF-NEG**, the *confirmative* vs. *negational* facet, captures the completeness and correctness of the cited work. A NEG citation usually is not derogatory, it may simply say that the cited work is weaker than the citing work or is otherwise missing some critical point. These distinctions are covered in more detail in the annotation guidelines (see Appendix B.1).

These four facets can be thought of as orthogonal dimensions along which citations can vary. This is the basis for flexible and adaptable citation analysis; e.g., a facet that is not relevant for a particular application can simply be omitted. If interactions between two facets are important for another application, they are made available by the citation classifier without complicating the model or its training.

Although there are now four facets to annotate for each citation instead of a single label, the annotation task is not more difficult. Making a binary decision is easier than trying to pick a label from ten possibilities with subtle differences between some of them. Yet, with the combination of different facets we still can achieve a finer-grained label. .

It is also important to note that this classification has no undefined class. Several previous annotation schemes have a default label, *neutral* or *other*, that is assigned to a citation when no other classes can be. In the work we

have seen that uses such annotation schemes, more than half of the citation instances are assigned this undefined label. In these cases, summarization or IR systems that want to make use of citation information obtain no useful information from the citation classifier for more than half of citations.

Summarizing, the MM scheme is the product of a dedicated study of citations and has not been developed for any particular task. Some other annotation schemes have been designed specifically for IR applications (e.g., Teufel et al., 2006b), and while this does not preclude them from being used for tasks such as summarization, they might not be as appropriate. More important is perhaps that flat annotation schemes, especially those with more classes, risk having classes that are difficult to distinguish between or may even overlap. With such an annotation scheme it is less straightforward how to define a search query or summary and decide which classes to include or exclude. With a faceted approach and one composed of binary decisions, one can build more intuitive and expressive queries or summaries.

4.3 Corpus

Our corpus, like corpora from some previous studies (Athar, 2011; Dong and Schäfer, 2011), is taken from NLP literature. Specifically, we have taken the 2004 ACL proceedings from the ACL Anthology Reference Corpus (ARC) (Bird et al., 2008). NLP literature was chosen because the annotators (NLP students) are more familiar with this data and can make more informed decisions when annotating the citations.

Preprocessing. We take the plain text files from the ACL ARC and use the sentence splitting of TreeTagger (Schmid, 1995), which marks the sentence boundaries with opening and closing XML tags. We then use a regular expression to find the citations in the text and similarly mark them up with XML, resulting in a very simple XML file with sentence and citation tags. When annotating and building the corpus, we simply add the annotation label as an attribute to the marked up citation.

Some statistics on the number of documents and citations in the corpus

can be found in Table 4.1 and the distribution of labels over the citations is in Table 4.2. Each citation in the corpus has been independently annotated by at least two of six annotators. Gold labels are chosen by a simple majority vote and ties are broken by an additional annotator when necessary. The annotators were given guidelines to help ensure consistent annotation. We built a browser-based annotation tool that displays the full text of the paper, so that the annotators can look at the wider context of the citation when necessary. In many cases the context necessary for annotation is only one sentence, but it will often span sentences or fill a paragraph.

Section	Documents	Citations
Main ACL	57	1668
Student	6	101
Poster/demo	21	239
Total	84	2008

Table 4.1: ACL 2004 corpus statistics.

As mentioned in Section 4.2, no facets were left undefined. This reduces the classification to only two classes and avoids a neutral class. For our purposes it is reasonable to avoid having a neutral class; e.g., a citation that is not explicitly CONF should still be implicitly considered CONF because including the citation is still an endorsement of the cited work.

Inter-annotator agreement on the corpus's annotation can be found in Table 4.3. These numbers indicate that the difficulty of the annotation task

conceptual (CONC)	1792	evolutionary (EVOL)	1804
operational (OP)	216	juxtapositional (JUX)	204
organic (ORG)	203	confirmative (CONF)	1836
perfunctory (PERF)	1805	negational (NEG)	172

Table 4.2: Distribution of annotated citations across the four facets.

	CONC-OP	EVOL-JUX	ORG-PERF	CONF-NEG
Agreement	0.86	0.88	0.72	0.91
Fleiss's κ	0.42	0.45	0.18	0.41

Table 4.3: Inter-annotator agreement using the observed agreement (percentage of instances on which annotators agree) and Fleiss's κ.

varies for the different facets, with ORG-PERF being most difficult.[2] Due to the highly skewed distribution, κ suffers from *prevalence* (Di Eugenio and Glass, 2004), yet three of the facets still have *moderate* agreement (according to Landis and Koch (1977)), and ORG-PERF has *slight* agreement. We feel that the observed agreement is high enough that we can rely on the gold labels for evaluation.

We released this annotated citation corpus along with our paper (Jochim and Schütze, 2012). To the best of our knowledge the corpus is the first to be annotated by individuals other than the study's authors. It is important to have independent annotators to limit any bias in the gold-standard annotation. One consequence of this is that our inter-annotator agreement scores are lower than those previously published, as the previous annotation came from the developers of the respective annotation schemes and from the authors reporting on the classification experiments using them.

4.4 Description of Features

Our goal is to accurately classify citations according to MM, the annotation scheme described in Section 4.2. We make the assumption that the necessary clues for correctly labeling citations, both manually and automatically, can be found in the context of the citation, i.e., the running text surrounding the citation. If we are able to extract the right clues from the citation context

[2]MM's definition was "is the reference truly needed for the understanding of the referring paper," so the annotation hinges on the understanding of the individual annotator, resulting in higher disagreement.

we can accurately label the citation's use.

Because there is not currently a standard corpus for the task of automatic citation classification, the results from others' previous work are difficult to compare. Previous studies have used different corpora, different annotation schemes, different feature sets, and different classifiers. In an effort to borrow from – and eventually compare ourselves to – previous work, we investigate some of the features used previously and introduce our own. The reader may want to refer to the overview of features in Table 4.4 as we describe the features in what follows.

Table 4.4: Feature list (grouped by feature class). NO99=Nanba and Okumura (1999); TST06=Teufel et al. (2006b); Ath11=Athar (2011); DS11=Dong and Schäfer (2011). "unknown" = exact definition of the feature (e.g., Boolean or Real) is unknown. Examples of possible feature values are given in italics where appropriate.

feature class	name	source	type or *example value*	description
lexical features	`cues`$_k$	NO99, TST06, DS11	Boolean	k Boolean features: one for each group of cue words/phrases
	`1-gram`	Ath11, own	*hard*	unigrams
	`1+2-gram`	Ath11	*hard language*	unigrams & bigrams
	`1+2+3-gram`	Ath11	*hard language like*	unigrams, bigrams, & trigrams
word-level linguistic feats.	`POS`	Ath11	*NN, JJ, IN*	POS tags
	`1-gram+POS`	Ath11	*quality+NN, new+JJ*	POS tag-word conjunctions
	`tense`	TST06	*present, past*	verbal tense
	`voice`	TST06	*active, passive*	verbal voice
	`modal`	TST06	*can, may*	modal verb (if any)
	`has-modal`	own	Boolean	sentence has modal verb
	`root`	own	*have, present*	dependency root node
	`main-verb`	own	*present, use*	main verb

Continued on next page

Table 4.4: (continued)

feature	name	source	type	description
	`has-1stPRP`	own	Boolean	first person POS
	`has-3rdPRP`	own	Boolean	third person POS
	`comp/sup`	own	*more, better*	comparative/superlative POS
	`but`	own	Boolean	has "but"
	`has-cf`	own	Boolean	has "cf."
linguistic structure features	`dep-rel`	Ath11	*pobj:to:information*	Stanford typed dependencies (de Marneffe et al., 2006)
	`POS-pattern`$_k$	DS11	Boolean	k Boolean features: one for each POS tag pattern
	`is-constituent`	own	Boolean	citation is a constituent
	`self-comp`	own	Boolean	author linked to comparative
	`other-comp`	own	Boolean	citation linked to comparative
	`other-contrast`	own	Boolean	citation is in contrastive clause
	`self-good`	own	Boolean	author linked to positive sentiment
location features	`section`	DS11	*Introduction, Method*	1 of 6 possible section headings
	`paper-loc`	TST06	unknown	citation position in paper
	`paragraph-loc`	TST06	unknown	citation position in paragraph
	`section-loc`	TST06	unknown	citation position in section
	`sentence-loc`	own	*beginning, middle, end*	location in the first quarter, middle half (25%-75%), and last quarter

Continued on next page

Table 4.4: (continued)

feature	name	source	type	description
frequency features	`popularity`	DS11	Integer	citations in the same sentence
	`density`	DS11	Integer	citations in the same context (sentence and its neighbors)
	`avgDensity`	DS11	Real	average `density` of neighboring sentences
sentiment features	`scilex`	Ath11	unknown	scientific polarity lexicon
	`cpol`	Ath11	unknown	general polarity lexicon
	`positive-words`	own	*best, advantage*	general positive lexicon
	`negative-words`	own	*problem, against*	general negative lexicon
other features	`self-cite`	TST06	Boolean	citation to own work
	`has-resource`	own	Boolean	resource entity found with NER
	`has-tool`	own	Boolean	tool entity found with NER

Lexical features. Much of the earlier work on automatic citation classification (Dong and Schäfer, 2011; Nanba and Okumura, 1999; Teufel et al., 2006b) relied on cue words and phrases ($\texttt{cues}_k$). These were often implemented as follows. For a class (e.g., Dong and Schäfer's *idea*), a list of cues (e.g., *following*, *similar to*, *motivate*) are defined that indicate that particular class. Finally, a Boolean feature (e.g., $\texttt{cues}_{idea}$) is set to true if any word from the list is in the citing context. This results in k Boolean features where k is often the number of classification labels (although it can be greater, e.g., Dong and Schäfer (2011)).

Different length n-grams were used by Athar (2011) with results indicating that combined unigram, bigram, and trigram features (`1+2+3-gram`) performed better than unigrams (`1-gram`) and unigrams plus bigrams (`1+2-gram`).

We use only unigrams because they perform at least as well as using unigrams, bigrams and trigrams in our experiments, without introducing a much larger, sparsely-populated feature set. Unigrams should also be quite robust and perform reasonably well across the four facets.

Word-level linguistic features. Part-of-speech (POS) tags of the words in the citation sentence were used as features by Athar (2011) (`POS` and `1-gram+POS`). Select linguistic features related only to the main verb were shown to be effective by Teufel et al. (2006b), e.g., tense (`tense`), voice (`voice`), and modality (`modal`).

We also include modality in our feature set (`has-modal`) along with separate features for the main verb (`main-verb`) and the root (`root`) as determined by the MATE dependency parser (Bohnet, 2010). We do not include POS as features per se, but some features are triggered by the occurrence of selected POS: first- and third-person pronouns (`has-1stPRP`, `has-3rdPRP`); and comparatives and superlatives (`comp/sup`). Comparatives and superlatives can help distinguish CONF from NEG. Pronouns on the other hand may be useful in classifying EVOL-JUX, e.g., first-person pronouns are used when clarifying the differences between proposed and cited approaches.

We include two other features for the contrastive conjunction "but" (`but`) and the abbreviation "cf." (`has-cf`). In our analysis of citations we looked at the role of contrastive conjunctions in citation sentences and found these simple features to be useful.

Linguistic structure features. Dependency relations (`dep-rel`) were used as features and showed a marked improvement over the baseline by Athar (2011). Dong and Schäfer (2011) used seven regular expression patterns over POS tags (`POS-pattern`$_k$) to capture syntactic information (e.g., ".*(VHP|VHZ) VV.*"); then $k=7$ Boolean features marked the presence (or absence) of these patterns.

We add other new features related to the linguistic structure of the citation sentence. For `is-constituent`, the citation is labeled as a constituent if the authors appear outside of the parentheses with only the date in paren-

theses, e.g., "Gusfield (1997) showed that ...", or if the citation acts as a placeholder for the cited work following a preposition, e.g., "... following the experiments in (Kaplan et al., 2004)." These cases are distinguished from citations like: "... are two popular examples of kernel methods (Fukunaga, 1990; Cortes and Vapnik, 1995)." We are relying here on a certain style of writing and citation format, like that found in ACL proceedings. We expect this feature to help for ORG-PERF as organic citations are more likely to show up as constituents in citation sentences.

The personal pronoun and comparative features mentioned above (`has-1stPRP`, `has-3rdPRP`, and `comp/sup`) are useful features alone, but we would like to extract a more specific feature that links them. We want features that indicate that the citing work has improved upon the cited work, e.g., "We obtain better results than (Hill et al., 2003)." To obtain these features we parse the sentence with MATE and extract relations from the parse tree. For the author/comparative relation, we first find the comparative in the sentence (e.g., *better*) and traverse the tree to find the subject of the phrase that contains that comparative. If the subject refers to the author of the paper (e.g., with a first person pronoun), we set the `self-comp` feature to true. This is shown in Figure 4.1 where the arcs illustrate the dependencies linking the first person pronoun subject "we" and the comparative "better". "Better" modifies the noun "results", and "we" and "results" are the subject and object, respectively, of "obtain". The other 'quality' citation relation features (`other-comp` and `self-good`) are extracted in a similar way.

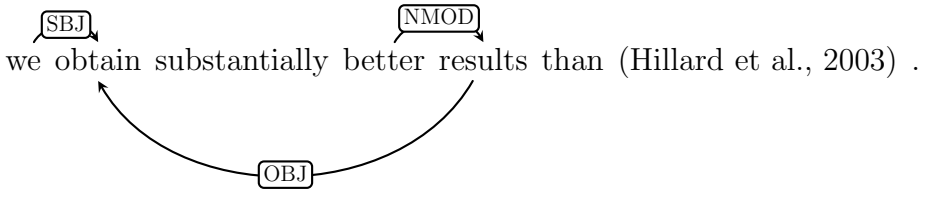

Figure 4.1: Dependency relations traversed for capturing `self-comp` feature.

We also found that JUX citations are often set apart using contrastive conjunctions (listed in Table 4.5), e.g., "**While** Johnson (2001) proposes a model that ..." We again traverse the parse tree to extract the relationship

Single word	Multi-word
although	in spite of
unlike	even though
though	even if
despite	all the same
notwithstanding	at the same time
whereas	in contrast
while	on the contrary
however	even so
instead	compared to
nevertheless	contrary to
yet	
still	

Table 4.5: Contrastive conjunctions.

between contrastive conjunctions and the citation (`other-contrast`), where the citation or cited authors show up in the dependent clause governed by the contrastive conjunction. This is shown in Figure 4.2 where the cited author "Johnson" is the subject of the clause with the verb "proposes", and "While" is the head of "proposes". In other words, the feature is set to true if the citation is found among the descendants of the contrastive conjunction.

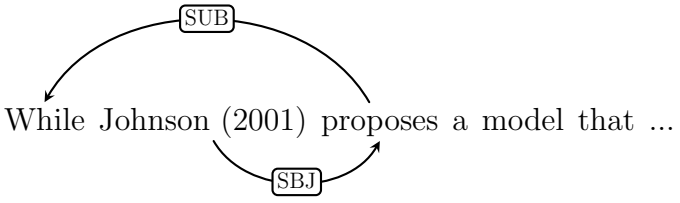

Figure 4.2: Dependency relations traversed for capturing `other-contrast` feature.

Location features. The section of the paper in which the citation is located (`section`) was used as a feature by Dong and Schäfer (2011). Teufel et al. (2006b) also included location features at different granularities: within

the paper (`paper-loc`), within the paragraph (`paragraph-loc`), and within the section (`section-loc`).

We include a different location feature approximating where the citation is found in the sentence (`sentence-loc`): beginning, middle, or end. This feature is motivated by the fact that citations at the end of the sentence are predominantly PERF.

Frequency features. Dong and Schäfer (2011) used the number of citations in a single sentence (`popularity`) and in the citation sentence plus its neighboring sentences (`density`) as features. They also included a third feature for the average density of neighboring sentences (`avgDensity`).

Sentiment features. Athar (2011) included two different polarity lexicons. One is hand-crafted and specific to the scientific domain (`scilex`). The other is the large general purpose polarity lexicon from Wilson et al. (2005) (`cpol`). He also tried features (`neg`) that account for negation. This was done by appending "_neg" to the end of the 15 lexical items that follow any negation term.

We were not able to obtain the scientific polarity lexicon, but we do use the polarity lexicon from Wilson et al. (2005) to extract sentiment features. Our polarity features are represented as a bag of words (BOW) where the citation context words present in the polarity lexicon are added to the BOW features `positive-words` or `negative-words` according to their polarity. Although CONF-NEG is not strictly a matter of sentiment, we still apply this feature hoping for improvements on this facet.

Using either of these polarity lexicons can be problematic. In the case of the general lexicon there is bound to be additional noise because the polarity within a given context or domain may differ from the more general polarity. On the other hand, using a specific hand-crafted lexicon extracted from one domain (e.g., NLP) could lead to problems when trying to apply this feature to literature from other domains.

Self-reference feature. Teufel et al. (2006b) used a feature, `self-cite`,

that indicates if one of the citing authors also (co-)authored the cited work. This feature is unique in that it is the only feature based on the reference and not the individual citation and therefore not taken from the context in which it is found.

NER features. Using lexical features alone, there are a number of words that help indicate OP (*operational*) citations in NLP, e.g., "parser", "tagger", "corpus". We decide to take this a step further and train a named-entity recognition (NER) system to identify NLP named entities. We identify two types of NLP named entities: corpora and tools. First, we create a gazetteer of NLP tools and corpora from an online list of these resources.[3] Next, we tag a portion of our corpus using the gazetteer list to label any occurrence of the words in the list and then manually check those labeled instances to be sure they are correctly labeled. In this way we can expediently create training data, with an emphasis on precision over recall. Finally, we train the SuperSenseTagger (Ciaramita and Altun, 2006) on this annotated portion, and tag the remaining part of the corpus. NER is not central to our task, so we did no direct evaluation of it; we looked only to see if it might lead to improvements in our classification. We include two features, `has-resource` and `has-tool`, for the two types of entities.

The NER features we extract are related only to the NLP domain. However, this approach for acquiring named entities is not domain dependent and can be used to develop a reasonably efficient NER system using lists of tools or resources from any domain.

4.5 Experiments

In this section we outline our classification experiments and then discuss the results in Section 4.6. We use the term *feature set* to describe a collection of features used by us or in previous studies; we use the term *feature class* to describe a collection of similar features as they are organized in Section 4.4 and in Table 4.4.

[3] `http://nlp.stanford.edu/links/statnlp.html`

Setup. All our experiments were conducted on the corpus described in Section 4.3. We trained the Stanford MaxEnt classifier (Manning and Klein, 2003) for each of the four facets in a 5-fold cross validation setup with default settings except that we set the regularization parameter $\sigma = 10$ based on previous experiments.

4.5.1 Feature Set Comparison

In our first set of experiments we test our own feature set and the feature sets described in previous studies. Each of these feature sets is a subset of the features described in Section 4.4 and is identified below by some of its more distinguishing features; e.g., **NgramDep** refers to the feature set that mainly uses n-grams and dependencies.

The **CueVerbLoc** feature set is intended to mimic (Teufel et al., 2006b) to the extent this is possible. It includes cue phrase features ($\texttt{cues}_k$), the verbal features `tense`, `voice`, and `modal` as well as `paper-loc`. The cue phrases used in (Teufel et al., 2006b) are not available so we applied automatic feature selection using mutual information (MI) (Manning et al., 2008) to select the most informative unigrams, bigrams, and trigrams for each class label. We borrow from the manual feature selection in (Teufel et al., 2006b) by assigning cue phrases to each of the labels (8 in our case – Teufel et al. used 12) and limiting the number of cue words to 75 per label. Some examples for OP cues are *penn treebank*, *wordnet*, and *parser*. The full list of cue phrases obtained using this approach can be found in Appendix B.2.

The **NGramDep** feature set corresponds to (Athar, 2011). It includes lexical features: unigrams, bigrams, and trigrams (`1+2+3-grams`) and the `dep-rel` features. Athar (2011) tested other features, but we have only reimplemented those that improved results.

The **CueFreqPOS** feature set is based on (Dong and Schäfer, 2011). It includes a list of cue words ($\texttt{cues}_k$), then the frequency features `popularity`, `density`, `avgDensity`, and the syntactic feature $\texttt{POS-pattern}_k$.

The feature set **OWN** includes all the features we have introduced in our work – those marked "own" in Table 4.4. Some features were designed to

help one facet or another, but we use them all together here for all facets.

The **PREV** feature set combines all features previously used for citation classification (i.e., CueVerbLoc + NgramDep + CueFreqPOS) and **ALL** includes all of the features we have (re)implemented and described in this chapter.

We note here that by reimplementing features from previous work we claim only to extract the same or similar information as the original authors. Due to sometimes major differences in the corpus, annotation scheme, and classifier used, we are not able to reproduce the same conditions that led to previous results. We are instead more interested in the types of features that seem to perform best on our dataset with our annotation scheme.

4.5.2 Citation Context Size

In the feature set comparison just described, the experiments are run using a fixed context window of one sentence. It is not clear how much context is best for feature extraction, so we conducted some additional experiments in which we fix the feature set and test the features extracted from different sized context windows. We would like to discover how much context surrounding the citation is best for extracting features. In previous work different sized context windows were used by different studies, e.g., Athar (2011) used only the sentence containing the citation while Dong and Schäfer (2011) used up to three sentences, the citation sentence plus the sentence before and after. Kaplan et al. (2009) and Abu-Jbara and Radev (2012) have illustrated the difficulties in delineating the exact boundary of the citation context. Athar and Teufel (2012) used different sized context windows for citation classification, and a similar task in IR aims to find the right amount of context for adding to a retrieval index (Ritchie et al., 2008). Initially, we follow this general idea and test context lengths of 1, 2, and 3 sentences.

4.5.3 Feature Class Comparison

After comparing our own feature set with those from previous work, it is interesting to investigate what feature classes assist most in the classification.

We perform this analysis by examining the impact of the seven feature classes described in Table 4.4. More specifically, we compare the results of their individual performance using *only features in the feature class*, and their ablation from the entire feature set using *all features except those in the feature class*. Finally, we extend the ablation study, *successively removing all feature classes* in order of importance (i.e., by their contribution to F_1 score).

4.5.4 DFKI Citation Dataset

Dong and Schäfer (2011) made their annotation publicly available[4] following the publication of their paper. Their dataset uses papers from the 2007 and 2008 ACL proceedings in the ACL Anthology. The dataset includes 1768 citing sentences with three levels of annotation. The four labels used in their paper are a combination of two of these levels. The first two levels capture citation function and the third marks the sentiment of the citation (i.e., *positive*, *negative*, or *neutral*).

We have not limited our experiments to only their dataset (which we refer to as the DFKI citation dataset) for several reasons, but principally because (i) our corpus annotation was well underway at the time they released the data, and (ii) the dataset they released is not the full corpus used in (Dong and Schäfer, 2011). For our task, there are a few drawbacks in using the DFKI citation dataset. First, the DFKI dataset only contains the citing sentence along with the annotation and not the full corpus, so it is impossible to extract all the features described in (Dong and Schäfer, 2011) as some features come from a wider context. Second, the annotation in the DFKI dataset is done on the citing sentence and not the actual citation. This is perhaps a subtle difference, but in an example like, "Following (Chiang, 2005), we used ... to calculate the BLEU scores (Papineni et al., 2002)," a single label, *Idea*,[5] covers both citations, even if the Papineni citation probably did not "inspire" the citing work. We would like these citations to be distinguished, and in fact,

[4] `https://aclbib.opendfki.de/repos/trunk/citation_classification_dataset/`

[5] *Idea* is defined in the DFKI annotation guidelines for when "The citation sentence refers to work which inspired the idea of the current work."

the annotation in our corpus is done on the citation and not the sentence.

Nevertheless, it is important that we conduct experiments on other previously-tested, publicly-available data. We conduct three experiments using the annotation from Dong and Schäfer (i.e., with the labels, *Background*, *Idea*, *Basis*, and *Compare*) attempting to come as close to replicating those in (Dong and Schäfer, 2011) as is possible. The first experiment uses the DFKI citation dataset as is, and only uses the bag-of-word features (`1-gram`) available in the citing sentence. The remaining two experiments use the preprocessing and feature extraction steps that we apply to our own corpus (see Section 4.3) on the ACL Anthology data used by Dong and Schäfer (i.e., ACL proceedings from 2007 and 2008). One of these experiments uses the Dong and Schäfer features (CueFreqPOS) and the other uses our own features (OWN). The first experiment differs from these latter two as there is a mismatch in what is classified due to the DFKI annotation of citing sentences and our annotation of citations.

4.6 Results and Discussion

4.6.1 Feature Set Results

The results for the different feature sets when using one sentence of context are found in Table 4.6. All of the F_1 results presented in this paper are macro-averaged F_1. We have included three baseline experiments. We use a majority baseline (BL) that labels each citation with the label occurring most often in the corpus, e.g., for CONC-OP, all citations are labeled CONC. We also include results for unigram, bigram, and trigram features (Ngram), which is the baseline used by Athar (2011). Finally, we include a BOW baseline because those features form the basis of the OWN feature set. The results in Table 4.6 show that our feature combination, OWN, outperforms all three baselines and all reimplemented previous feature sets for all four facets. With a few exceptions (BOW for all facets; Ngram for EVOL-JUX; and PREV for ORG-PERF), these results are significant.[6] Combining all features (All)

[6] $p < .05$ using the approximate randomization test (Noreen, 1989).

performs best – outperforming OWN – for the facets EVOL-JUX and ORG-PERF.

The greatest improvement over the baseline is with the OWN features for CONC-OP. Several of the other feature sets also do better on CONC-OP than BL, but OWN is still significantly better than PREV, the combination of all previous feature sets. Simple BOW features along with our new features (e.g., `has-resource` and `has-tool`) increase F_1 by 7 points over PREV. As an example, in a sentence citing "The Penn TreeBank (Marcus et al., 1993)," the citation is incorrectly classified using PREV. The NER tool recognizes Penn TreeBank as a corpus, which results in the OWN feature `has-resource` to be set to true and a correct classification of the citation as OP.

EVOL-JUX proves to be more difficult than CONC-OP with either no or very small improvements over BL for all feature sets except for Ngram and OWN. The BOW features from our OWN feature set are responsible for most of the improvement of F_1 from 47.3 to 52.9. BOW features contribute to the improvement with OWN features for all four facets.

OWN features improve F_1 by 10.7 (from 47.3 to 58.0) over the BL for ORG-PERF, and are also better by 3.2 (54.8 vs. 58.0) than PREV. Some features that contribute to the better results are `root` and `main-verb` with values such as "describe" and "present"; these appear to be useful in identifying ORG citations. In this facet, the feature set CueFreqPOS sees its most significant improvement over BL. This is due in a large part to the frequency features that are not found in other feature sets.

Finally, CONF-NEG is the most difficult facet. All feature sets except our own performed only as well as or even worse than BL. OWN features improve F_1 by 3.3 (from 47.8 to 51.1), which is due in part to the location feature that finds citations in the middle of sentences to be CONF, while NEG citations are more likely to come at the beginning. Like the other three facets, OWN again performs better than all other feature sets.

To get an idea of a possible upper bound for this task, we include a *human classifier* ("Human" in Table 4.6): we take the annotation from the most experienced annotator and consider it as classification output. CONC-OP is the "easiest" facet for the human classifier to label, similar to automatic

	CONC-OP	EVOL-JUX	ORG-PERF	CONF-NEG
baseline (BL)	47.2**	47.3**	47.3**	47.8**
Ngram	53.2**	50.7	51.3**	47.8**
BOW	66.9	52.5	57.2	50.1
CueFreqPOS	48.4**	49.4*	54.1*	47.7**
NgramDep	53.3**	47.3**	50.5**	47.8**
CueVerbLoc	51.1**	47.3**	47.3**	47.8**
PREV	61.2**	48.5**	54.8	47.5**
OWN	**68.2**	52.9	58.0	**51.1**
All	64.5**	**53.4**	**59.2**	48.9*
Human	94.7	91.1	91.7	93.5

Table 4.6: Macro-F_1 for different feature sets. Marked with $*/**$: significantly worse than OWN ($p < .05/.01$). Bold: best performing feature set per facet. "Human" uses annotation from one annotator to simulate classification output.

classification. However, the most difficult facet for automatic classification, CONF-NEG, appears to be straightforward for the human classifier. This is consistent with the high observed agreement for CONF-NEG (.91, Table 4.3).

4.6.2 Context Size Results

For the context size experiments we fixed the feature set, using OWN, and tested different sized windows of context: one sentence – using only the sentence containing the citation; two sentences – using the citation sentence and following sentence; and three sentences – using the citation sentence and two following sentences.

The results for the different context lengths (found in Table 4.7) show that the clues for classification may be more or less local depending on the facet. We expected that with a restricted window of context there would be cases where the features that make the citation NEG for example, are found outside the context. In fact, these results indicate that taking features from more than one sentence performs better for the CONF-NEG and also

the EVOL-JUX facets. This result coincides with the feedback from some of the human annotators who thought that these two facets required more context to annotate well. CONC-OP and ORG-PERF have higher F_1 scores with only one sentence. These results coincide more with those of Athar and Teufel (2012) where F_1 is greatest with the least amount of surrounding context. However, the only significant differences we found between context lengths is for CONC-OP, where results for both one and two sentences of context are significantly better than for three sentences of context; and for CONF-NEG where F_1 for two sentences of context is significantly better than for one sentence of context. These results suggest that context size is an important factor, but one that does not have a uniform effect on the four facets.

Ideally, we would like to identify the citation context boundaries in a more dynamic manner with techniques similar to those of Abu-Jbara and Radev (2012). In this way, we could extract features from the true context of a citation, whether it be one or more sentences.

	CONC-OP	EVOL-JUX	ORG-PERF	CONF-NEG
1 sentence	**68.2**^{c}	52.9	**58.0**	51.1
2 sentence	65.4^c	**55.1**	56.0	**54.5**^{a}
3 sentence	61.9	53.9	55.0	53.3

Table 4.7: Macro-F_1 scores for different context lengths using OWN feature set. Bold: best performing context size per facet. The results marked with a (or c) are significantly better than the context of 1 (or 3) sentence(s) ($p < .05$).

4.6.3 Feature Class Results

In the discussion of the feature class results we will refer to Tables 4.8, 4.9, and 4.10 and their line numbers (1–28). Table 4.8 presents F_1 results using only a single feature class (lines 1–7); Table 4.9 shows F_1 using all features ("All") and F_1 using all features except the listed feature class (lines 8–14); and finally, extended ablation results are given in Table 4.10, where a feature

class is successively removed from "All" (seven classes) until one feature class remains (lines 15–28). Our goal is to get a better idea of which feature classes are informative for a given facet.

Results for CONC-OP. LEXICAL features appear to be the most important for this facet. Alone they do well against the baseline (61.6 vs. 47.2, Table 4.8 line 1) and when removed from the entire feature set F_1 drops more than for any other feature class (from 64.5 to 58.2, Table 4.9 line 8). Both of these Δ's are significant. The feature class NER has the second highest F_1 (54.1, line 7) when used alone, which makes sense as it was designed for this facet. Removing NER features hurts F_1 (down to 64.0, line 14), but not significantly. Using only WORD-LEVEL or STRUCTURE features also leads to significant improvement: increases of 4.8 and 4.3 (lines 2 and 3 in Table 4.8). After that, SENTIMENT features improve F_1 but not significantly (line 6), while the LOCATION and FREQUENCY features show no difference from the BL (lines 4–5). The ablation results show that after the significant contributions of the LEXICAL features, the removal of other feature classes does not affect the results much: Removing STRUCTURE, LOCATION, and SENTIMENT features actually increases F_1 (Table 4.9 lines 10, 11, 13), and although WORD-LEVEL, FREQUENCY, and NER features seem to contribute somewhat to the entire feature set, their ablation shows no significant change (lines 9, 12, 14).

Results for EVOL-JUX. For this facet, three of the seven feature classes, LOCATION, FREQUENCY, and NER, lead to no change from the baseline when run alone (Table 4.8 lines 4, 5, 7). Another three feature classes, LEXICAL, WORD-LEVEL, and SENTIMENT, significantly improve over BL (lines 1, 2, 6). Conversely, the FREQUENCY features, with no improvement alone, help improve results of the entire feature set; when those features are removed, F_1 drops by 2.2 (from 53.4 to 51.2, Table 4.9 line 12). Also, the SENTIMENT features, which do well against the baseline (line 6), hurt F_1 when added to the full feature set (decrease from 54.1 to 53.4, Table 4.9 line 13).

	Feat. class	CONC-OP		EVOL-JUX		ORG-PERF		CONF-NEG	
	BL	47.2		47.3		47.3		47.8	
		F_1	Δ BL	F_1	Δ BL	F_1	Δ BL	F_1	Δ BL
1	LEXICAL	**61.6**†	**14.4**†	**52.7**†	**5.4**†	**56.1**†	**8.8**†	47.7	0.0
2	WORD-LEVEL	52.0†	4.8†	52.4†	5.0†	51.6†	4.2†	49.7	2.0
3	STRUCTURE	51.5†	4.3†	48.8	1.5	52.0†	4.7†	47.8	0.0
4	LOCATION	47.2	0.0	47.3	0.0	47.3	0.0	47.8	0.0
5	FREQUENCY	47.2	0.0	47.3	0.0	47.3	0.0	47.8	0.0
6	SENTIMENT	48.0	0.9	52.7†	5.3†	47.2	-0.1	**49.9**†	**2.1**†
7	NER	54.1†	7.0†	47.3	0.0	47.3	0.0	47.8	0.0

Table 4.8: Macro-F_1 results when a single feature class is used. Marked with †: significantly better than BL ($p < .05$). Bold: best performing feature class per facet.

		CONC-OP		EVOL-JUX		ORG-PERF		CONF-NEG	
	All	64.5		53.4		59.2		48.9	
		F_1	Δ All	F_1	Δ All	F_1	Δ All	F_1	Δ All
8	LEXICAL	**58.2***	**6.2***	53.3	0.1	60.2	-1.0	49.5	-0.6
9	WORD-LEVEL	64.0	0.4	53.6	-0.3	59.3	-0.1	48.9	0.1
10	STRUCTURE	66.7	-2.2	53.1	0.3	59.5	-0.3	**48.8***	**0.1***
11	LOCATION	65.0	-0.5	53.2	0.1	**55.8***	**3.4***	49.6	-0.6
12	FREQUENCY	64.4	0.1	**51.2**	**2.2**	58.3	0.9	49.8	-0.9
13	SENTIMENT	65.0	-0.5	54.1	-0.7	59.2	0.0	49.0	0.0
14	NER	64.0	0.4	53.7	-0.3	58.8	0.4	49.4	-0.5

Table 4.9: Ablation results: Macro-F_1 and decrease in macro-F_1 when each feature class is ablated; i.e., each result shown is a classification result using six feature classes. Marked with $*$: significantly lower than All ($p < .05$). Bold: best performing feature class per facet.

Results for ORG-PERF. Individually, the feature classes LEXICAL, WORD-LEVEL, and STRUCTURE all had significant improvements (lines 1–3 in Table 4.8). The other four classes do not help for this facet (lines 4–7). However, in the ablation results, omitting these feature classes also *increases* F_1 (Table 4.9 lines 8–10). Only removing LOCATION significantly decreases F_1 (line 11). This result indicates that several of the feature classes are correlated for classifying this facet. They contain useful information for the task (as indicated by good performance when used individually), but mutual correlation has the effect of bad generalization when all of them are used together. The results show that this type of analysis (which has not been performed before for citation classification) is important to understand how features impact performance and what steps are needed to achieve better performance.

Results for CONF-NEG. For this facet, only SENTIMENT and WORD-LEVEL (Table 4.8 line 2 and 6) improve over BL, and the remaining five feature classes do only as well as BL. Removing four of the seven feature classes actually seems to improve F_1 (lines 8, 11, 12, and 14 in Table 4.9), with F_1 only increasing by adding WORD-LEVEL or STRUCTURE features (lines 9–10).[7] In fact, it seems that including the feature classes LEXICAL, STRUCTURE, LOCATION, FREQUENCY, and NER might only be detrimental for this facet, as F_1 using only SENTIMENT features is 49.9 (Table 4.8 line 6) compared to using all features at 48.9 (Table 4.9 "All").

To further analyze the relative importance of a feature class for a facet we extend the ablation results by successively removing that feature class whose removal results in the lowest F_1, among the possible ablations, until all have been removed. These results are shown in Table 4.10. As an example, in CONC-OP we start with all features ($F_1 = 64.5$) and calculate F_1 after removing each of the feature classes individually. In this case, removing LEXICAL leads to the largest drop in F_1, from 64.5 to 58.2 (Table 4.10 line 15). In the next iteration, we again compare F_1 after removing each of the six remaining feature classes. Removing NER features results in the lowest F_1

[7] Lines 1 and 6 have the same F_1 (52.7) but different Δ and Lines 9–10 have different F_1 (48.9 vs. 48.8) but the same $\Delta = 0.1$ due to rounding.

	CONC-OP		EVOL-JUX	
	All	64.5	All	53.4
15	LEXICAL	58.2*	FREQUENCY	51.2
16	NER	53.6*	STRUCTURE	50.3*
17	STRUCTURE	50.3*	LEXICAL	50.6
18	WORD-LEVEL	47.9*	NER	50.7
19	SENTIMENT	47.2*	LOCATION	52.7
20	FREQUENCY	47.2*	SENTIMENT	52.4
21	LOCATION	47.2*	WORD-LEVEL	47.3*
	ORG-PERF		CONF-NEG	
	All	59.2	All	48.9
22	LOCATION	55.8*	STRUCTURE	48.8*
23	LEXICAL	55.4*	WORD-LEVEL	48.2
24	STRUCTURE	53.0*	LOCATION	47.4
25	WORD-LEVEL	48.6*	NER	47.4
26	SENTIMENT	47.3*	SENTIMENT	47.4
27	FREQUENCY	47.3*	FREQUENCY	47.7
28	NER	47.3*	LEXICAL	47.8

Table 4.10: Extended ablation results: Left columns indicate the feature class removed and right columns show macro-F_1 results. Marked with $*$: significantly lower than All ($p < .05$).

(now 53.6, line 16), and we proceed by removing one of the five remaining feature classes, etc. These results reaffirm what was discussed in Tables 4.8 and 4.9, but present it as a list of the feature classes in descending order of importance. This table helps us to compare different facets; we can easily see that LEXICAL and NER features are important for CONC-OP, while LOCATION features are not. Compare this to CONF-NEG where LEXICAL and NER features are not important and WORD-LEVEL is higher in the list. Note also, that F_1 does not always decrease (e.g., removing LEXICAL for EVOL-JUX). Some combinations of subsets of features will perform better than the previous superset. In this case, we see that after having removed the STRUCTURE features, removing any other feature class can only improve results.

The results of our feature class experiments give us some valuable insight into how to design features for citation classification. First, we consider the first three feature classes, LEXICAL, WORD-LEVEL, and STRUCTURE. All three contain quite general text classification features, and consequently are quite robust and informative across the four facets of citations that we consider. WORD-LEVEL seems robust in that it is the only feature class, in Table 4.8, with positive Δ BL for all four facets, while LEXICAL has the largest Δ BL values for three of the four facets (i.e., CONC-OP, EVOL-JUX, and ORG-PERF). The last four feature classes – LOCATION, FREQUENCY, SENTIMENT, NER – represent different citation features which seem to impact certain citation facets. NER was designed particularly for CONC-OP and does in fact contribute most to that facet; LOCATION helps only ORG-PERF (i.e., the position of the citation indicates its importance) where it contributes significantly to a combination of features; similarly FREQUENCY contributes significantly to a combination of features for EVOL-JUX; and finally, SENTIMENT is important for EVOL-JUX and CONF-NEG, as was expected. There is no single feature class that is the most important for all facets, which lends credence to the claim that these facets capture different properties of citations. We conclude that our multi-faceted scheme benefits from a diverse feature set and that although general, easily-extractable features help classification more consistently, the extraction of more specific features is important for improvements on certain classification tasks.

4.6.4 DFKI Dataset Results

The results from experiments with the DFKI citation dataset are in Table 4.11. The first experiment, DFKI-BOW, simply uses the bag-of-word features from the citing sentence taken directly from the DFKI dataset. Unfortunately, not all of the features they describe in their paper can be recovered using the published DFKI dataset. To replicate their features more completely, we preprocessed the same papers as Dong and Schäfer (2011), taken from ACL 2007 and 2008 in the ACL Anthology Network (Radev et al., 2009), and added the annotation from the DFKI dataset. The problem with this approach is that our preprocessing handles citation annotation and the DFKI dataset is annotated by (citing) sentence. We resolve this by assigning each citation in the sentence with the label from the DFKI dataset. This results in 1872 annotated citations in the experiments DFKI-CueFreqPOS and DFKI-OWN versus 1768 annotated sentences in DFKI-BOW.

The results for DFKI-BOW cannot be compared to those in DFKI-CueFreqPOS or DFKI-OWN. We see that our feature set OWN outperforms the CueFreqPOS feature set on this dataset as well. The F_1 scores using this annotation fall in the same range as F_1 results using our annotated citation corpus (see Table 4.6 above). Even without having an adequate mapping between the DFKI annotation scheme and the MM annotation scheme, it at least seems as though our annotation scheme can be used for automatic classification to achieve results comparable to those obtained with another recent annotation scheme (cf. 60.7, OWN in Table 4.11 and 68.2, 52.9, 58.0, 51.1, OWN in Table 4.6).

4.7 Related Work

The vast literature on citation analysis goes back a half-century and spans a number of different disciplines (e.g., applied linguistics and information science; see White (2004) for an introduction to the relationship between them). The tasks that we are interested in are more related to information science and Moed (2005) provides a thorough look at citation analysis from

	Background	Idea	Basis	Compare	Macro-F_1
DFKI-BOW	89.1	49.5	72.4	32.3	60.8
DFKI-CueFreqPOS	87.0	47.2	56.4	20.1	52.7
DFKI-OWN	89.8	43.4	69.0	41.0	60.7

Table 4.11: F_1 results using DFKI annotation. DFKI-BOW uses bag-of-word features from the original DFKI dataset. DFKI-CueFreqPOS and DFKI-OWN use additional features from a corpus we preprocess that includes DFKI annotation labels.

this perspective.

Within citation analysis we are most interested in the literature related to classification schemes that have been proposed for categorizing citations in scientific literature, of which there are many. Liu (1993) provides a detailed look at these classification schemes from the early ones of Garfield (1964) to those of the early 1990s. Bornmann and Daniel (2008) provide a similar survey that covers more recent classification, including some automatic classification schemes. Those two surveys cover the breadth of possible citation classification schemes. We will discuss a few of them and mention why we ultimately choose the MM scheme.

Garfield's (1964) original scheme introduces 15 different motivations for why an author might cite a paper, which Weinstock (1971) later revisits as he explores the emergence of citation indexes. This classification is appealing because capturing the true motivation of the citation could lead to interesting insights. However, this type of annotation is difficult to obtain from the original authors, and by having independent annotators speculate on the motivation of the original author, the benefits of that annotation may be lost. In fact, many of the studies which followed Weinstock aimed to characterize the *function* of citations as opposed to the motivation. One example is the MM scheme we adopt here.

Chubin and Moitra (1975) attempt to simplify and flatten the MM scheme using six categories: affirmative essential basic, affirmative essential sub-

sidiary, affirmative supplementary additional, affirmative supplementary perfunctory, negational partial, and negational total. Spiegel-Rösing (1977) produces a classification scheme with 13 categories that she uses to evaluate one journal's scholarly contributions. We note that several other studies (Cano, 1989; McCain and Turner, 1989) have also reused or refined MM in some way, which reinforces our choice. As stated earlier in Section 4.2, we feel that the multi-faceted composition of MM provides us with a more flexible annotation scheme and a powerful one that can easily represent the quality of a citation as well as its relation to the citing author.

These early annotation schemes were manually applied to a limited amount of scientific literature and did not consider automatic application on large amounts of text. One early application of automatic citation classification (Nanba and Okumura, 1999) uses an annotation scheme with only three classes (Basis, Compare, Other) that are reportedly based on the 15 classes from Weinstock (1971). Teufel et al. (2006a) introduce a much more complete annotation scheme with 12 classes designed for IR. They thoroughly motivate and analyze their annotation scheme and report inter-annotator agreement of κ=.72. More recently, sentiment analysis has been applied to citations. Athar (2011) classifies citations as *positive*, *negative*, and *objective*, and finds marked improvement in classification using dependency relation features. Athar and Teufel (2012) extend this work and consider context windows of different widths. For each of these three studies the largest class is the one with the least informative label: Nanba and Okumura's *Other* is 52% of citations; Teufel et al.'s *Neutral* is 63%; and Athar's *objective* is 86%. This means that an application receives little information about a majority of citations. In contrast, our annotation scheme does not have a neutral label and always assigns a multi-faceted label that will contain some useful information as no facet can be left undefined.

Dong and Schäfer (2011) conduct a classification study using their own classification scheme with four labels relating to the function of the MM *organic/perfunctory* facet. In addition to adding new syntactic features (`POS-pattern`$_k$, see above), they test ensemble-style self-training to overcome the problem of limited annotated data. Their paper also includes a new dataset with an-

notated citing sentences. It is important to use previously-tested, publicly-available data, however, their dataset does not contain the full corpus from which they extracted features. Due to this restriction we cannot extract many of the features that they use (e.g., features in the LOCATION and FREQUENCY classes). The annotation in their dataset is also attached to the sentence and not individual citations. This makes it impossible to classify individual citations and prevents us from using the citation-specific features that we have developed (OWN features in STRUCTURE class, e.g., `is-constituent`). We believe that annotating and classifying citing sentences (as opposed to citations) is not specific enough for tasks like IR and bibliometrics. Thus, it is essential that we have a citation-annotated corpus for accurate classification.

4.8 Summary

In this chapter, we address the task of citation classification for applications that access and analyze the scientific literature. We propose using MM, a standard classification scheme for citations that was developed independently of automatic classification and therefore is not bound to any particular citation application. We introduce new features designed for citation classification and show that they improve performance as measured by F_1. We go on to show how different classes of features may also affect performance.

Building on the progress we have made with citation classification, we would like to incorporate the labeled citations in downstream tasks like those alluded to earlier: summarization, IR, and bibliometrics. Our proposals and our initial experiments in this direction are described in the next chapter.

Chapter 5

Conclusion

5.1 Summary of Contributions

The objective of this thesis is to improve the data access in intellectual property, and we have made a number of contributions to this end.

Built MT system from multilingual IR collection. We first took a multilingual patent collection that includes parallel translations for claims. We aligned the sentences and then the words in these parallel portions of text. The word alignment output was then used as a dictionary (Chapter 2) or as input for training a machine translation system (Chapter 3). The approach used in Chapter 2 focused only on translating terms, using the word alignment as a bilingual translation dictionary. Chapter 3 takes this a step further by using the word alignments to train an SMT system, effectively resulting in a phrase translation dictionary. These dictionaries were used to expand monolingual queries into multilingual queries by adding translations for multilingual patent prior art search. To the best of our knowledge, this was the first application of an SMT system trained on an IR collection that was used for multilingual IR on the same collection.

Improved recall with multilingual query expansion. By translating and expanding queries for patent retrieval, we were able to improve retrieval

results, in particular recall. Initially, we evaluated two different term translation solutions: (i) using an existing generic bilingual dictionary, and (ii) using the patent-specific dictionary taken from the parallel patent claims (described above). With these experiments we showed that query translation and expansion can improve retrieval, also in conjunction with traditional statistical query expansion (i.e., Rocchio). Next, we expanded our approach to include phrase translations and expand the queries with phrases. Our results varied for individual term and phrase translations for French and German queries. Phrase translation was generally more beneficial for French than for German. For both languages phrase translation was more volatile than term translation, with more queries performing better than the baseline for phrase translation, but also more queries performing worse than the baseline. We saw in both sets of experiments (Chapters 2 and 3), in all languages, that our translation and expansion method showed particularly positive results on hard queries, queries where traditional query expansion often has difficulty.

Classified citation function. We showed that we can improve citation (function) classification by combining lexical, linguistic, and sentiment features (among others). We identified the citations in scientific literature (i.e., ACL proceedings) and annotated them for citation function, as defined in Appendix B.1, using the four-facet classification scheme of Moravcsik and Murugesan (1975). Using this annotated corpus we were able to build a large feature set and study the effects of different classes of features on citation classification accuracy. We found that the lexical features are the most important for this classification in general. However, we also found that the importance of different feature classes varied across facets, which motivates (i) our use of a faceted classification scheme in place of a flat classification, and (ii) our extended feature set where some new task-specific features lead to improvements in accuracy for some facets (see Chapter 4 for details).

Incorporated visual analytics with NLP and IR. This thesis has been carried out as a part of the project *Scalable Visual Analysis of Patent and Scientific Document Collections*, where we have investigated the benefits of

combining IR, NLP, and visual analytics to improve information access. As the primary research focus of this thesis is IR and NLP, we have left more detailed discussion of visualization and visual analytics in an appendix. This thesis does not contribute to the advancement of visualization or visual analytics, but our collaboration (between the NLP and visualization groups) strongly supports the notion that text analytics has something to contribute to visual analytics and vice versa. More details on the output of this collaboration can be found in the following appendix (Appendix A).

5.2 Future Work

There is still progress to be made in simplifying access to the information found in intellectual property collections. We propose here the most promising directions that could be followed to further improve IR and NLP methods for intellectual property, and include possible ways of extending this with visual analytics.

5.2.1 NLP

Along with IR, there are a number of other tasks for which we can apply NLP. Summarization and keyword extraction, for example, have been used for both patents and scientific literature (e.g., Bouayad-Agha et al., 2009; Qazvinian and Radev, 2008). There is also interest in the task of projecting the impact of a patent or scientific article (Yogatama et al., 2011). Taking that a step further, we would like to extend this projection to predict future developments in a patent domain or field of scientific study. We think that by leveraging some tools in NLP this would be possible.

A first step would be to improve the citation classification presented in Chapter 4. We plan to do this with the following two extensions: (i) by using bibliographic information in the references section and also external information about those references (i.e., for a paper listed in the references, check citation statistics from an external resource, e.g., Google Scholar or the ACM Digital Library); (ii) build a second classifier which relies on the

external citation information and then combine the *internal* citation classifier (described in Chapter 4), which only uses information local to the document, with this *external* classifier by means of co-training or stacking.

With improved citation function classification, we can apply citation function labels to a corpus of scientific literature with greater confidence. Various ways of analyzing research trends in scientific literature have already been proposed that rely on citation networks (many have been proposed, one early example is Small and Griffith, 1974). We plan to improve these techniques by augmenting the citation networks to form specific citation function networks.

Other NLP techniques that can be applied that might help in observing research trends are keyword extraction (Gupta and Manning, 2011) and topic modeling (Hall et al., 2008). By using either technique and observing changes over time, we can follow the development of a particular area or domain and potentially project that development into the future.

5.2.2 IR

There are several ways that we could extend our approach to accessing intellectual property with IR. A logical next step to our work in Chapters 2 and 3 is to incorporate the translations in the IR model. This could be done by incorporating the translation probabilities into a probabilistic IR framework, e.g., adding translation probabilities would be a natural extension within the language modeling framework. Much of the work in patent retrieval until now has essentially been on query building and query refinement. Little work has gone into adapting the IR models, which are often developed for ad hoc retrieval, for intellectual property retrieval by leveraging characteristics unique to this domain (e.g., fields like *abstract* or *IPC*).

We can also use the output of our citation classification for improving intellectual property retrieval. For example, we could apply the topic-sensitive PageRank algorithm (Haveliwala, 2003) to the citation network using citation function labels as topics. These PageRank scores could then be applied as weights in the retrieval system. Such an approach might make it easier to find an article explaining the details of maximum entropy classification

instead of a paper presenting a new maximum entropy part-of-speech tagger or vice versa. Other topics or extracted keywords could of course also be used in place of citation function.

5.2.3 Visual Analytics

There are a number of avenues to pursue for improving access and analysis of intellectual property using visualization and visual analytics. Here we will concentrate on only a few extensions to our current work that involve NLP and IR.

In Appendix A we present FeatureForge, a tool to visually aid classification and clustering, which has been built primarily with text classification in mind. We would like to extend this tool to handle a number of NLP classification and clustering tasks however. Feature engineering for NLP can be a difficult process and our goal is to incorporate NLP tools into the visual feature engineering tool to ease feature definition and determine, in one analysis loop, if the feature is useful for machine learning or not.

Another visual classification tool from Heimerl et al. (2012a) provides a way to perform visual active learning with the advantage of observing how the annotation choice(s) affect the classification on a 2-D representation of the classifier (i.e., this relies on linear classifiers). We would like to extend this to handle regression problems and apply it to our retrieval results in a learning-to-rank scenario. Currently the tool focuses on classification instances close to the decision boundary. Instead we would like to focus on improving precision and therefore focus on the instances that already match a given classification or query, but where we want the most relevant documents to have the highest classification score.

Appendix A

Visual Analytics for IR and NLP

A.1 Introduction

The work presented in this thesis was done within a larger project that aimed to combine NLP, IR, and visual analytic techniques to better access the information found in large, technical text collections – specifically intellectual property collections of patents and scientific literature. In this appendix we want to introduce some of the visual techniques which were used alongside the IR and NLP approaches presented in the thesis, even though these techniques are beyond the research scope of this thesis. In Section A.2, we present some of the visual analytic techniques related to patent retrieval (these relate to Chapters 2 and 3). In Section A.3, we cover visual tools used to enhance the citation classification that was described in Chapter 4.

A.1.1 Related Work

A significant amount of the work in the visualization and visual analytics community has dealt with text analysis (Alencar et al., 2012), with much of that concentrating on visualizing collections of documents. The interaction, in the broadest sense, of visual and textual tools varies widely: from more static visualization of specific linguistic phenomena (Mayer et al., 2011) to

the integration of visual and textual analytic tools aimed at a particular task (Heimerl et al., 2012b). Our interest here lies more in *visual analytics* (Thomas and Cook, 2005), which we might distinguish from static visualization in that the visual tools are designed as part of an interactive process integrating the user in the analytic loop. Melding or interleaving visual and textual analysis seems to us to be among the most promising avenues for making sense of very large collections of text data.

To bridge the gap between NLP and visualization/visual analytics researchers, there have been a few visualization tutorials for the NLP community to promote synergy in the two fields (Collins et al., 2008; Penn et al., 2009); and at VisWeek 2012, Oelke et al. (2012) presented some standard NLP tools and their applications to encourage their use in visual analytics.

We will discuss here just a couple of the more prominent examples involving text analytics and visual analytics. More comprehensive surveys of this literature can be found in (Penn et al., 2009)[1] or (Alencar et al., 2012). Jigsaw (Stasko et al., 2007) is an example of a successful visual analytics tool that relies on NLP techniques. It uses named entity recognition to identify and link named entities, such as persons, organizations, or locations, throughout a document collection. Jigsaw then uses a variety of different views to aid in exploring the information in the document collection. The "List" view, for example, provides a useful and intuitive way to investigate how named entities interact.

The Action Science Explorer system (Dunne et al., 2012) is another visual analytics tool using NLP techniques for text processing and analysis. We also mention it here because of its relevance to our citation classification work in Chapter 4. The tool has different views and approaches for navigating scientific literature. Of particular interest to us are the various summarization techniques applied (Qazvinian and Radev, 2008; Qazvinian et al., 2010) to the scientific literature and how they are displayed. It would be interesting to apply the citation functions from our classification within a tool like this, observing how the summarization or citation network graph change using this additional information.

[1] `http://esslli2009.labri.fr/documents/carpendale_penn.pdf`

Surveying the 'NLP+IR+visual analytics' landscape, much of the work involves *shallow* NLP applications. This is logical as these solutions already provide much of the information desired for analysis. However, applying visual analysis to deeper NLP techniques, like syntactic and semantic parsing, would be interesting as well, as both rely on combining automatic and manual techniques for achieving a more thorough understanding of the data.

A.2 Visually Interactive Patent IR

Our goal is to combine the NLP methods from Chapters 2 and 3 with visual analytics methods in a novel way to improve IR. Some approaches to this were explored within the PatExpert project (Wanner et al., 2006; Giereth et al., 2007; Koch et al., 2011). Our focus is instead on the multilinguality of patents. This is of particular interest to our work for two reasons:

(i) The amount of translated text available for retrieval is increasing, and so is the number of collections that contain the same documents in multiple languages, such as patent collections or Wikipedia. These may be parallel corpora or comparable corpora. For example, Wikipedia constitutes a comparable corpus with documents in different languages that are not exact translations of each other, but contain significant overlap in content.

(ii) Today's typical users of IR systems, and more specifically patent retrieval systems, are very likely to be multilingual. However, their level of competence in different languages usually varies considerably. For example, they may speak one or two of the languages perfectly, while in another they have good passive knowledge but limited active competence.

In Chapter 2, we address this multilinguality scenario for patent retrieval by computing a statistical word alignment on the retrieval collection resulting in a set of bilingual translation dictionaries. We then translate patent queries on the assumption that many patent professionals are not capable of, or comfortable with, manually translating patent queries themselves, even though

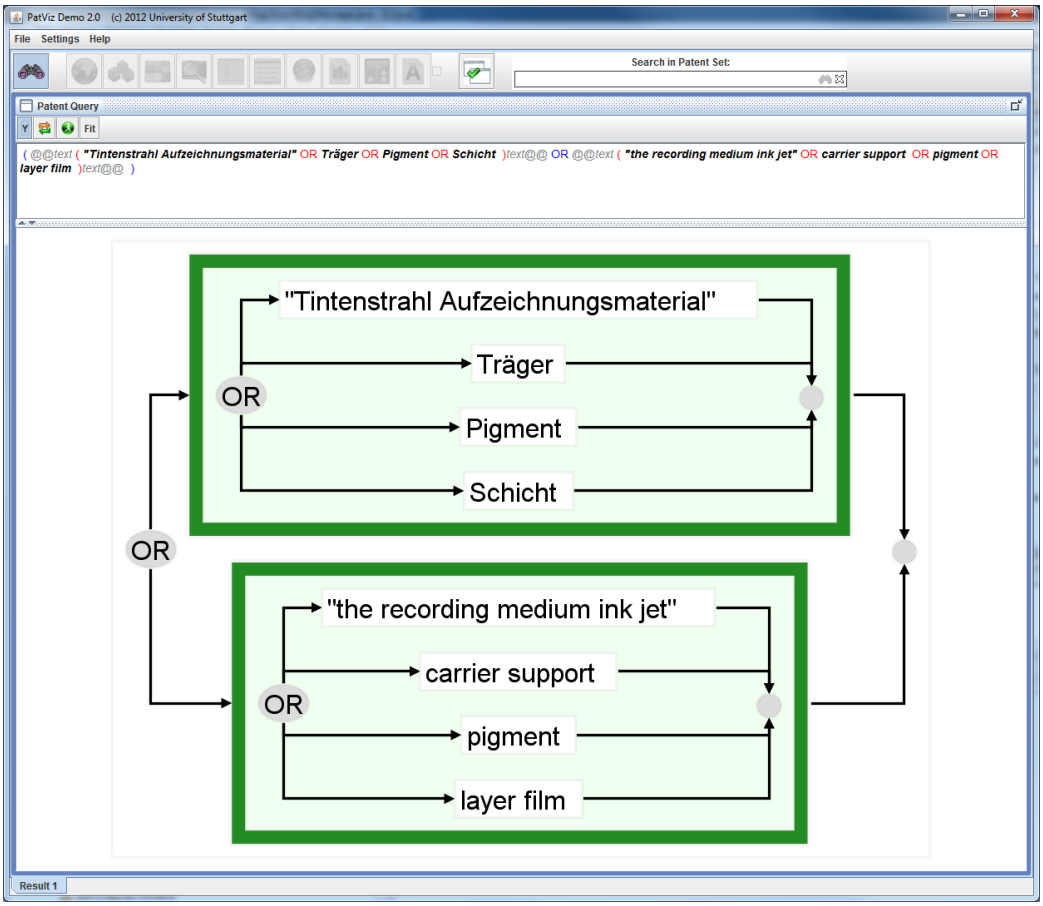

Figure A.1: Visual query building tool (Koch et al., 2011).

some others who speak perfectly all languages involved might prefer doing it themselves. Thus, we use an NLP method (statistical word alignment) on a multilingual patent collection (hence exploiting point (i) above) to help patent professionals that are partially, but not completely multilingual (e.g., they can read French, but cannot translate into French). The latter functionality addresses point (ii) above.

A.2.1 Visually Building Multilingual Queries

One research goal is to test interactive visual interfaces that let users select a subset of the translations available for a query, ideally leveraging the information available from the statistical word alignment. We believe that with the complexity introduced by using multiple languages, careful consideration needs to be made for optimizing multilingual search interfaces. Earlier studies have already shown that patent retrieval can benefit from interactive user interfaces (Larkey, 1999), and we want to extend this finding to the case of multilingual patent retrieval. One solution for doing this is by building multilingual queries graphically using an interactive tool. Alink et al. (2009) used a graphical query builder for monolingual queries and Koch et al. (2011) promoted iterative query refinement with their graphical query tool. It is this latter iterative tool that we can use for building multilingual queries. Automatic query translation and query expansion are important prerequisites to help users quickly define queries covering multilingual patent documents, but an interactive approach provides a higher level of control to patent specialists, who continually manually fine-tune each query.

We can use the translation dictionaries (with translation probabilities) from Chapters 2 and 3 to automatically suggest translations for all query terms or all query terms with translation probability over a user-defined threshold. This will then be used to build the graphical query like the one seen in Figure A.1. The patent specialist can then run the fully translated query, or, seeing that the translation *the* for 'Tintenstrahl-Aufzeichnungsmaterial'[2] is not correct in this context, remove the term(s) and run the revised query. This solution should help patent searchers with different levels of language proficiency. It also preserves the Boolean structure of the query, which is still commonly used for patent queries (Azzopardi et al., 2010), and shows the

[2]This translation is an artifact of the word alignment where "Tintenstrahl-Aufzeichnungsmaterial" is most often aligned to the words "the recording medium ink jet"; consequently, the probability mass of translating to "Tintenstrahl-Aufzeichnungsmaterial" is divided quite evenly between those five English words, i.e., $p(tintenstrahl - aufzeichnungsmaterial|the) = p(tintenstrahl - aufzeichnungsmaterial|recording) = p(tintenstrahl - aufzeichnungsmaterial|material)$, etc.

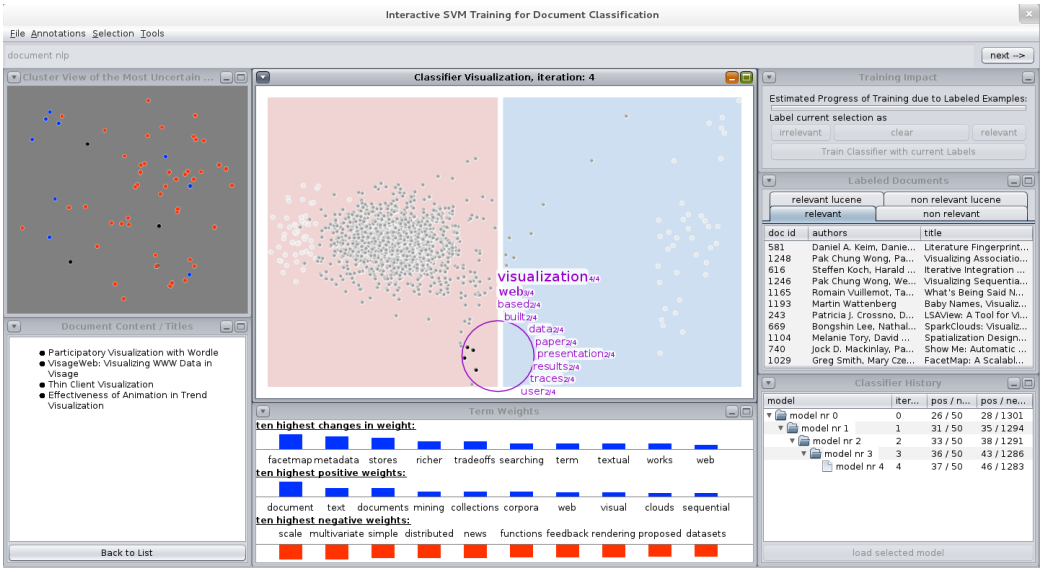

Figure A.2: Interactive SVM training for document classification from Heimerl et al. (2012b).

relation between translated terms.

A.3 Visual Analytics for Document Classification

Before describing our work in visual analytics for document classification, we will first introduce a tool developed by Heimerl et al. (2012b) for training a document classifier (see Figure A.2). This tool was built primarily around a two-dimensional classifier view in which the user could clearly observe the state of the classifier at different stages of training. The tool is inspired by *active learning*, which is a technique used in machine learning for economically increasing the number of labeled instances for training before reaching a certain threshold. This has traditionally been done by the learner "querying" the human annotator for the correct label, one instance at a time. The visual active learning tool makes it easy to explore the instances to be annotated

and also allows the user to identify similar instances that should be similarly annotated, thus easing the annotation effort. The visual active learning tool of Heimerl et al. was applied to three different corpora for text classification: 20 Newsgroups[3], Reuters RCV1 (Lewis et al., 2004), and a corpus of abstracts from VisWeek[4] publications. Traditional active learning is quite a high baseline to beat, but their results are promising: users of the tool preferred the additional visual feedback and results (measured by F_1) using visual approaches were competitive with those using more traditional active learning. The potential for improvement with the visual approaches is also greater as users become more accustomed to the capabilities of the visual interface.

The visual active learning tool illustrates the potential for adding visual analysis to an NLP technique like active learning. In the case of active learning, the goal is to improve classification by increasing the number of labeled instances. An alternative approach to improve classification is by improving the classifier's feature representation. To that end, in subsequent work, we developed a visual feature engineering tool, FeatureForge (Heimerl et al., 2012a), for improving the results of text classification (seen in Figure A.3). Specifically, we applied the tool to the citation classification task discussed in Chapter 4. The goal of the tool is to quickly and easily explore the data instances to be classified and the feature space which was defined on them. This was done by integrating a supervised classification approach (e.g,. MaxEnt or SVMs) with an unsupervised clustering approach (e.g., hierarchical clustering using Ward's linkage). Classification and clustering are both widely used in NLP (Manning and Schütze, 1999), but usually they are applied in different circumstances, depending on the availability of labeled data. With a visual analytics tool like FeatureForge we can investigate the relation between how the data instances are "naturally" grouped (clustering) and how those instances have been labeled (classification). By observing how heterogeneous some clusters are (in other words, by finding the clusters which

[3] `http://qwone.com/~jason/20Newsgroups/`

[4] IEEE VisWeek is an annual event comprised of conferences on scientific visualization (SciVis), information visualization (InfoVis), and visual analytics (VAST) (`http://ieeevis.org/`).

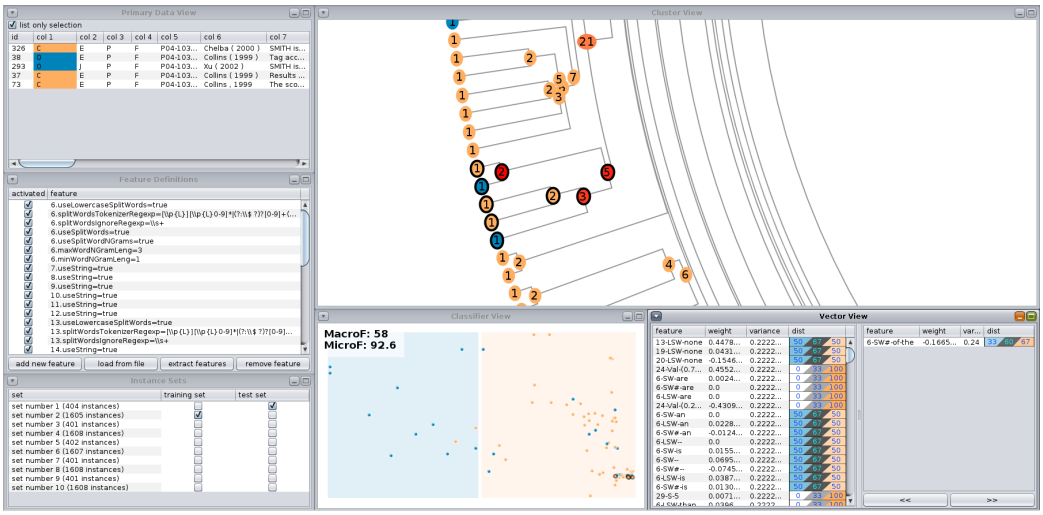

Figure A.3: FeatureForge visual feature engineering tool of (Heimerl et al., 2012a).

were not dominated by a single label), we can determine either (i) where the classification is having trouble due to an underdefined feature set (i.e., cases where an additional feature would be able to distinguish the instances in the cluster), or (ii) where the incorrect label has been assigned due either to annotation error or underdefined annotation guidelines. In either case, it highlights useful information which will lead to better classification in the former case and a better gold-standard data set in the latter.

The tool is also useful for selecting individual data instances or groups of instances and reporting on the importance of each of the features in making that classification decision. Knowing which features positively and negatively affect the classification helps the user make an informed decision on how to edit the feature set within the feature engineering tool. New features can be defined and added that could potentially improve classification, likewise features that hurt classification can be deselected for successive trials. A complete description of the capabilities of FeatureForge can be found in (Heimerl et al., 2012a).

Our evaluation of the tool consisted of a preliminary case study where

we explored the citations in our citation corpus (Section 4.3). In the case study, we identified some ambiguities in the annotation guidelines and also found clustered instances that would likely be separated with some specific additional features, e.g., features that consider a narrower window of context to distinguish different citation in the same sentence.

We plan to extend the FeatureForge tool by integrating additional NLP tools. This would enhance the existing analytic loop where the analyst (i) initially classifies and clusters the data instances, (ii) identifies possible features to discriminate between instances from different classes that have similar feature vectors, (iii) define the new features within the FeatureForge system, and (iv) rerun the classification and clustering algorithms. By allowing for the definition of increasingly informative features in step (iii) – e.g., defining features on named entities after integrating a Named Entity Recognition system – we should be able to improve classification.

A.4 Summary

The research areas of IR and NLP are only just now maturing, and their combination with visual analytics approaches is an even more recent phenomenon with the rapid expansion of the visual analytic field in the last decade. As we have shown in this appendix, despite being in its infancy, the research which combines these three fields is quite promising.[5] With more tutorials like those of Penn et al. (2009) and Oelke et al. (2012) and more collaborations like ours at the University of Stuttgart, the NLP community should be able to better harness the potential of visual analytics, going beyond static visualizations (e.g., parse trees). Furthermore, the NLP community needs to be sure to disseminate effective text analysis techniques so that visual analysts can go beyond bag-of-word vector space models and topic modeling to have a better understanding of the linguistic data.

[5]For some specific examples of our future plans to combine IR, NLP, and visual analytics, see Section 5.2.

Appendix B

Citation Classification Resources

B.1 Annotation Guidelines

B.1.1 Introduction

These guidelines outline an annotation scheme for citations in scientific literature. Citations are used in the text of scientific literature to refer to other sources (most often they refer to other published literature). For example, in the sentence below, "(Oviatt 1996)" is a citation that points to another paper.

(B.1) Multimodal systems provide a natural and effective way for users to interact with computers through multiple modalities such as speech, gesture, and gaze **(Oviatt 1996)**.

For this annotation scheme we would like to consider two aspects of the citation: (1) what is the author saying about the quality of the cited work, and (2) what is the relationship of the citing work to the cited work, i.e., how is the author using the cited work.

With this in mind, we would like to annotate each citation along four dimensions or facets (taken from the classification scheme by Moravcsik and Murugesan (1975)):

- conceptual vs. operational
  - *Is this an idea or a tool?*
- evolutionary vs. juxtapositional
  - *Is the author building on the cited work or working in contrast to it?*
- organic vs. perfunctory
  - *Is this **particular** citation necessary for understanding the paper or can the paper still be understood without it?*
- confirmative vs. negational
  - *Is the cited work correct or are there some limitations to it?*

The first two dimensions correspond to the *utility* of the cited work and the last two dimensions relate to the *quality* of the cited work.

For our annotation, all citations should be completely defined, i.e., no facets left undefined. Finally, a note on terminology, **author** will be used in the guidelines to describe either the citing paper or the authors of the citing paper, i.e. that paper that makes reference to another paper. We will then use **cited work** for either the cited paper or authors of the cited papers.

B.1.2 Conceptual vs. Operational

Generally, if the citation refers to the use of some tool or resource it should be labeled `operational`, otherwise if it is an idea or algorithm it should be labeled `conceptual`. Some examples of tools and resources in NLP might include: taggers, parsers, stemmers, classifiers, or corpora.

Note that there are often cases when a paper has both a conceptual and operational component. Be careful to annotate this accurately for each citation in the case where the work is cited more than once.

B.1.2.1 Operational

Label the citation `operational` if:

- the citation refers to the use of a tool (e.g. tagger, parser, stemmer, etc.), a corpus, etc.

  (B.2) However, most of the existing models have been developed for English and trained on the Penn Treebank **(Marcus et al., 1993)**

B.1.2.2 Conceptual

Otherwise, for example if the citation refers to an idea or algorithm, label `conceptual`. Some specific examples might be citations to theories, algorithms, or any abstract concept found in the cited work.

(B.3) Context is typically treated as a set of unordered words, although in some cases syntactic information is taken into account **(Lin, 1998; Grefenstette, 1994; Lee, 1999)**.

Also in the case that the author refers to implementing the cited work, use the label `conceptual`.

B.1.2.3 Possibly tricky examples?

(B.4) More specifically, we combine a probabilistic topological field parser for German (Becker and Frank, 2002) with the HPSG parser of **(Callmeier, 2000)**. **<OPER >**

(B.5) Various parsing techniques have been developed for lexicalized grammars such as Lexicalized Tree Adjoining Grammar (LTAG) (Schabes et al., 1988), and Head-Driven Phrase Structure Grammar (HPSG) **(Pollard and Sag, 1994)**. **<CONCEPT >**

(B.6) In the following decade, great success in terms of parse disambiguation and even language modeling was achieved by various

lexicalized PCFG models **(Magerman, 1995; Charniak, 1997; Collins, 1999; Charniak, 2000; Charniak, 2001)**.
<CONCEPT >

(B.7) Table 2 compares the results of our algorithm with the results in **(Och and Ney, 2000)**, where an HMM model is used to bootstrap IBM Model 4.**<OPER >**

In (B.7), note that the citation to "results" refers to results from a tool and are therefore labeled `operational`.

B.1.3 Evolutionary vs. Juxtapositional

We define *evolutionary* to be any citation that is compatible with what is being claimed by the author, and *juxtapositional* is any citation that contradicts or contrasts the claims of the author.

Again, a cited work may be cited in one context as evolutionary and in another context as juxtapositional. For example, in a discussion of using machine learning for predicting pitch accent, one citation context may describe the common problem of predicting pitch accent, which is labeled `evolutionary`, and a second citation may describe the different machine learning approach used in the cited work, which is then labeled `juxtapositional`.

B.1.3.1 Juxtapositional

- If the author proposes an alternative to the cited work, label `juxtapositional`.

  (B.8) Our approach differs from **Lin (1998)** in three important ways: (a) by introducing dependency paths...

- If there is any contrastive or juxtapositional element in the citation then label it `juxtapositional`.

  (B.9) **Alshawi et al. (2000)** also presented a two-level arranged word ordering and chunk ordering by a hierarchically organized collection of finite state transducers. The main difference from

our work is that their approach is basically deterministic, while the chunk-based translation model is non-deterministic. The former method, of course, performs more efficient decoding but requires stronger heuristics to generate a set of transducers. Although the latter approach demands a large amount of decoding time and hypothesis space, it can operate on a very broad-coverage corpus with appropriate translation modeling.

B.1.3.2 Evolutionary

If the cited work is the basis of the author's work, is used in the author's work, or even if the cited work is compatible with what is being claimed by the author, we define the citation as `evolutionary`. Below are listed some typical instances where the citation should be labeled `evolutionary`. This is not, however, an exhaustive list, i.e. `evolutionary` instances are not limited to the conditions listed below.

- If the citation is used or even compatible with citing work, mark as `evolutionary`.

  (B.10) we follow **Ennis and Bi (1998)** and use the identities

- If the citation refers to an agreed upon definition, term, or metric, label as `evolutionary`. For example, in example (B.11), although the BLEU score is not being extended, just by using it we assume it is an endorsement of the metric.

  (B.11) We utilize BLEU **(Papineni et al., 2002)** for the automatic evaluation of MT quality in this paper.

- If the citation discusses a shared problem, label as `evolutionary`. This context should be labeled `evolutionary`, even if later in the paper there is a separate citation that discusses how the author distinguishes itself from the cited work.

  (B.12) Information Extraction (IE) is the process of identifying events or actions of interest and their participating entities from

a text. As the field of IE has developed, the focus of study has moved towards automatic knowledge acquisition for information extraction, including domain-specific lexicons **(Riloff, 1993; Riloff and Jones, 1999)** and extraction patterns **(Riloff, 1996; Yangarber et al., 2000; Sudo et al., 2001)**.

- If it is a tool (i.e., labeled `operational`), then label `evolutionary` if the tool is simply being used (i.e., when the author uses a particular tagger) or if a series of third-party tools are being compared (i.e., a review of several different taggers). However, label as `juxtapositional` if the cited tool is being directly compared to the author's tool being presented.

B.1.4 Organic vs. Perfunctory

Generally, *organic* citations will be those that are very important for understanding the author's work. These can be citations that form the basis of the author's work, or any citations without which the paper would not make sense, or citations to otherwise unique work that cannot be referred to with any other citation. *Perfunctory* citations on the other hand are citations used to point to related literature, work, or authors that are not necessarily essential to understanding the author's paper.

B.1.4.1 Perfunctory

Label `perfunctory` if:

- the citation could easily be replaced by another citation (or removed altogether) and the general point could still be understood and make sense.

  (B.13) Vector spaces enjoy widespread use in information retrieval **(Salton and McGill, 1983; Baeza-Yates and Ribiero-Neto, 1999)**...

- the citation is in a list of citations (i.e., the citation could be replaced or omitted). This is the case for explicit lists like example (B.14) or implicit lists like example (B.15). One exception to this rule may be if all of the cited work in the list has at least one common author, so that it could be considered a unique work that spans several papers. In this case the citations may be labeled `organic`.

  (B.14) Corpus-based methods and machine learning techniques have been applied to anaphora resolution in written text with considerable success **(Soon et al., 2001; Ng & Cardie, 2002**, among others).

  (B.15) The utilization of language technology for the creation of hyperlinks has a long history (e.g., **Allen et al., 1993**).

- the citation comes at the end of a sentence without being specifically referred to in the text and with no further explanation. The assumption being that that citation justifies the statement.

  (B.16) Even narrow-coverage context-free natural language grammars produce explosive ambiguity **(Church and Patil, 1982)**.

- the citation refers to details in another paper (usually by the author)

  (B.17) See **Baldwin and Bond (2003)** for further details.

- the citation is to a tool (i.e. a citation to a tagger or parser that could be replaced by using another tagger or parser).

B.1.4.2 Organic

If the citation refers to important work or work that is uniquely necessary for making sense of the author's work then it should be labeled `organic`. Examples of this may be when the author's work is based on or inspired the cited work, when the cited work is fundamental in realization of the author's work, or when the author cites something very specific that can only be found in that particular cited work.

Note it should also be labeled `organic` if:

- there is a list of citations with the same author (or one common author). (This is an exception to one of the *perfunctory* rules above.)

Typical `organic` example:

(B.18) In order to exploit syntactic dependencies in a larger context, we propose a new model of supertagging based on Sparse Network of Winnow (SNoW) **(Roth, 1998)**.

B.1.5 Confirmative vs. Negational

This facet is similar to the NLP task of *sentiment analysis*, which is basically determining if the author is describing something as positive or negative. However, sentiment is manifested differently in published scientific literature with respect to product reviews for example. We should reconsider what constitutes positive and negative language in scientific literature and keep in mind that negative citations have been shown to be rare in published articles (Moravcsik and Murugesan, 1975).

Following the labels used by Moravcsik and Murugesan, we use `negational` to refer to negative citations and `confirmative` to refer to positive citations.

We will consider `negational` citations to be those where the author is critical of the cited work, highlights shortcomings or limitations of the cited work (and likely proposes solutions to it), or generally disagrees with the assertions in the cited work.

We will consider `confirmative` citations to be those where the author supports the cited work, highlights particular positive aspects of the cited approach, or generally agrees with or at least accepts the assertions in the cited work. We will also consider citations that do not seem to be positive or negative to be `confirmative`. This is because simply by citing the work we assume that the author agrees with it or thinks positively of it.

Note that different citations to the same paper can be assigned different labels. For example, a citation might be introduced and praised for its

initial contribution and later criticized for its shortcomings. If this is nicely separated by having two citations, label each of them accordingly. If for one citation there is mixed positive and negative feedback, the annotator should label the citation as `negational`.

B.1.5.1 Negational

Label `negational` if:

- the citation is explicitly negative: illustrating major faults in the cited work's methodology, results, or conclusions, etc.
- the author points out limitations in the cited work (and proposes alternative solutions). If these critical comments follow statements of praise, then a decision must be made by the annotator on the positivity/negativity of the citation. However, the annotator should lean towards `negational`.

  (B.19) Various supervised learning methods for Named Entity (NE) tasks were successfully applied and have shown reasonably satisfiable performance.**((Zhou and Su, 2002)(Borthwick et al., 1998)(Sassano and Utsuro, 2000))** However, most of these systems heavily rely on a tagged corpus for training. ...

- If the citation is marked `juxtapositional`, take care in labeling `confirmative/negational`. Mark `negational` if the citing work fills a void or corrects something in the cited work. If the cited work is different and distinct enough (still `juxtapositional`) the citation might not necessitate a negative value and therefore be marked `confirmative`, i.e. if we are "comparing apples to oranges."

  For example, in (B.20), the author's work and the cited work differ, but the objectives of each are distinct. The author does not necessarily have any negative comments.

(B.20) This differs from the BioCreAtIvE competition tasks that aimed at classifying entities (gene products) into classes based on Gene Ontology **(Ashburner et al., 2000)**.

Typical `negational` example:

(B.21) Unlike well-known bootstrapping approaches **(Yarowsky, 1995)**, EM and CE have the possible advantage of maintaining posteriors over hidden labels (or structure) throughout learning;

B.1.5.2 Confirmative

Remaining citations may be labeled `confirmative`.

Some more specific cases:

- if the citation refers to the use of a tool, label `confirmative`, even if there is no explicit value judgment (it is assumed that any use of the tool at all is positive).
- if the author uses the cited algorithm, technique, etc., without alteration.

Typical `confirmative` example:

(B.22) A later study **(Pang and Lee, 2004)** found that performance increased to 87.2% when considering only those portions of the text deemed to be subjective.

B.2 Automatically Extracted Cue Phrases

Table B.1: Cue phrases automatically extracted using mutual information (MI) (Manning et al., 2008). Cue phrases were extracted for each fold in 5-fold cross-validation. Subscript indicates the test fold the cue phrase features were applied to, i.e., cue phrase '1' was extracted from folds 2-5.

Label	Cue phrase
CONC	agreement^{0}, algorithm^{01234}, algorithms^{34}, alternative^{23}, analysis^{01234}, anchor^{3}, annotation^{2}, application^{3}, approach^{01234}, approaches^{01234}, assumption^{2}, automatic^{01234}, based^{1}, bds^{03}, between^{01234}, bilingual^{24}, bleu^{3}, candidate^{01234}, centering^{03}, class^{4}, coherence^{0123}, compute^{023}, computed^{1}, considered^{034}, defined^{4}, dependency^{2}, descriptions^{0234}, details^{2}, different^{24}, discriminative^{0123}, discussed^{01234}, distribution^{024}, documents^{12}, e.g.^{01234}, em^{14}, entities^{2}, entropy^{034}, error^{0123}, examples^{0124}, feature^{4}, finding^{0234}, first^{1}, following^{01}, follows^{0124}, form^{0124}, function^{1234}, generative^{12}, good^{034}, graph^{012}, head^{024}, i.e.^{0}, important^{01234}, information^{1}, instances^{01}, instead^{0123}, interpretation^{0}, introduction^{1}, kernels^{4}, labels^{04}, language^{01234}, language model^{4}, large^{2}, ldd^{03}, lexical^{034}, linguistic^{0}, literature^{03}, local^{3}, main^{3}, many^{0134}, maximum^{0134}, method^{01234}, methods^{024}, metrics^{1}, model^{01234}, models^{34}, more^{01234}, n-gram^{01234}, natural^{1}, natural language^{01234}, nlp^{12}, node^{024}, np^{0124}, number^{01234}, one^{01234}, pairs^{1}, paper^{1}, parameter^{13}, parameters^{4}, parse^{1234}, parsing^{234}, phrase^{1}, probabilities^{2}, probability^{134}, problem^{01234}, proposed^{24}, ranking^{01234}, recent^{01234}, recently^{23}, related^{3}, related work^{01234}, representation^{01234}, resolution^{034}, respectively^{34}, result^{01234}, same^{23}, score^{1234}, see^{1}, sense^{1}, sequence^{014}, sets^{3}, show^{12}, shown^{01234}, similarity^{1}, simple^{4}, size^{12}, smith^{01234}, smith used^{23}, structure^{01234}, studies^{024}, successfully^{1}, such^{012}, suggested^{023}, surface^{0}, tasks^{01234}, technique^{1}, temporal^{0}, texts^{2}, theory^{0134}, those^{0}, three^{01234}, time^{3}, translation^{134}, translations^{1}, trees^{2}, two^{04}, useful^{0234}, values^{13}, way^{4}, web^{0}, within^{34}, work^{01234}

Continued on next page

Table B.1: (continued)

Label	Cue phrase
OP	abstracts^{234}, acquired^{01234}, add^{01}, against^{013}, along^{3}, annotated^{2}, answer^{0123}, at&t^{3}, automatically^{01234}, automatically acquired^{03}, available^{01234}, best^{03}, bllip^{234}, bllip corpus^{234}, boundaries^{0134}, built^{012}, category^{14}, clustering^{3}, collection^{1234}, comlex^{0123}, comparison^{0}, containing^{234}, corpus^{01234}, correct^{014}, currently^{34}, data^{0134}, database^{4}, dcu^{13}, developed^{2}, development^{123}, domain^{234}, empty^{12}, english^{0124}, erg^{0124}, evaluate^{01234}, evaluation^{014}, experiments^{0123}, expressions^{2}, extracted^{1234}, forms^{013}, framenet^{0123}, fsm^{013}, genia^{24}, german^{01234}, gnome^{0234}, gnome corpus^{0234}, grammar^{02}, identify^{3}, implementation^{0123}, implemented^{23}, itspoke^{0124}, kernel^{013}, kiosk^{0234}, lattice^{0}, library^{01234}, lingo^{0124}, million^{134}, million words^{4}, mobile^{024}, modified^{04}, module^{13}, multimodal^{0234}, name(both)^{0134}, names^{124}, negra^{0124}, nei^{0124}, news^{1234}, news stories^{234}, nodes^{1}, parallel corpus^{1234}, parsed^{34}, parsed using^{34}, parser^{01234}, parsers^{2}, part^{01234}, pcfg^{0}, penn^{0123}, penn treebank^{01}, perform^{01234}, performed^{0}, proper^{0134}, provided^{12}, provides^{0234}, purposes^{014}, question^{4}, resource^{01234}, rwth^{013}, scf^{0124}, scf types^{0124}, scfs^{1}, section^{3}, semantic^{3}, software^{013}, speech^{034}, standard^{14}, stories^{234}, street^{124}, system^{01234}, tag^{1234}, tagged^{01234}, tagger^{01234}, tagging^{1}, tags^{12}, taken^{0234}, template^{034}, temporal^{1}, through^{023}, together^{0234}, tool^{0123}, toolkit^{01234}, top^{01234}, trec^{0}, treebank^{0123}, trigram^{0}, types^{0124}, used^{24}, users^{034}, using^{0124}, wall^{124}, wall street^{124}, wide-coverage^{3}, wordnet^{01234}, wsj^{0124}, xtag^{0124}

Continued on next page

Table B.1: (continued)

Label	Cue phrase
EVOL	's^{4}, according^{01234}, acquired^{024}, active^{1}, agreement^{1}, algorithm^{01234}, already^{2}, although^{3}, annotated^{0}, application^{3}, automatic^{0}, automatically^{0}, baseline^{24}, candidate^{3}, case^{3}, centering^{0123}, classifier^{0}, clustering^{0124}, coherence^{23}, common^{13}, compute^{23}, computed^{1}, context^{4}, corpus^{01234}, cost^{0234}, current^{0}, data^{1234}, defined^{0124}, described^{03}, details^{01234}, developed^{4}, dialogue^{014}, discourse^{0234}, discussed^{1}, domain^{3}, each^{01234}, em^{14}, entropy^{34}, error^{3}, experiments^{3}, extracted^{123}, features^{014}, figure^{3}, first^{234}, following^{01234}, follows^{2}, frequency^{34}, function^{1234}, functions^{234}, general^{0}, german^{0134}, given^{03}, human^{0234}, implementation^{2}, important^{3}, information^{01234}, input^{0}, instances^{14}, introduced^{24}, introduction^{01234}, kernel^{012}, kernels^{0124}, knowledge^{0}, learning^{01234}, less^{124}, lexical^{0124}, lexicon^{0}, list^{13}, local^{2}, machine^{01234}, machine learning^{4}, machine translation^{01234}, maximum^{0134}, measure^{2}, measures^{1}, method^{0}, metrics^{1234}, multimodal^{034}, n-gram^{1}, new^{124}, np^{0124}, number^{01234}, order^{12}, original^{4}, output^{3}, paper^{0124}, parser^{0}, penn^{3}, penn treebank^{3}, phrase^{0}, pos^{0234}, process^{134}, proposed^{0}, provided^{4}, question^{0123}, ranking^{1234}, recall^{13}, recent^{0134}, reference^{3}, references^{3}, relations^{1}, research^{01234}, resolution^{2}, respectively^{4}, retrieval^{124}, scf^{0124}, scfs^{0124}, score^{2}, second^{0}, section^{2}, see^{3}, semantic^{1}, sense^{01234}, senses^{0234}, sentences^{01234}, sequence^{13}, shown^{0124}, simple^{01234}, smith used^{2}, speech^{24}, strategy^{123}, string^{012}, structures^{3}, suggested^{0}, surface^{1}, system^{1}, systems^{01234}, target^{0134}, tasks^{0}, technique^{14}, temporal^{2}, test^{01234}, text^{01234}, theory^{3}, three^{2}, time^{2}, training^{4}, translation^{0}, translations^{0}, two^{0123}, type^{3}, use^{1}, used^{01234}, useful^{3}, using^{01234}, value^{2}, verb^{04}, verbs^{014}, version^{0124}, way^{0}, weighted^{01234}, wordnet^{01234}, words^{01234}

Continued on next page

Table B.1: (continued)

Label	Cue phrase
JUX	achieve[2], achieves[03], actual[024], addressed[12], against[0134], algorithms[0123], alignment[01234], alignments[02], allows[01], antecedents[0124], approach[01234], approaches[134], approximately[134], association[0234], based[3], behind[0124], below[3], best[012], bilingual[3], bilingual corpus[24], boosting[2], broad[0], called[3], capacity[0124], cfg[0134], chinese[134], classification[3], combine[2], compared[234], comparison[0123], computation[34], concepts[04], constituents[1], constraint[0123], constraints[4], contexts[2], dependency[0124], dependency structure[0], descriptions[0124], dictionaries[24], dienes[0124], different[01234], discriminative[0123], distinction[012], dynamic[1234], e.g. smith[02], efficiency[03], empty[03], entire[012], entities[01234], evaluated[124], extract[0234], f-structure[0134], finding[012], fine-grained[4], fixed[01234], formal[04], fsm[0134], generative[0123], generative capacity[0124], gesture[0134], hand-crafted[0134], handle[0134], identification[123], improved[0124], improvement[134], interested[12], involving[12], kind[34], language pairs[13], large[4], latter[02], ldds[0134], level[134], lfg[4], limited[01], links[0234], literature[12], log-linear[134], looking[23], make use[3], medline[0234], model[12], models[01234], much[0234], node[0234], none[04], noun phrase[1], observed[0234], pairs[01234], parallel corpora[2], parser[13], part-of-speech tagging[13], path[034], performance[023], polarity[2], previous[012], productions[034], programming[1234], rates[4], reasons[12], recorded[012], relationship[234], relationships[0234], relationships between[034], reported[0134], reranking[23], researchers[0234], results[01], rwth[013], showing[01], small[014], smith[01234], smith proposed[34], smith use[234], solver[013], statistical[23], strong[124], structural[2], studies[13], subset[3], successfully[2], template[034], top[134], towards[124], translation models[2], unlike[0134], unsupervised[0124], upon[0124], van[0124], versions[134], word alignment[0234], word lattice[1234], work[1]

Continued on next page

Table B.1: (continued)

Label	Cue phrase
ORG	adaptation^{24}, adapted^{12}, algorithm^{01234}, along^{4}, application^{1}, associated^{3}, automatic^{0}, basic^{0124}, bleu^{1234}, candidate^{2}, categories^{4}, category^{0124}, class^{2}, cluster^{0134}, components^{0134}, computed^{234}, computer^{123}, computing^{1234}, consisted^{0123}, constant^{123}, data^{0123}, definition^{023}, definitions^{3}, dependency structures^{024}, described^{01234}, details^{0123}, development^{02}, edges^{123}, em^{0}, empty^{02}, entity^{4}, equation^{02}, experimental^{3}, expression^{03}, family^{1234}, feature^{3}, features^{034}, files^{0123}, find^{14}, followed^{01}, following^{01234}, following smith^{01234}, framework^{0134}, gaussian^{13}, generative^{4}, given^{123}, implemented^{01234}, incremental^{03}, initial^{134}, inspired^{01234}, instances^{0124}, introduced^{124}, isomorphic^{0124}, items^{1}, labeled^{4}, labeling^{4}, language model^{0234}, left^{0124}, lexical sets^{0124}, lexicalized^{024}, likelihood^{3}, log^{0123}, lower^{4}, method^{0234}, metric^{012}, model^{0234}, modeling^{13}, more details^{0123}, motivated^{034}, mrs^{0123}, negra^{0124}, noun^{0}, optimality^{0134}, optimality theory^{0134}, overlap^{0134}, paper^{01234}, parameters^{0234}, parsing model^{1234}, perceptron^{2}, phrase^{02}, phrase structure^{2}, position^{0123}, present^{1}, presented^{0123}, previous work^{01234}, procedure^{13}, proximity^{134}, pruning^{2}, ptb^{0234}, purposes^{124}, r2^{1234}, rate^{4}, realizer^{0134}, references^{1}, requires^{2}, respectively^{014}, right^{0124}, same^{4}, segmentation^{4}, semantics^{0123}, sense^{0134}, senses^{014}, set^{0234}, sets^{01234}, similar^{4}, smoothing^{1234}, speaker^{0134}, standard^{0}, surface^{013}, surface realizer^{0134}, symbolic^{0134}, templates^{24}, term^{0234}, test^{0}, test data^{013}, testing^{012}, thesaurus^{0234}, three^{1}, toronto^{1234}, training^{01234}, training data^{023}, training set^{04}, transform^{234}, transformation^{34}, two^{3}, used^{2}, using^{23}, value^{1234}, version^{01}, word^{04}, wsj^{0124}, yi^{123}

Continued on next page

Table B.1: (continued)

Label	Cue phrase
PERF	...^{0,1}, accuracy^{2}, achieved^{1}, acquired^{1}, against^{0134}, allows^{2}, already^{234}, anchor^{0234}, annotation^{4}, approaches^{01234}, assumption^{24}, automatically^{1}, bank^{1}, baseline^{2}, best^{2}, better^{3}, between^{03}, chinese^{1}, classifier^{2}, collocation^{124}, common^{2}, compare^{2}, considered^{0}, constructed^{03}, corpora^{01234}, corpus^{01234}, current^{13}, developed^{2}, dialogue^{014}, different^{023}, discourse^{01234}, discourse markers^{0234}, discriminative^{0}, discussed^{3}, documents^{124}, domain^{04}, e.g.^{01234}, e1^{1}, entities^{0234}, erg^{1}, evaluate^{01234}, evaluation^{1234}, example^{01234}, examples^{0}, expressions^{03}, extracted^{0123}, extraction^{01234}, f-score^{0134}, f-structure^{013}, finding^{0234}, focus^{3}, four^{14}, general^{2}, good^{01234}, grammar^{3}, grammars^{3}, head^{4}, human^{0234}, important^{3}, improve^{4}, include^{04}, including^{4}, information^{0124}, instance^{1}, instead^{0123}, introduction^{01234}, large^{01234}, ldd^{0134}, ldd resolution^{014}, learning^{2}, less^{0124}, lfg^{4}, lists^{1}, literature^{023}, machine^{0134}, machine learning^{3}, machine translation^{3}, markers^{2}, means^{3}, mereological^{04}, methods^{0124}, models^{01234}, n-gram^{0}, natural^{01234}, natural language^{01234}, new^{01234}, np^{0124}, one^{0}, output^{3}, over^{1}, pairs^{014}, parallel^{2}, parsing^{0}, part^{14}, particular^{1234}, pos^{0234}, probabilities^{3}, problem^{124}, provide^{1}, provided^{4}, query^{23}, recall^{13}, recent^{01234}, recognition^{0}, reference^{0}, related^{01234}, related work^{01234}, relations^{4}, relationships^{2}, reported^{234}, research^{4}, resolution^{01234}, resources^{03}, result^{1}, results^{13}, retrieval^{4}, rules^{13}, selection^{2}, semantic^{3}, sentence^{24}, show^{0}, shown^{0123}, simple^{1}, smith used^{2}, speech^{0}, state^{134}, string^{3}, studies^{0124}, subcategorisation^{0}, successfully^{0}, such^{0134}, summarization^{1234}, systems^{01234}, tag^{1}, tagging^{2}, task^{2}, temporal^{01234}, texts^{0123}, through^{03}, times^{1}, transducers^{1}, translation^{1234}, trigram^{1}, type^{4}, types^{4}, unit^{0124}, units^{0}, useful^{01234}, various^{2}, verb^{3}, verbs^{01234}, web^{01234}, weighted^{4}, well^{0234}, within^{4}, wordnet^{023}, words^{2}

Continued on next page

Table B.1: (continued)

Label	Cue phrase
CONF	's^{4}, according^{01234}, algorithm^{01234}, although^{0234}, applied^{1}, automatically^{0}, baseline^{4}, between^{01234}, candidate^{0}, case^{3}, centering^{0123}, classification^{1}, classifier^{0123}, clustering^{0124}, coherence^{3}, compute^{23}, computed^{1}, constraints^{2}, context^{0134}, corpora^{0}, corpus^{01234}, cost^{0234}, current^{0}, data^{14}, defined^{0124}, described^{0}, details^{01234}, developed^{0134}, discourse^{0234}, discourse markers^{4}, each^{01234}, em^{14}, entropy^{34}, error^{3}, experiment^{034}, experiments^{3}, extracted^{123}, feature^{02}, features^{014}, first^{0234}, following^{01234}, follows^{2}, form^{2}, found^{1}, function^{13}, functions^{234}, german^{0134}, given^{034}, i.e.^{2}, implementation^{2}, important^{01234}, including^{0234}, information^{01234}, input^{0}, introduced^{24}, introduction^{01234}, kernel^{012}, knowledge^{0}, language^{0}, language model^{0}, learning^{234}, lexical^{01234}, linear^{3}, list^{13}, local^{2}, markers^{0234}, maximum^{0134}, method^{1}, metrics^{1234}, multimodal^{034}, n-gram^{1}, new^{124}, np^{0124}, number^{01234}, obtained^{01234}, order^{1}, out^{2}, paper^{01234}, parse^{2}, penn^{3}, penn treebank^{3}, performance^{1}, phrase^{0}, pos^{0234}, present^{1}, presented^{0}, previous work^{0}, process^{134}, provided^{4}, question^{0123}, ranking^{134}, recent^{03}, recently^{4}, references^{3}, relations^{14}, research^{01234}, retrieval^{124}, rules^{1}, scf^{0124}, scfs^{0124}, score^{2}, second^{01234}, section^{01234}, see^{3}, selection^{0124}, semantic^{1}, sense^{24}, sentences^{01234}, sequence^{01234}, shown^{124}, simple^{01234}, speech^{2}, state^{34}, strategy^{123}, suggested^{0234}, syntactic^{01234}, systems^{01234}, tagger^{3}, tagging^{0}, target^{0134}, task^{3}, tasks^{0}, technique^{14}, test^{01234}, testing^{1}, text^{0123}, theory^{3}, those^{2}, three^{2}, time^{2}, training^{4}, two^{01234}, type^{3}, types^{2}, unit^{124}, used^{01234}, useful^{0234}, using^{01234}, value^{01234}, verbs^{1}, version^{0124}, way^{0}, web^{0}, weighted^{1}, words^{3}, wsj^{1}

Continued on next page

Table B.1: (continued)

Label	Cue phrase
NEG	accuracy[0], achieve[12], achieves[013], actual[024], addressed[12], against[0134], algorithms[0], alignment[01234], alignments[02], approach[0134], approaches[34], appropriate[2], approximately[134], association[0234], assumption[123], based[1234], before[2], behind[0124], better[01234], bleu[0234], boosting[01234], both[13], cfg[013], chinese[1234], compare[024], compared[23], comparison[0134], computation[134], concept[0134], constituents[012], constraint[013], contexts[124], criteria[2], dependency[02], dependency structure[04], descriptions[012], development[2], dienes[0124], different[01234], discriminative[023], distance[2], domains[3], dynamic[123], e.g.[4], entire[02], entities[024], evaluating[0234], even[1], expression[2], extract[0234], finding[12], fine-grained[14], fixed[0124], formal[014], general[1234], generative[3], generative model[12], gesture[0134], gold[034], gold standard[34], hand-crafted[0134], handle[0134], hidden[3], hidden markov[034], hierarchy[1], hpsg[1], human[0234], importance[4], improved[0], improvement[134], improves[034], independently[03], kind[1234], language pairs[13], large[1], lattice parser[123], ldd[0], ldds[0134], lfg[4], limited[0134], links[0234], log-linear[134], markov[3], markov models[03], medline[0234], methods[24], model[0124], models[01234], mt[234], much[01234], node[23], none[04], noun[1], noun phrase[123], observed[0234], obtain[0], outperform[014], over[0134], pairs[01234], parser[1], particular[12], path[0234], prediction[3], programming[1234], proposed[13], purpose[12], quality[024], rates[4], reasons[124], recall[04], reference[2], reference translations[023], related work[0], relationship[234], researchers[234], results[0], rule[1], scores[2], search[01], select[0], show[012], showing[01], small[014], smith[024], smith proposed[1234], smith use[234], solver[013], statistical[3], subset[3], success[13], successfully[02], template[034], temporal[1], times[0], translation[2], translation models[24], trigram[12], try[23], unlike[13], unsupervised[0124], upon[0124], van[2], versions[134], well[1], word alignment[0234]

Bibliography

Abu-Jbara, A. and Radev, D. R. (2012). Reference scope identification in citing sentences. In *Proceedings of NAACL-HLT*, pages 80–90. 80, 96, 101

Alencar, A. B., de Oliveira, M. C. F., and Paulovich, F. V. (2012). Seeing beyond reading: a survey on visual text analytics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):476–492. 117, 118

Alink, W., Cornacchia, R., and de Vries, A. P. (2009). Running CLEF-IP experiments using a graphical query builder. In *Proceedings of CLEF-IP 2009 Workshop*, pages 9:1–9:18. Springer. 121

Athar, A. (2011). Sentiment analysis of citations using sentence structure-based features. In *Proceedings of ACL Student Session*, pages 81–87. 84, 87, 89, 90, 93, 95, 96, 98, 109

Athar, A. and Teufel, S. (2012). Context-enhanced citation sentiment detection. In *Proceedings of NAACL-HLT*, pages 597–601. 96, 101, 109

Atkinson, K. H. (2008). Toward a more rational patent search paradigm. In *Proceedings of 1st workshop on Patent IR*, pages 37–40. 30, 33, 58, 60

Azzopardi, L., Vanderbauwhede, W., and Joho, H. (2010). Search system requirements of patent analysts. In *Proceedings of SIGIR*, pages 775–776. 30, 60, 121

Azzopardi, L. and Vinay, V. (2008). Retrievability: an evaluation measure for higher order information access tasks. In *Proceedings of CIKM*, pages 561–570. 52

Ballesteros, L. and Croft, W. B. (1997). Phrasal translation and query expansion techniques for cross-language information retrieval. In *Proceedings of SIGIR*, pages 84–91. 53, 61, 77

Bashir, S. and Rauber, A. (2009). Improving retrievability of patents with cluster-based pseudo-relevance feedback documents selection. In *Proceedings of CIKM*, pages 1863–1866. 52, 69, 76

Bashir, S. and Rauber, A. (2010). Improving retrievability of patents in prior-art search. In *Proceedings of ECIR*, pages 457–470. 52

Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.-Y., Lee, D., Powley, B., Radev, D., and Tan, Y. F. (2008). The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of LREC*, pages 1755–1759. 84

Bohnet, B. (2010). Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of COLING*, pages 89–97. 90

Bornmann, L. and Daniel, H.-D. (2008). What do citation counts measure? A review of studies on citing behavior. *Journal of Documentation*, 64(1):45–80. 108

Bouayad-Agha, N., Casamayor, G., Ferraro, G., Mille, S., Vidal, V., and Wanner, L. (2009). Improving the comprehension of legal documentation: the case of patent claims. In *Proceedings of 12th International Conference on Artificial Intelligence and Law*, pages 78–87. 52, 113

Brants, T. (2003). Natural language processing in information retrieval. In *Proceedings of Computational Linguistics in the Netherlands*, pages 1–13. 21

Braune, F. and Fraser, A. (2010). Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Proceedings of COLING (Posters)*, pages 81–89. 33, 60

Cano, V. (1989). Citation behavior: Classification, utility, and location. *Journal of the American Society for Information Science*, 40:284–290. 109

Carterette, B., Pavlu, V., Fang, H., and Kanoulas, E. (2009). Million query track 2009 overview. In *Proceedings of TREC*. 47

Chubin, D. E. and Moitra, S. D. (1975). Content analysis of references: Adjunct or alternative to citation counting? *Social Studies of Science*, 5:423–441. 108

Ciaramita, M. and Altun, Y. (2006). Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of EMNLP*, pages 594–602. 94

Collins, C., Penn, G., and Carpendale, M. S. T. (2008). Interactive visualization for computational linguistics. In *Proceedings of ACL (Tutorial Abstracts)*, page 6. 118

Councill, I. G., Giles, C. L., and Kan, M.-Y. (2008). ParsCit: An open-source CRF reference string parsing package. In *Proceedings of LREC*, pages 661–667. 79

Darwish, K. and Oard, D. W. (2003). Probabilistic structured query methods. In *Proceedings of SIGIR*, pages 338–344. 37

de Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, pages 449–454. 88

Di Eugenio, B. and Glass, M. (2004). The kappa statistic: A second look. *Computational Linguistics*, 30(1):95–101. 86

Diaz, F. and Metzler, D. (2007). Pseudo-aligned multilingual corpora. In *Proceedings of IJCAI*, pages 2727–2732. 54

Dong, C. and Schäfer, U. (2011). Ensemble-style self-training on citation classification. In *Proceedings of IJCNLP*, pages 623–631. 80, 84, 87, 89, 90, 92, 93, 95, 96, 97, 98, 107, 109

Dunne, C., Shneiderman, B., Gove, R., Klavans, J., and Dorr, B. J. (2012). Rapid understanding of scientific paper collections: Integrating statistics, text analytics, and visualization. *JASIST*, 63(12):2351–2369. 118

Dunning, T. and Davis, M. W. (1992). A single language evaluation of a multi-lingual text retrieval system. In *Proceedings of TREC*, pages 193–198. 53

Franz, M., McCarley, J. S., Ward, T., and Zhu, W.-J. (2001). Quantifying the utility of parallel corpora. In *Proceedings of SIGIR*, pages 398–399. 54, 59

Ganguly, D., Leveling, J., Magdy, W., and Jones, G. J. F. (2011). Patent query reduction using pseudo relevance feedback. In *Proceedings of CIKM*, pages 1953–1956. 53

Gao, W., Niu, C., Nie, J.-Y., Zhou, M., Wong, K.-F., and Hon, H.-W. (2010). Exploiting query logs for cross-lingual query suggestions. *ACM Transactions on Information Systems*, 28(2):6:1–6:33. 53, 64

Garfield, E. (1955). Citation indexes to science: A new dimension in documentation through association of ideas. *Science*, 122:108–111. 16

Garfield, E. (1964). Can citation indexing be automated? In *Statistical Association Methods for Mechanized Documentation, Symposium Proceedings*, pages 189–192. 108

Giereth, M., Koch, S., Kompatsiaris, Y., Papadopoulos, S., Pianta, E., Serafini, L., and Wanner, L. (2007). A modular framework for ontology-based representation of patent information. In *Proceedings of Conference on Legal Knowledge and Information Systems*, pages 49–58. 119

Goto, I., Lu, B., Chow, K. P., Sumita, E., and Tsou, B. K. (2011). Overview of the patent machine translation task at the NTCIR-9 workshop. In *Proceedings of 9th NTCIR Workshop*, pages 559–578. 51

Graf, E. and Azzopardi, L. (2008). A methodology for building a patent test collection for prior art search. In *Proceedings of 2nd Workshop on Evaluating Information Access*, pages 60–71. 51

Graf, E., Azzopardi, L., and van Rijsbergen, K. (2009). Automatically generating queries for prior art search. In *Proceedings of CLEF*, pages 480–490. 38, 52

Gupta, S. and Manning, C. (2011). Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of IJCNLP*, pages 1–9. 114

Hall, D., Jurafsky, D., and Manning, C. D. (2008). Studying the history of ideas using topic models. In *Proceedings of EMNLP*, pages 363–371. 114

Haveliwala, T. (2003). Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796. 114

Heimerl, F., Jochim, C., Koch, S., and Ertl, T. (2012a). FeatureForge: A novel tool for visually supported feature engineering and corpus revision. In *Proceedings of COLING (Posters)*, pages 461–470. 26, 115, 123, 124

Heimerl, F., Koch, S., Bosch, H., and Ertl, T. (2012b). Visual classifier training for text document retrieval. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 18(12):2839–2848. 118, 122, 123

Jochim, C., Lioma, C., and Schütze, H. (2011). Expanding queries with term and phrase translations in patent retrieval. In *Proceedings of IRFC*, pages 16–29. 25, 26

Jochim, C., Lioma, C., Schütze, H., Koch, S., and Ertl, T. (2010). Preliminary study into query translation for patent retrieval. In *Proceedings of 3rd workshop on Patent IR*, pages 57–66. 25, 26

Jochim, C. and Schütze, H. (2012). Towards a generic and flexible citation classifier based on a faceted classification scheme. In *Proceedings of COLING*, pages 1343–1358. 26, 86

Jurafsky, D. and Martin, J. H. (2000). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall. 24

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 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, pages 88–95. 96

Kettunen, K. (2009). Choosing the best MT programs for CLIR purposes - can MT metrics be helpful? In *Proceedings of ECIR*, pages 706–712. 59, 66

Koch, S., Bosch, H., Giereth, M., and Ertl, T. (2011). Iterative integration of visual insights during scalable patent search and analysis. *IEEE Trans. Vis. Comput. Graph.*, 17(5):557–569. 119, 120, 121

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: open source toolkit for statistical machine translation. In *Proceedings of ACL*, pages 177–180. 60

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of NAACL*, pages 48–54. 60, 76

Kraaij, W., Nie, J.-Y., and Simard, M. (2003). Embedding web-based statistical translation models in cross-language information retrieval. *CoRR*. 53, 54, 59, 75

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174. 86

Larkey, L. S. (1999). A patent search and classification system. In *Proceedings of ACM conference on Digital Libraries*, pages 179–187. 51, 121

Lavrenko, V., Choquette, M., and Croft, W. B. (2002). Cross-lingual relevance models. In *Proceedings of SIGIR*, pages 175–182. 53

Lavrenko, V. and Croft, W. B. (2001). Relevance based language models. In *Proceedings of SIGIR*, pages 120–127. 66

Lefever, E., Macken, L., and Hoste, V. (2009). Language-independent bilingual terminology extraction from a multilingual parallel corpus. In *Proceedings of EACL*, pages 496–504. 54

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397. 123

Lioma, C. A. (2008). *Part of speech N-grams for information retrieval*. PhD thesis, University of Glasgow. 23

Liu, M. (1993). Progress in documentation the complexities of citation practice: A review of citation studies. *Journal of Documentation*, 49(4):370–408. 108

Lopez, P. and Romary, L. (2009). Multiple Retrieval Models and Regression Models for Prior Art Search. In *Proceedings of CLEF-IP 2009 Workshop*, pages 8:1–8:18. 52

Lupu, M. (2012). Patent information retrieval: an instance of domain-specific search. In *Proceedings of SIGIR*, pages 1189–1190. 51

Lupu, M., Mayer, K., Tait, J., and Trippe, A., editors (2011). *Current challenges in patent information retrieval*. Springer. 51

Magdy, W. (2012). *Toward higher effectiveness for recall-oriented information retrieval: A patent retrieval case study*. PhD thesis, Dublin City University. 20

Magdy, W. and Jones, G. J. (2010). A new metric for patent retrieval evaluation. In *Prodeedings of Workshop on Advances in Patent Information Retrieval*. 30, 40, 66

Magdy, W. and Jones, G. J. (2011a). A study on query expansion methods for patent retrieval. In *Proceedings of 4th workshop on Patent IR*, pages 19–24. 52, 54, 69, 76

Magdy, W. and Jones, G. J. F. (2011b). An efficient method for using machine translation technologies in cross-language patent search. In *Proceedings of CIKM*, pages 1925–1928. 54, 76

Magdy, W., Lopez, P., and Jones, G. J. F. (2011). Simple vs. sophisticated approaches for patent prior-art search. In *Proceedings of ECIR*, pages 725–728. 38, 52

Mahdabi, P., Andersson, L., Keikha, M., and Crestani, F. (2012). Automatic refinement of patent queries using concept importance predictors. In *Proceedings of SIGIR*, pages 505–514. 53, 74, 76

Mahdabi, P. and Crestani, F. (2012). Learning-based pseudo-relevance feedback for patent retrieval. In *Proceedings of IRFC*, pages 1–11. 53, 76

Mahdabi, P., Keikha, M., Gerani, S., Landoni, M., and Crestani, F. (2011). Building queries for prior-art search. In *Proceedings of IRFC*, pages 3–15. 38

Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press. 24, 123

Manning, C. D. and Klein, D. (2003). Optimization, maxent models, and conditional estimation without magic. In *Proceedings of HLT-NAACL*, page 8. 95

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. 41, 95, 137

Mayer, T., Rohrdantz, C., Butt, M., Plank, F., and Keim, D. A. (2011). Visualizing vowel harmony. *Linguistic Issues in Language Technology*, 4(1). 117

McCain, K. W. and Turner, K. (1989). Citation content analysis and aging patterns of journal articles in molecular genetics. *Scientometrics*, 17(1–2):127–163. 109

Metzler, D. and Croft, W. B. (2004). Combining the language model and inference network approaches to retrieval. *Inf. Process. Manage.*, 40(5):735–750. 65

Moed, H. F. (2005). *Citation Analysis in Research Evaluation*. Springer. 80, 107

Moravcsik, M. J. and Murugesan, P. (1975). Some results on the function and quality of citations. *Social Studies of Science*, 5:86–92. 81, 82, 112, 127, 134

Nakagawa, H., Torisawa, K., and Kitsuregawa, M. (2008). WWW 2008 workshop: NLPIX2008 summary. In *Proceedings of WWW*, pages 1277–1278. 15

Nanba, H., Abekawa, T., Okumura, M., and Saito, S. (2004). Bilingual PRESRI - integration of multiple research paper databases. In *Proceedings of RIAO*, pages 195–211. 80

Nanba, H. and Okumura, M. (1999). Towards multi-paper summarization using reference information. In *Proceedings of IJCAI*, pages 926–931. 87, 89, 109

Nie, J.-Y., Simard, M., Isabelle, P., and Durand, R. (1999). Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the web. In *Proceedings of SIGIR*, pages 74–81. 54

Noreen, E. (1989). *Computer-intensive methods for testing hypotheses: An introduction*. Wiley. 98

Oard, D. W. and Diekema, A. R. (1998). Cross-language information retrieval. *Annual Review of Information Science and Technology*, 33:223–256. 24

Oard, D. W., He, D., and Wang, J. (2008). User-assisted query translation for interactive cross-language information retrieval. *Inf. Process. Manage.*, 44(1):181–211. 53, 75

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51. 34, 60

Oelke, D., Momtazi, S., and Keim, D. A. (2012). Natural language processing for text visualization. Tutorial at VisWeek 2012. 118, 125

Osborn, M., Strzalkowski, T., and Marinescu, M. (1997). Evaluating document retrieval in patent database: a preliminary report. In *Proceedings of CIKM*, pages 216–221. 51

Peng, F. and McCallum, A. (2004). Accurate information extraction from research papers using conditional random fields. In *Proceedings of HLT-NAACL*, pages 329–336. 79

Penn, G., Carpendale, S., and Collins, C. (2009). Interactive visualization for computational linguistics. Tutorial at ESSLLI 2009. 118, 125

Peters, C., Nunzio, G. M. D., Kurimo, M., Mandl, T., Mostefa, D., Peñas, A., and Roda, G., editors (2010). *Multilingual Information Access Evaluation Vol. I Text Retrieval Experiments - Tenth Workshop of the Cross-Language Evaluation Forum (CLEF 2009). Revised Selected Papers*, Lecture Notes in Computer Science (LNCS). Springer. 155

Popovic, M., Stein, D., and Ney, H. (2006). Statistical machine translation of German compound words. In *Proceedings of FinTAL*, pages 616–624. 45

Qazvinian, V. and Radev, D. R. (2008). Scientific paper summarization using citation summary networks. In *Proceedings of COLING*, pages 689–696. 80, 113, 118

Qazvinian, V., Radev, D. R., and Özgür, A. (2010). Citation summarization through keyphrase extraction. In *Proceedings of COLING*, pages 895–903. 80, 118

Radev, D. R., Muthukrishnan, P., and Qazvinian, V. (2009). The ACL anthology network. In *Proceedings of 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, pages 54–61. 107

Ritchie, A., Robertson, S., and Teufel, S. (2008). Comparing citation contexts for information retrieval. In *Proceedings of CIKM*, pages 213–222. 80, 96

Ritchie, A., Teufel, S., and Robertson, S. (2006). How to find better index terms through citations. In *Proceedings of Workshop on How Can Computational Linguistics Improve Information Retrieval?*, pages 25–32. 80

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The SMART Retrieval System – Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall. 32, 40, 52

Roda, G., Tait, J., Piroi, F., and Zenz, V. (2009). CLEF-IP 2009: Retrieval experiments in the intellectual property domain. In *Proceedings of CLEF*, pages 385–409. Springer. 37, 52, 59, 64

Rubens, N. (2006). The application of fuzzy logic to the construction of the ranking function of information retrieval systems. *Computer Modelling and New Technologies*, 10(1):20–27. 40

Salton, G. and Buckley, C. (1990). Improving retrieval performance by relevance feedback. *JASIS*, 41(4):288–297. 52

Schmid, H. (1995). Improvements in part-of-speech tagging with an application to German. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50. 84

Schütze, H. (2010). IR, NLP, and visualization. In *ECIR*, page 11. 21

Small, H. G. and Griffith, B. C. (1974). The structure of scientific literatures I: Identifying and graphing specialties. *Science Studies*, 4(1):17–40. 114

Smucker, M. D., Allan, J., and Carterette, B. (2007). A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of CIKM*, pages 623–632. 42, 44, 68

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21. 39

Spiegel-Rösing, I. (1977). Science studies: Bibliometric and content analysis. *Social Studies of Science*, 7:97–113. 109

Stasko, J., Gorg, C., Liu, Z., and Singhal, K. (2007). Jigsaw: Supporting investigative analysis through interactive visualization. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, pages 131–138. 118

Stymne, S. (2008). German compounds in factored statistical machine translation. In *Proceedings of 6th Conference on Natural Language Processing (GoTAL-08)*. 45

Sugiyama, K., Kumar, T., Kan, M.-Y., and Tripathi, R. C. (2010). Identifying citing sentences in research papers using supervised learning. In *Proceedings of CIKM*, pages 67–72. 80

Sun, L., Jin, Y., Du, L., and Sun, Y. (2000). Word alignment of English-Chinese bilingual corpus based on chunks. In *EMNLP-VLC*, pages 110–116. 54

Tait, J., editor (2008). *1st ACM workshop on Patent IR*. ACM. 29, 30

Tait, J., editor (2009). *2nd ACM workshop on Patent IR*. ACM. 30

Teufel, S. (1999). *Argumentative Zoning: Information Extraction from Scientific Articles*. PhD thesis, University of Edinburgh. 19

Teufel, S., Siddharthan, A., and Tidhar, D. (2006a). An annotation scheme for citation function. In *Proceedings of SIGdial Workshop on Discourse and Dialogue*, pages 80–87. 109

Teufel, S., Siddharthan, A., and Tidhar, D. (2006b). Automatic classification of citation function. In *Proceedings of EMNLP*, pages 103–110. 80, 81, 84, 87, 89, 90, 92, 93, 95

Thomas, J. J. and Cook, K. A., editors (2005). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Center. 118

Tiwana, S. and Horowitz, E. (2009). Findcite: automatically finding prior art patents. In *2nd international workshop on Patent IR*, pages 37–40. 30

Toucedo, J. C. and Losada, D. E. (2010). University of Santiago de Compostela at CLEF-IP09. In Peters et al. (2010). 52

Wang, J. and Oard, D. W. (2006). Combining bidirectional translation and synonymy for cross-language information retrieval. In *Proceedings of SIGIR*, pages 202–209. 36, 53, 54, 59, 64

Wanner, L., Brügmann, S., Diallo, B., Giereth, M., Kompatsiaris, Y., Pianta, E., Rao, G., Schoester, P., and Zervaki, V. (2006). Patexpert: Semantic processing of patent documentation. In *Proceedings of Conference on Semantic and Digital Media Technologies (Posters)*. 119

Wäschle, K. and Riezler, S. (2012a). Analyzing parallelism and domain similarities in the MAREC patent corpus. In *Proceedings of IRFC*, pages 12–27. 77

Wäschle, K. and Riezler, S. (2012b). Structural and topical dimensions in multi-task patent translation. In *Proceedings of EACL*, pages 818–828. 77

Weinstock, M. (1971). *Encyclopedia of Library and Information Science*, volume 5, chapter Citation indexes. Dekker, New York, NY. 108, 109

White, H. D. (2004). Citation analysis and discourse analysis revisited. *Applied Linguistics*, 25(1):89–116. 107

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT-EMNLP*, pages 347–354. 93

Xue, X. and Croft, W. B. (2009). Automatic query generation for patent search. In *Proceedings of CIKM*, pages 2037–2040. 38, 52, 65

Yang, Y., Carbonell, J. G., Brown, R. D., and Frederking, R. E. (1998). Translingual information retrieval: Learning from bilingual corpora. *Artif. Intell.*, 103(1-2):323–345. 54

Yogatama, D., Heilman, M., O'Connor, B., Dyer, C., Routledge, B. R., and Smith, N. A. (2011). Predicting a scientific community's response to an article. In *Proceedings of EMNLP*, pages 594–604. 113