FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Boolean query characteristics in patent searching

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Media Informatics and Visual Computing

by

Roland Honeder
Registration Number 0107472

to the Faculty of Informatics

at the TU Wien

Advisor: Priv.-doz. Dr. Allan Hanbury

Vienna, 6th August, 2018

Roland Honeder

Allan Hanbury

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

Erklärung zur Verfassung der Arbeit

Roland Honeder
Nondorf an der Wild 10, 3754 Irnfritz-Messern

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 6. August 2018

Roland Honeder

Kurzfassung

Eine steigende Zahl an Patentanträgen - im Verein mit mittlerweile über einer halben Million noch nicht untersuchter Patentanträge (Frühjahr 2016) im United States Patent & Trademark Office (USPTO) - bedingt die Durchführung effizienter Patentsuche, deren Ziel die Feststellung der Patentierbarkeit eines Patentantrages ist. Wesentliches Werkzeug hierfür sind Suchmaschinen, die die Verarbeitung Boolescher Suchanfragen durch den Patentsucher unterstützen.

Diese Arbeit untersucht Merkmale von mehr als 15 Millionen Boolescher Suchanfragen, die im Laufe der Patentsuche generiert und an das Examiner Assisted Search Tool (EAST) übermittelt wurden. Transkriptionen der Anfragen stehen in SRNT Dokumenten zum Download zur Verfügung. Der Sucherfolg spiegelt sich in der Verfügbarkeit eines 892 Dokumentes wieder. Ist für ein SRNT Dokument ein adäquates 892 Dokument vorhanden, bedeutet dies, dass im Rahmen der Suche relevante Patentliteratur entdeckt wurde. Darauf basierend ließ sich das Datenmaterial hinsichtlich ihres Sucherfolges in zwei Erfolgsgruppen ($SRNT_{892}$, $SRNT_{no892}$) teilen.

Query Expansion (QE) ist eine Maßnahme zur Verbesserung des Suchresultates durch das Hinzufügen relevanter Begriffe. Die verschiedenen Strategien werden im Rahmen dieser Arbeit erläutert.

In der Patentsuche wird QE manuell und auf zwei Arten angewandt. Einerseits durch das Einfügen alternative Suchbegriffe in eine Suchanfrage; andererseits durch die Verwendung des Truncation Operators, der Rücksichtnahme auf Wortvariationen (z.B. Endungen) erlaubt.

QE wird in Suchen beider Erfolgsgruppen häufig angewandt. Die Erwartung, sie sei verstärkt in der Gruppe erfolgreicher Suchen vorzufinden, wurde indes nur zum Teil erfüllt. Listen alternativer Suchbegriffe treten in beiden Gruppen mit ähnlicher Häufigkeit, der Truncation Operator vermehrt in der Gruppe erfolgreicher Suchen auf.

Weitere Untersuchungen - etwa über die Verwendung Boolescher Operatoren, der durchschnittlichen Querylänge, der Verschachtelungstiefe Boolescher Anfragen - zeigten ähnliche Resultate für beide Erfolgsgruppen. Ausgeprägtere Unterschiede ließen sich in der durchschnittlichen Dokumentlänge (= Anzahl Suchanfragen pro Suche), in der Verwendung der Suchfelder (die eine Suche z.B. nach Autorennamen ermöglicht) sowie der Verwendung von Referenzierung (= die Adressierung vorangegangener Suchanfragen) feststellen.

Abstract

As of early 2016, 550,000 patent applications submitted to the USPTO are unexamined. The considerable number of unexamined applications is likely to reduce the amount of time that patent examiners can spend on the examination of an application. Fast-paced technological progress, on the other hand, implies that patent examiners need to invest more time in patent examination. In light of these facts and with sometimes millions at stake the importance of conducting patent searches efficiently is obvious.

This thesis analyzes various characteristics of more than 15,000,000 Boolean search queries submitted by professional patent examiners to the EAST patent search engine at the USPTO. Search queries generated during the examination of a patent application are available from the USPTO as SRNT ("search related notes") documents.

In order to assemble a rather large dataset of search query logs, more than one million patent applications had to be retrieved and processed. The set of obtained SRNT documents was split into two groups. For one group ($SRNT_{892}$) relevant patents had been cited by the patent examiner. For the other group ($SRNT_{no892)}$), no relevant patent documents had been found during the search.

Query Expansion (QE) is a popular and well-studied technique for improving search results by adding relevant terms to an user query. Professional patent searchers apply QE manually. Either by providing lists of related terms within a Boolean query, or by using the truncation operator as instruction for the patent search engine to consider variations of a word. Contrary to my expectation it is shown that there is no difference between "successful" and "unsuccessful" searches in terms of quantity by which lists of alternate terms are provided. However, it is also shown that the use of the truncation operator is more popular in "successful" searches.

Most of the examined search features, such as the average query length, the use of parentheses, the use of Boolean operators, yield relatively similar results for both document sets. Noteworthy differences have been found in the average document length (= number of queries per search), in the use of patent database specific search fields (e.g. to search in the "claims" section of a patent) and in the use of references (to address former queries).

Contents

Kurzfassung **v**

Abstract **vii**

Contents **ix**

1 Introduction **3**
1.1 Problem Statement 3
1.2 Contribution of this work 5
1.3 Outline 7

2 Patent Law and Patent Office Operations **9**
2.1 Why Patents? 9
2.2 Historical Development 10
2.3 The Patent Examination Process in the United States 12
2.4 Patent Searching 18
2.5 Summary 23

3 Query Expansion **25**
3.1 Introduction 25
3.2 Corpus-specific global techniques 28
3.3 Approaches based on Linguistic-analysis 32
3.4 Relevance Feedback 35
3.5 Collaborative approaches and web data 41
3.6 Summary 41

4 Data Acquisition and Preparation **43**
4.1 Introduction 43
4.2 Data retrieval and initial processing 45
4.3 Data improvement 51
4.4 Summary 56

5 Data Analysis **57**
5.1 Introduction 57

5.2 General examinations 58
5.3 Use of Boolean operators 67
5.4 Use of Query Expansion 71

6 Conclusions **79**
6.1 Results 79
6.2 Accuracy of the results 81
6.3 Summary 81
6.4 Outlook and future work 81

A Using PyPAIR **83**
A.1 Preparing the download 83
A.2 Retrieving patent applications. 84
A.3 Transforming SRNT documents into text files 84
A.4 Data normalization 86
A.5 Calculating quality indicators and cleaning the dataset 87

Bibliography **91**

If we did not have a patent system, it would be irresponsible, on the basis of our present knowledge of its economic consequences, to recommend instituting one. But since we have had a patent system for a long time, it would be irresponsible, on the basis of our present knowledge, to recommend abolishing it.

Fritz Machlup, An Economic Review of the Patent System (1958)

CHAPTER 1

Introduction

1.1 Problem Statement

The responsibilities of a patent office such as the United States Patent and Trademark Office (USPTO) range from handling and examining patent applications to the grant of patents. The steps for prior art search are usually similar in patent offices around the world. A simplified representation of the process is as follows. After an invention has been made, the inventor prepares a patent application which is filed at the patent office of the country where a patent is desired to be obtained. In case the formal requirements of the patent application are met, a patent examiner will proceed to analyze the inventive concept described in the application for patentability. In the United States, the patentability criteria are novelty, non-obviousness[1] and usefulness of the invention.

In order to be able to assess the novelty aspect of an invention, a search for prior art is carried out during the examination process. Patent and non-patent documents are both relevant to this assessment. In order to be able to conduct an exhaustive search, the examiner is expected to consider an appropriate *search strategy.* At this stage, the results of the prior art search determine whether a patent can be granted, therefore the importance of an exhaustive search cannot be overstated. Missed relevant prior art may lead to patent infringement lawsuits[2] or may be used to invalidate a granted patent later.

Rapid growth

Being able to conduct exhaustive searches as effectively and efficiently as possible is stressed when taking a look at patent statistics over the last few years. The World Intellectual Property Organization (WIPO) reports that in 2011 the worldwide number

[1] To a person of ordinary skill in the subject matter.

[2] In 2013 the American Intellectual Property Law Association (AIPLA) conducted a survey with a participation of 1,799 of it's members. According to the survey, median litigation costs for patent infringement suits range from 700,000 USD (less than 1 Million USD at risk) to up to 5,000,000 USD (more than 25 Million USD at risk). [AIP13]

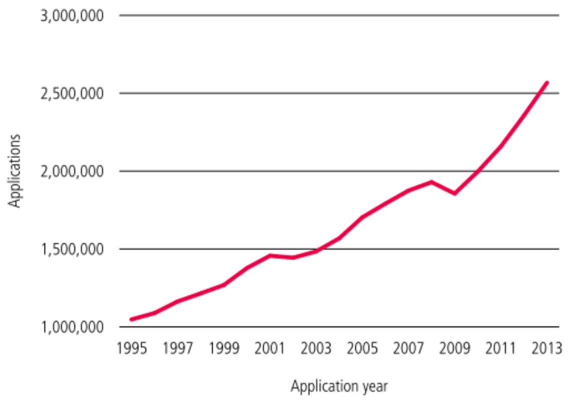


Figure 1.1: Patent applications worldwide (1995-2013). Source: [Wip14]

of patent applications exceeded the 2 million mark for the first time.[3] Nearly 1 million patents were granted - a growth rate of 9.7% compared to 2010. [Wip12]

In 2013 the worldwide number of patent applications already exceeded the 2.5 million mark. Between 2011 and 2013 the number of patent filings worldwide grew by more than 400,000.[4] Most patents were granted by the United States Patent and Trademark Office (USPTO) with 277,835 grants, followed closely by the Japan Patent Office (JPO) with 277,079 grants. More than 50% all patent applications worldwide in 2013 were filed at either the Chinese Patent Office[5] or at the USPTO[6]. In 2013 more than 7.5 million patents were in force worldwide, more than 2 million patents granted by the USPTO alone. [Wip14]

Backlog

In addition to the rapid growth figures the size of the backlog of unexamined patent applications is one of the biggest challenges the USPTO is facing. Current data[7] indicates that more than 550,000 patent applications are unexamined as of early 2016. The average time-span before a patent application is picked up for the first time is 16 months. One has to keep in mind that long pendency periods are likely to have a negative impact on innovators and competitors. Moreover, an already considerable size of the backlog will reduce the amount of time that patent examiners can invest in the examination of an application, whereas a fast-paced technological progress and an exploitation of new technological fields imply that more time needs to be spent for examination. A deterioration of the *quality* of granted patents is likely.

[3] 2011 was also the first year where the Chinese Patent Office (SIPO) has overtaken the USPTO as the world largest patent office, when measured by the number of filings for patents, utility models, trademarks and industrial designs.

[4] 72% of the growth can be attributed to the Chinese Patent Office (SIPO)

[5] 825,136 patent applications or 32.1% of all patent applications worldwide

[6] 571,612 patent applications or 22.2% of all patent applications worldwide

[7] See `http://www.uspto.gov/dashboards/patents/main.dashxml`

This thesis examines various characteristics of Boolean queries (e.g. the use of operators) submitted by patent searchers to the Examiner Automated Search Tool (EAST), a patent search system used at the USPTO's public search facility in Alexandria, VA. To achieve this, more than one million patent applications were downloaded from the USPTO. Included SRNT[8] files were extracted and processed with OCR to be able to store search data (e.g. search query string, date of search, number of results,...) on patent database searches in the form of structured text. More than 500,000 SRNT documents were processed. Relevant documents detected by the patent examiner during a search are cited as references within a 892 document. The presence or absence of an 892 document for a SRNT document was used to split the set of SRNT documents into group $SRNT_{892}$ (relevant documents were found during the search) and $SRNT_{no892}$ (no relevant documents were obtained).

Query Expansion (QE) is a well known approach to improve search results by expanding a user's query with further keywords, equivalents and synonyms of terms provided by the user.

By analyzing about 500,000 search query logs (see Figure 1.2) we expect to gain insight into the way patent application examiners construct Boolean queries and how QE techniques are applied in the process.

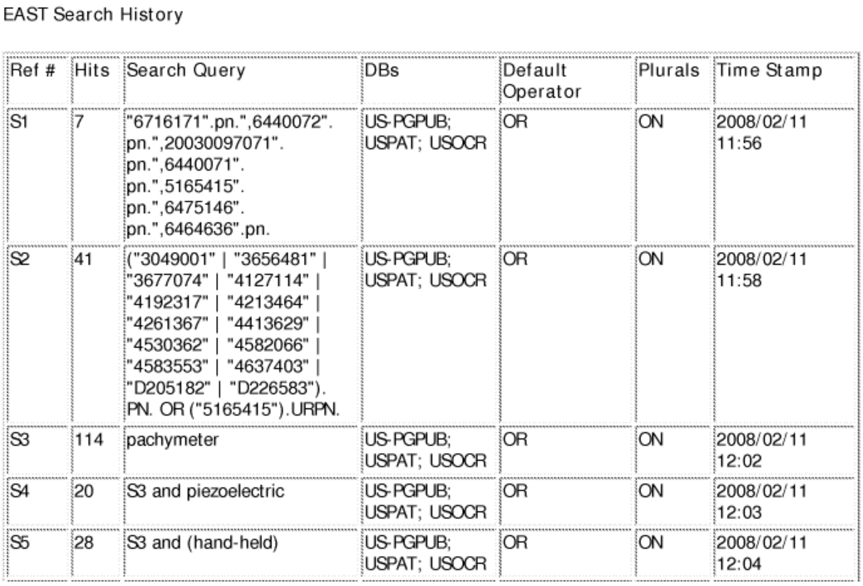
1.2 Contribution of this work

SRNT documents are only available in the form of scanned images. Assembling a well structured, textual representation of a rather large amount of SRNT documents was necessary. To achieve this, more than one million patent applications were retrieved from the USPTO. In order to extract Boolean patent search queries from SRNT *image* documents, it was necessary to implement algorithms that:

- understand the (usual) document layout of (typical) SRNT documents, i.e. detect table-like structures and the location of columns and rows,
- are capable of taking measures to guarantee the best result for OCR tasks, i.e. the removal of certain parts of an image and the application of image processing algorithms known to aid OCR,
- detect and remove badly processed documents from the dataset, in order to improve it's overall quality.

To my knowledge, this is the first time that such a massive amount of Boolean patent search queries has been assembled into a single, uniform dataset. This allows to examine various characteristics of Boolean search queries issued by professional patent searchers.

[8] A SRNT document contains transcriptions of searches performed for the patent application, e.g. patent database searches.



EAST Search History

<table>
<tr><th>Ref #</th><th>Hits</th><th>Search Query</th><th>DBs</th><th>Default Operator</th><th>Plurals</th><th>Time Stamp</th></tr>
<tr><td>S1</td><td>7</td><td>"6716171".pn.",6440072".
pn.",20030097071".
pn.",6440071".
pn.",5165415".
pn.",6475146".
pn.",6464636".pn.</td><td>US-PGPUB; USPAT; USOCR</td><td>OR</td><td>ON</td><td>2008/02/11 11:56</td></tr>
<tr><td>S2</td><td>41</td><td>("3049001" | "3656481" |
"3677074" | "4127114" |
"4192317" | "4213464" |
"4261367" | "4413629" |
"4530362" | "4582066" |
"4583553" | "4637403" |
"D205182" | "D226583").
PN. OR ("5165415").URPN.</td><td>US-PGPUB; USPAT; USOCR</td><td>OR</td><td>ON</td><td>2008/02/11 11:58</td></tr>
<tr><td>S3</td><td>114</td><td>pachymeter</td><td>US-PGPUB; USPAT; USOCR</td><td>OR</td><td>ON</td><td>2008/02/11 12:02</td></tr>
<tr><td>S4</td><td>20</td><td>S3 and piezoelectric</td><td>US-PGPUB; USPAT; USOCR</td><td>OR</td><td>ON</td><td>2008/02/11 12:03</td></tr>
<tr><td>S5</td><td>28</td><td>S3 and (hand-held)</td><td>US-PGPUB; USPAT; USOCR</td><td>OR</td><td>ON</td><td>2008/02/11 12:04</td></tr>
</table>

Figure 1.2: Example of an SRNT document containing search queries issued by a patent examiner at the start of his search, logged by the EAST system.

The result of the examination will be helpful in answering various questions regarding prior art searching, such as:

- How many search queries do SRNT documents contain on average?
- How do patent examiners make use of query expansion? How many synonyms and related terms are used on average to expand a query term?
- Are searches with a thorough use of query expansion more likely to find relevant patent documents? Does the number of used synonyms and related terms influence the finding of relevant patent documents?
- How popular are the various Boolean operators (AND, OR, NEAR, WITH, SAME, ...)?
- How do various parameters (e.g. average query length, usage of operators) of search sessions and queries differ between $SRNT_{892}$ ("successful searches") and $SRNT_{no892}$ ("unsuccessful searches")? Do they differ at all?

The answers to these questions might be beneficial for performing a supplementary assessment of the *quality* of a prior art search; they could also provide useful hints for patent searchers as well as for enhancements made to patent search engines.

1.3 Outline

This thesis is structured as follows. In Chapter 2, general concepts of the patent law, it's historical development and the challenges of patent searching are briefly examined.

In Chapter 3 Query Expansion is discussed, providing an overview of various Query Expansion techniques and strategies.

Chapter 4 discusses the generation of the dataset. The results of the analysis are presented in Chapter 5. Finally, the conclusions drawn from the analysis are summarized in Chapter 6.

CHAPTER 2

Patent Law and Patent Office Operations

This chapter provides background information for this thesis. It briefly discusses the idea behind todays patent system and it's historical development. Then, the general mechanics of patent searching are presented.

2.1 Why Patents?

The rationale behind the establishment of a patent system is stated in Article 1, Section 8, Clause 8 of the U.S. Constitution:

> "To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries."[a]
>
> [a] U.S. CONST. art. I, $ 8, cl. 8.

A patent is an agreement - a quid pro quo - between government and inventor. By laying an invention open (in Latin, the adjective patens means open), innovation and technical development are promoted. This, in turn, leads to economic growth. The inventor on the other hand is given an exclusive right for a limited time for using, making or selling his/her invention. In other words, a patent is a legal grant by the government to exclude others from using, making or selling an invention. The concept of exclusion is fundamental. This concept aims to solve the so called "appropriability problem" [Dam94], which means that an inventor could not recover the costs of invention without a patent's protective force; information gained from the invention could be used commercially by others. Also, the level of innovation would arguably be lower. On the other hand, the issue of patents of low quality or even absurd patents[1] has been complained about as being detrimental to innovation. [Sha04]

In a nutshell, the patent system serves as 1. protective force regarding an inventor's idea, and 2. stimulation to the economic growth and technological development.

Discussing the patent law in all it's facets and failings would, even if conducted extremely superficially, reach far beyond the scope of this thesis. However, two core concepts shall be mentioned at this point:

First To Invent / First To File

It goes without saying that a government will try to assign a patent to the correct (i.e., first) inventor. Cases where an invention is made at the same time by two different parties are, of course, not ruled out. There are two concepts to deal with this issue:

- First-To-File: here, the inventor who *applies* earlier for a patent is recognized as inventor. This implies that companies usually try to file a patent application as quickly as possible.
- First-To-Invent: here, the first true inventor behind an inventive concept is (tried to be) identified in a potentially costly process. Evidence for inventor-ship can, for example, be provided by former inventions of the applicant. With the Patent Reform Act of 2011 the USA changed form the First-To-Invent system to the First-To-File system.

Priority Date

The priority date is the earliest filing date of a patent application. This date establishes the novelty of an invention compared to other art. Since it is an absolutely fundamental concept to Patent law, it is crucial to many types of patent searches.

2.2 Historical Development

This section outlines the historical development of the patent system.

2.2.1 Early History

The possibly earliest description of a practice that can be considered a forerunner of today's patent law can be found in the Deipnosophistai[2], main work of Greek rhetorician Athenaeus of Naucratis. He describes the concept of exclusion in the city of Sybaris, 500BC:

[1] An often mentioned example in this regard is Amazon's "one-click" patent (US 5960411).

[2] "Philosophers at Dinner."

"[I]f any confectioner or cook invented any peculiar and excellent dish, no other artist was allowed to make this for a year; but he alone who invented it was entitled to all the profit to be derived from the manufacture of it for that time; in order that others might be induced to labor at excelling in such pursuit."

(Deipnosophistai, Vol.3, Book XII, Ch. 20) (see: [Spi11])

It is surprising to see that the cornerstones of today's patent system (protection by exclusion, protection for a limited amount of time) are encompassed by this statement - as is the ruler's hope that it would stimulate "innovation" or, at least, creativity.

2.2.2 Venetian patent law

Royal grants

In England, royal grants (open letters, letters patent in Latin) were provided to a recipient with a monopoly as early as the 14th century, the earliest authenticated instance being provided in 1331 to John Kempe. These grants were issued in order to import existing trades to England and not in respect of new inventions. As pointed out in [Mgb03], claims that the grant to John Kempe is a progenitor of the modern patent system lack merit due to the missing aspect of novelty. In Florence, architect Filippo Brunelleschi received an open letter in 1421 following his *demand* that his invention (an iron clad sea-craft, which, by his claim, could transport marble across the lake Arno[3]) must be protected.

First formal patent system

However it is the Venetian Republic that is widely recognized as having the first formal patent system, following the Patent Act of 1474. Claimed inventions were examined by a General Welfare Board. The protection duration lasted for ten years; infringements of patent grants or unauthorized use were punished. Already in 1474 a registry of patents was set up. In the sixty years between 1490 and 1550 over 120 patents were granted, most of them dealing with water mills, pumps and similar mechanical devices[Mgb03].

Via emigration the ideas of a patent system were spread further over Central and Western Europe.

2.2.3 Development of the Patent Law in the United States Of America

Three prominent patent acts - 1790, 1793, 1836 - shaped the US patent system.

Patent Act of 1790 On April 10, 1790 the first United States patent statute was enacted on; the first patent (patent X000001) was granted to Samuel Hopkins of Philadelphia on July 31, 1790. The patent was granted for an improvement in the making of pot ash and pearl ash by a new apparatus and process. Among the first patent board members was Thomas Jefferson, at this time Secretary of

[3] On a side note: the sea-craft sank on it's first trip on lake Arno.

State and considered the first patent examiner.[4] Patents could be granted by the patent board members for "any useful art, manufacture, engine, machine, or device, or any improvement therein not before known or used". Furthermore, the invention had to be "sufficiently useful and important" (Patent Act of 1790, Ch.7, 1 Stat. 109-112). The duration for each patent was not allowed to exceed 14 years. The exact duration, however, was decided by the patent board on a case-by-case basis. Applications had to provide the specification of the patent, a drawing and, if possible, a model of the invention.

Patent Act of 1793 In the first four years after the Patent Act of 1790 the number of granted patents was moderate: in 1790, three patents were granted; in 1791, 33 patents were granted; in 1792, 11 patents were granted; and in 1973, 20 patents were granted. [Pat] The number of patent applicants was quite likely much higher. This fact made it difficult for the patent board members to examine each application. It was therefore concluded and, in the Patent Act of 1793, stated, that *no* patent examination was carried out prior to grant of a patent. Of course, patents were still limited to "any new and useful art, machine, manufacture or composition of matter, or any new and useful improvement on any art, machine, manufacture or composition of matter"[5] (Patent Act of 1793, Ch.11, 1 Stat. 318-323)

The importance of a new invention is often hard to determine. Therefore, the term *important* was removed from the statement. Every claimed invention that was applied for with a proper and formally correct patent application was granted a patent, even if the "invention" was an obvious copy of a previous invention. The courts were responsible to clarify a patents validity on the grounds of it's novelty.

Patent Act of 1836 The Patent Act of 1836 saw the introduction of a middle-way between the strict approach of the Patent Act of 1790 and the acceptance of all patent applications of the Patent Act of 1793. The patent law of 1836 is still in effect. The examination of patent applications was introduced again. A patent office was now responsible for the examination and the granting of patents. The patent applicant now also had to include the *claims* of his invention in order to determine the scope of the patent. The patent term of 14 years could now be extended to up to 21 years upon application.

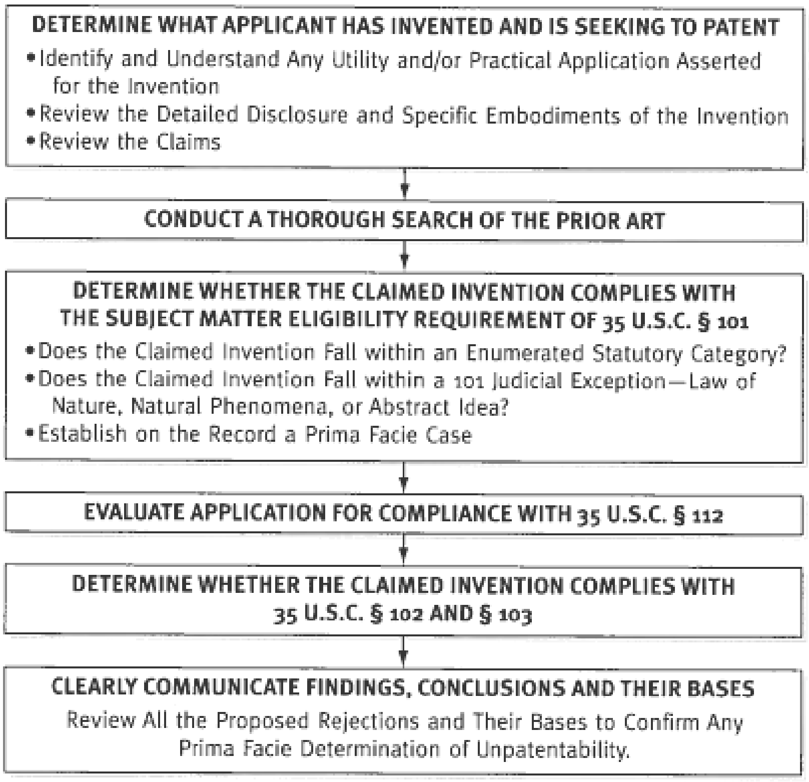
2.3 The Patent Examination Process in the United States

The patent examination process (starting with the filing of a non-provisional patent application, ending with either the issuance of a patent or the abandonment of the

[4] It is interesting to see how Thomas Jefferson's opinion on patents changed over just a few years. While, in 1787, he still opposed patents, he would note three years later that "[the new patent law] has given a spring to invention beyond his conception." [Pat]

[5] A 1952 modification saw the replacement of the term *art* by the term *process*; otherwise, the language is unchanged to this day.

application; see Figure 2.1) is a complex and time-consuming[6] procedure.



DETERMINE WHAT APPLICANT HAS INVENTED AND IS SEEKING TO PATENT

- Identify and Understand Any Utility and/or Practical Application Asserted for the Invention
- Review the Detailed Disclosure and Specific Embodiments of the Invention
- Review the Claims

CONDUCT A THOROUGH SEARCH OF THE PRIOR ART

DETERMINE WHETHER THE CLAIMED INVENTION COMPLIES WITH THE SUBJECT MATTER ELIGIBILITY REQUIREMENT OF 35 U.S.C. § 101

- Does the Claimed Invention Fall within an Enumerated Statutory Category?
- Does the Claimed Invention Fall within a 101 Judicial Exception—Law of Nature, Natural Phenomena, or Abstract Idea?
- Establish on the Record a Prima Facie Case

EVALUATE APPLICATION FOR COMPLIANCE WITH 35 U.S.C. § 112

DETERMINE WHETHER THE CLAIMED INVENTION COMPLIES WITH 35 U.S.C. § 102 AND § 103

CLEARLY COMMUNICATE FINDINGS, CONCLUSIONS AND THEIR BASES

Review All the Proposed Rejections and Their Bases to Confirm Any Prima Facie Determination of Unpatentability.

Figure 2.1: The patent examination process. Source: [HNR07, p. 15]

2.3.1 Filing and Initial Examination

The Office of Initial Patent Examination (OIPE) is the first place where a patent application is processed after it has been filed (by hand, mail delivery or electronically) at the USPTO. The OIPE is responsible for reviewing the formalities of the patent application (eventually requesting complete information from the applicant in case of improperley prepared applications), for the assignment of an application number, for the classification of the invention; finally, the application is transmitted to the suitable Technology Center[7] (TC) (see Table 2.1), from where it is assigned to one patent examiner.

[6] As stated in [HNR07], the examination process generally takes between 2-5 years. The duration is in part related to the invention's technological field - biomedical, electrical and business methods generally take longer. As already mentioned, the backlog of pending patent applications is contributing negatively to the examination duration.

[7] See http://www.uspto.gov/about/contacts/phone_directory/pat_tech/

Table 2.1: Technology Centers at the USPTO. Each of the nine TC centers (printed in bold) is divided into smaller and more specific units.

Art Unit	
1600	**Biotechnology and Organic Chemistry**
1610	Organic Compounds: Bio-affecting, Body Treating, Drug Delivery, Steroids, Herbicides, Pesticides, Cosmetics, and Drugs
1620	Organic Chemistry
..	
1700	**Chemical and Materials Engineering**
1791	Tires, Adhersive Bonding, Glass/Paper making, Plastics Shaping & Molding
1792	Coating, Etching, Cleaning, Bonding, Single Crystal Growth
..	
2100	**Computer Architecture, Software, and Information Security**
2110	Computer Architecture
2120	Miscellaneous Computer Applications
..	
2400	**Computer Networks, Multiplex communication, Video Distribution, and Security**
2410/2460/2470	Multiplex; VoIP
2420	Cable, Television
..	
2600	**Communications**
..	
2800	**Semiconductors, Electrical and Optical Systems and Components**
..	
2900	**Designs**
..	
3600	**Transportation, Construction, Electronic Commerce, Agriculture, National Security and License & Review**
..	
3700	**Mechanical Engineering, Manufacturing, Products**
..	

Publication of Patent Applications

Each patent application filed on or after November 29, 2000 is to be published 18 months after the applications earliest priority date. However, applications that are no longer considered as pending by the USPTO or applications that are subject to a secrecy order are excepted from this rule. The applicant may request an earlier publication of the application.

Before November 29, 2000, patent applications were not automatically published.[8] The fact that patent applications were not automatically published contributed to the rise of so called "submarine" patents, i.e. patent applications that stayed secret for a long time, e.g. 10-20 years. Their examination process was often intentionally delayed by an applicant who would claim infringement if a similar invention was made during this period. [Bel13] Once a patent application is published it is available to the public, as are it's modifications. The patent application is now citable by patent examiners as prior art.

2.3.2 Patent examination process

Exactly *one* distinct invention must be claimed by a patent application. If an application claims more than one invention, the applicant is required to restrict the claims to a set of claims associated to a single invention.[9]

The most crucial task in patent examination is the evaluation of an application's claims against a likely huge amount of patent and nonpatent literature. The Information Disclosure Statement[10] states that all known prior art that is of interest to the patent examiner in his determination of the patentability of the invention has to be disclosed by the applicant. The claims are usually amended in an iterative process that takes place between patent applicant and patent examiner. The patent examiner issues Office Actions[11], summarizing the status (e.g. allowed, rejected, objected to, or subject to restriction) of all of the application's claims. For rejected claims a brief explanation of the cause of their rejection is given by the examiner. Furthermore, the patent examiner may have objections to the specification of the patent application or to drawings; he/she may acknowledge priority of a previous application.

If the claimed invention is patentable in accordance with the principles of novelty, usefulness and non-obviousness, a Notice of Allowance[12] is issued. If, however, the patent examiner comes to the conclusion that a continuation of the examination process is futile (i.e. the invention is simply not patentable), a *final* rejection is sent, to which

[8] Intellectual Property and Communications Omnibus Reform Act of 1999.

[9] See http://www.uspto.gov/web/offices/pac/mpep/s802.html#d0e98006 for the definition.

[10] http://www.uspto.gov/web/offices/pac/mpep/s609.html

[11] See http://www.uspto.gov/sites/default/files/web/offices/pac/dapp/opla/preognotice/fai_office_action_summary.pdf for an example of an Office Action Summary document.

[12] http://www.uspto.gov/web/offices/pac/mpep/s1303.html

the applicant may appeal.[13] A detailed patent prosecution flow chart (see Figure 2.1) is found in [HNR07, p. 14].

2.3.3 Granted Patent

The current patent term in the U.S. is 20 years from the date the application was filed in the U.S (see Table 2.2). This term may be extended by (at most) five years as a compensation for delays experienced during the patent examination process. After the patent has been granted, maintenance fees are due at 3.5, 7.5 and 11.5 years. Failing to pay the maintenance fees leads to an expiring of the patent, putting the claimed invention into public domain.

Table 2.2: Maximum patent term in the U.S.

Filed	**Maximum patent term**	
1790-1835	14 years from issuance	Patent Act of 1790
1836-1860	21 years from issuance	Patent Act of 1836
1861-1994	17 years from issuance	
1995	20 years from filing	Uruguay Round Agreements Act

However, it has to be noted that the grant of patent does not guarantee its absolute validity. The U.S. do not implement a public review period, during which other parties can argue against the validity of the granted patent.

Sections of a Patent

US patent documents contain the following different sections.

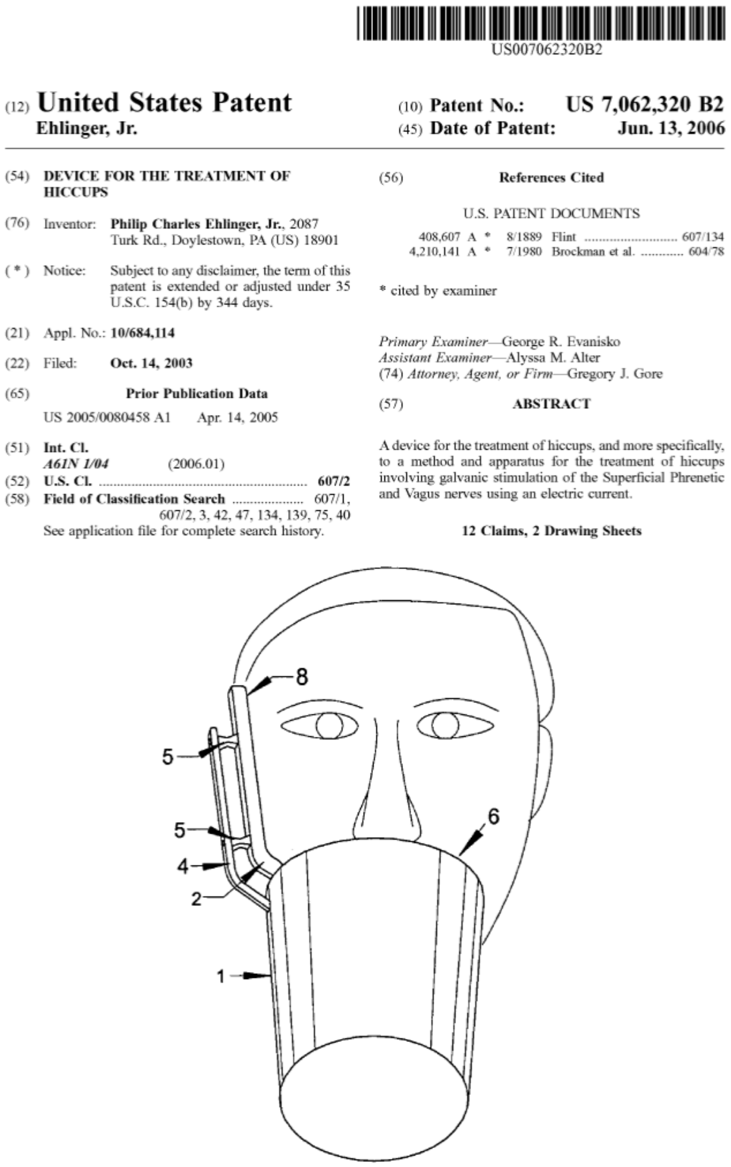
Front page The front page (or summary page; see Figure 2.2) contains bibliographic data of the invention, such as the patent number, the date when the patent was granted, the inventor's names, the application number, the filing date, cited references, the patent classification code and an abstract summarizing the general idea of the invention.

Drawings A set of drawings to illustrate an invention. Drawings are required if they are necessary for understanding an invention.

Background of the Invention This section provides context to the invention. It describes the problem to be solved and the current state of technology.

Brief summary A summary of the detailed description of the invention.

[13] In case of a denial of the appeal of the Final Rejection, the applicant may first appeal to the Board of Patent Appeals and Interferncces (BPAI). A subsequent denial may lead to an appeal at the Federal Circuit. Rarely a patent case is brought to the U.S. Supreme Court.



US007062320B2

(12) **United States Patent**
Ehlinger, Jr.

(10) **Patent No.: US 7,062,320 B2**
(45) **Date of Patent: Jun. 13, 2006**

(54) **DEVICE FOR THE TREATMENT OF HICCUPS**

(76) Inventor: **Philip Charles Ehlinger, Jr.**, 2087 Turk Rd., Doylestown, PA (US) 18901

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 344 days.

(21) Appl. No.: **10/684,114**

(22) Filed: **Oct. 14, 2003**

(65) **Prior Publication Data**

US 2005/0080458 A1 Apr. 14, 2005

(51) **Int. Cl.**
A61N 1/04 (2006.01)

(52) **U.S. Cl.** **607/2**

(58) **Field of Classification Search** 607/1, 607/2, 3, 42, 47, 134, 139, 75, 40
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

408,607 A * 8/1889 Flint 607/134
4,210,141 A * 7/1980 Brockman et al. 604/78

* cited by examiner

Primary Examiner—George R. Evanisko
Assistant Examiner—Alyssa M. Alter
(74) *Attorney, Agent, or Firm*—Gregory J. Gore

(57) **ABSTRACT**

A device for the treatment of hiccups, and more specifically, to a method and apparatus for the treatment of hiccups involving galvanic stimulation of the Superficial Phrenetic and Vagus nerves using an electric current.

12 Claims, 2 Drawing Sheets

Figure 2.2: Cover sheet of patent US 7062320 - a device for treating hiccups.

Detailed description The description of the invention must be complete and detailed enough to enable a person of ordinary skill within the technological field of the invention to make and use the invention.

Claims Contains the list of claims the inventor makes. This section is the most important section, as it defines the scope of the patent.

2.4 Patent Searching

Fast-paced technological progress as well as today's vast amount of technological knowledge have turned patent searching into a time-intensive and complex task. As already mentioned, it is also a task with high risks involved; missed prior art may lead to costly legal prosecution. There is strong evidence that bad patents, i.e. patents that have been granted incorrectly due to missed prior art, impose substantial cost on the society [FKS07].

Not only does patent searching demand a profound knowledge of the subject matter and it's technological domain, but also discipline with regard to *how* a search on prior art is planned and carried out.

2.4.1 Types of Patent Searches

Patent searches are conducted for different reasons. [HNR07] differentiate between 1. patentability, 2. validity, 3. infringement, 4. clearance, 5. state-of-the-art, and 6. patent landscape searches.

Patentability Search

The patentability search helps in assessing the novelty and non-obviousness aspects of the invention.

Relevant to this assessment is any written material (patent and non-patent literature, pending patent applications, conference proceedings, scientific papers...) that was published *anywhere* in the world before a date recognized as critical. The date recognized as critical is 1. the filing date of the patent application (if a patent application has been filed), or 2. the current date (if no patent application has been filed yet) .

Validity/Invalidity Search

Validity/invalidity searches are conducted after a patent or a patent application has been published. They determine the absolute novelty of an invention at the time the invention was made.

This type of patent search can be viewed as a more exhaustive patentability search. During initial patent examination critical existing prior art may have been overseen. The goal of this type of patent search is to find evidence that such prior art does (or does not) exist. This type of search is often conducted within the context of potential patent infringements or business opportunities. Validity/invalidity searches are conducted over patent and non-patent literature published before the earliest claimed priority date of the patent in question. The subject of the search are the claims of the patent or patent application. The search includes full patent specifications, claims of global patents filed on or before that date as well as any technical or non-technical literature published on or before that date.

Invalidity Search Company A is sued by Company B over an alleged patent infringement. Company A performs an invalidity search on company B's patent over which A is sued. Company A finds evidence that the company's B patent is invalid and claims that company B's patent is invalid and thus unenforceable.

Validity Search Company B plans to sue Company A over patent infringement. Company B performs a validity search to claim that their patent is valid. Since litigation is costly this provides them reasonable assurance of a successful outcome. A licensee may conduct a validity search to ensure that the proposed royalty payments are justified. A licensor (patent owner) conducts validity search as a highly defensible patent will command greater royalties.

Infringement Search

An infringement search is usually conducted *before* a product or service is made, used or sold.

The focus of the search are claims of patents and patent applications that are currently in force. The target is to locate enforceable patents with claims that might eventually interfere with the service or product. Neither expired patents nor non-patent literature is considered.

Clearance Search

A clearance search is similar to an infringement search; however, a clearance search also tries to determine where an inventive concept *could be* protected and used. The search is not limited to certain countries. Non-patent literature is searched as well. A clearance search might reveal unexploited markets around the world.

State-of-the-Art Search

While all types of searches mentioned above are related to *one* specific invention or inventive concept, the result of a state-of-the-art search is meant to reflect the current state-of-the-art in a technical field.

Consequently the search includes all available patent and non-patent literature. The state-of-the-art search is a valuable tool in a company's assessment of it's strategic decisions - including the direction of research activities, the evaluation of it's patent portfolio, marketing considerations, licensing opportunities and strategic business acquisitions.

Patent Landscape Search

A patent landscape search is helpful in visualizing the historical development of a technological field (groundbreaking patents, patenting activity over certain periods of time) and identifying current and future competitors.

2.4.2 Scoping a Patent Search

It seems superfluous to stress the importance of preparing a patent search thoroughly, e.g. by scoping the subject matter, identifying the patent classification areas, etc. Approaching a patent search systematically is an absolute necessity. An effective starting point is to identify and gain information on the *problem* an invention solves, on what an invention (physically) *is* and on what an invention *does*. Based on these answers, keywords for performing a text search can be generated. Patent searching is "a learning process in and of itself" [HNR07]. With each iteration of the search the patent searcher may discover new aspects of the invention as well as different keywords and terms; hence his/her findings will influence subsequent iterations. Full text search is a game of narrowing and broadening [HNR07].

The arguably most popular way to perform patent searches today is text based, i.e. the patent searcher issues queries into a patent search engine[14]. Other ways to perform patent searches are based on following the citations of a patent document or on the classification of the invention. Both methods are discussed in the sections below.

To define which keywords a patent document must or must not contain the searcher provides operators (see Table 2.3) in his search queries.

What problem does the invention solve? A description of the problem the invention is trying to solve. Based on the given description a list of relevant keywords is generated. The following example is derived from patent US 7062320 (Figure 2.2).

- Problem: Chronic Singultus, also known as Synchronous Diaphragmatic Flutter or hiccup, is an irritating phenomenon with possibly awkward consequences to the social and private lives of the sufferers.
- Keywords: Synchronous Diaphragmatic Flutter, Singultus, hiccup, spasmodic hiccup, chronic, intractable, persistent...
- Query: (singultus OR hiccup OR synchronous diaphragmatic flutter) AND (chronic OR intractable OR persistent)

What is the invention? Either a physical description of the apparatus or, in case of the inventive concept being a method, a description of the process and it's steps.

[14] A comparison of different free and commercial patent search engines can be found at `http://www.intellogist.com/wiki/Compare:Patent_Search_System`

Table 2.3: Boolean operators.

Operator	Example use	Included documents
OR	termA OR termB	All documents that contain *termA*, *termB*, or both.
AND	termA AND termB	All documents that contain *termA* and *termB* in any order.
NOT	termA NOT termB	All documents, that contain *termA* but not *termB*.
XOR	termA XOR termB	All documents that contain *termA* and not *termB*, and all documents that contain *termB* and not *termA*.
ADJ	termA ADJ termB	All documents, where *termB* is adjacent to *termA*.
NEAR	termA NEAR termB	All documents that contain *termA* AND *termB* within a range of *10* words.
NEAR/n	termA NEARn termB	All documents that contain *termA* and *termB* within a range of *n* words.
SAME	termA SAME termB	All documents, where *termA* and *termB* occur in the same paragraph.
SAME/n	termA SAMEn termB	All documents, where *termA* and *termB* occur within a range of *n* paragraphs.
WITH	termA WITH termB	All documents, where *termA* and *termB* occur in the same sentence.

- Description: A device that is composed of a vessel and electrodes...
- Keywords: vessel, container, bowl, canister, electrode, plate, wire, cathode, anode, cable...
- Query: (vessel OR container OR bowl) AND (elector OR plate OR cathode)

What does the invention do? The invention stimulates the superficially coursing vagus and phrenic nerves in order to reliably interrupt the Hiccup Reflexive Arc.

- Keywords: stimulate, interrupt, block, prevent, ...
- Query: (stimulate OR interrupt OR block OR prevent)

Usually an invention solves exactly *one* problem. A comprehensive search will lead to the discovery of documents with several solutions to the same problem, and similar inventions for completely different problems. The question is to discover all possible alternate solutions to the same problem as well as all alternate problems with the same solution.

In [HNR07, p. 68], applying systematic text query progression to a text search is recommended:

Structure or function The starting point of the search is the generic structure or function of the invention. Text queries to include the problem the invention solves are applied gradually. Alternative structural or functional solutions to the same problem will not be detected.

Problem The starting point of the search is the problem the invention solves. Text queries to include the structure or function of the invention are added gradually. Alternative problems to the same structural or functional solution will not be detected.

Structure, function and problem Structure and function of the invention as well as the problem are initially combined. Groupings of keywords are removed gradually from the initial query. Redundancy between the results of the queries has to be considered.

Citation Search

A citation search discloses how an invention was interpreted by other searchers, i.e. which technologies were considered relevant. Conducting a citation search is useful if a searcher is not sure about where to start his/her search. By following the citations, the evolution of an invention is gradually laid open. Backward citations refer to patent documents and publications that are cited as reference on the subject patent; they were generated during patent examination and can can be used to discover the first fundamental discovery of a technology. Forward citations are patent documents and publications that are subsequently citing the subject patent.

Classification Search

Patents are classified into a hierarchical system of symbols based on their technological field. Different classification systems exist, such as the US Patent Classification System[15] (USPC); the Derwent classification system[16] (DWPI); the Cooperative Patent Classifica-

[15] http://www.uspto.gov/web/patents/classification/uspcindex/indextouspc.htm

[16] http://ip-science.thomsonreuters.com/support/patents/dwpiref/reftools/classification/

tion[17] system (CPC), a joint effort between USPTO and EPO, and the International Patent Classification system[18] (IPC), established in 1971.

Patents that have been misclassified (so called "outliers") and thus cannot be obtained via classification search can be found via text search. Any data field of the patent or patent application (inventor name, patent owner, filing date...) can be searched for. In contrast to a citation search the text search requires the searcher to already know the terminology of the underlying invention and the technological field.

2.5 Summary

In this chapter, first the idea behind and the historical development of today's patent system is briefly presented. It is shown that the *core idea* of patents, the protection of an invention for a limited time, can be traced back at least 2500 years, with the first formal patent system being instantiated around 500 years ago.

Also, the mechanics of searching (for relevant patent literature) and the different types of patent searches (each with a different purpose) are discussed. Patent searching is a complex task with high risks involved. Therefore, the importance of conducting well-versed patent searches can not be overstated.

[17] http://www.cooperativepatentclassification.org/

[18] http://www.wipo.int/classifications/ipc/en/

CHAPTER 3

Query Expansion

Query Expansion (QE) aims at improving search results by adding relevant terms to a query provided by the searcher. This chapter presents popular QE strategies.

3.1 Introduction

A search query provided by a standard user is often just a vague description, an overly simplistic model of what the user is searching for. Searchers in general often only provide very brief queries with two or three keywords [Kak12]. In [ZTL00] the average query in a set of 50,538,653 queries gathered from the WebCrawler search engine was found to contain 3.3 terms; more than 25.5% of all web queries contained only a single term. The user might be inexperienced concerning the use of search engines in general, choosing only a very limited set of keywords or disregarding naming variations. This circumstance constitutes a fundamental challenge in the field of Information Retrieval (IR). A more experienced searcher will eventually apply QE intuitively, simply by adding synonyms and related words to his/her query or by using functions a search engine may implement. [Miz98] uses the terms Real Information Need (RIN) and Perceived Information Need (PIN; a *user-perceived* representation of the RIN) to describe the user's "problematic situation" [OBB82]. The (eventually incorrect and likely incomplete) Perceived Information Need is used by the user to formulate his/her search query.

Query Expansion has been studied for more than 50 years [CR12]. The general idea behind it is to augment and/or reformulate a user query with similar terms and phrases in order to achieve a better match between a user's information need and the retrieved documents [Kak12]. To achieve this, many different techniques and strategies have been suggested. This chapter provides an overview of popular QE approaches.

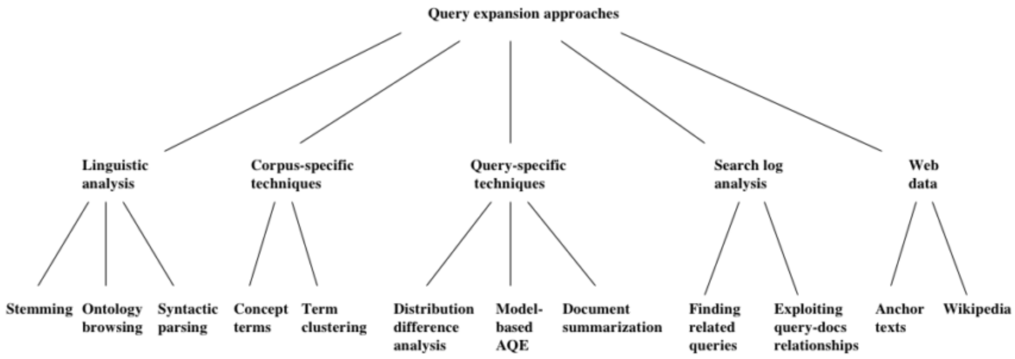


Figure 3.1: Approaches to Query Expansion. Source: [CR12]

In the field of professional patent searching the use of manual QE is common (see Chapter 4), either by the use of truncation operators (to consider variants of a word) or by providing a list of alternate terms. Still, professional patent examiners might benefit from a more automatic approach provided by the search engine. For example, a user might not be too familiar with his/her field of interest (e.g. an inventive concept) at the time he/she starts his/her search. Consequently he/she will miss relevant keywords.

3.1.1 Approaches to Query Expansion

The arguably most common way to classify QE approaches is based on the source from which the expansion terms are drawn from. An excellent overview (see Figure 3.1) of Query Expansion approaches can be found in [CR12]. Here, Query Expansion approaches are classified as follows: a) approaches based on linguistic analysis; b) corpus-specific approaches; c) query-specific approaches; d) approaches based on search log analysis; and e) approaches exploiting web data, e.g. anchor texts of internet pages.

Similar classifications have been used in [GWR99], where QE approaches are classified into *query-specific*, *corpus-specific* and *language-specific* approaches; in [GW06] the terms *extensional* (i.e. Relevance Feedback), *intensional* (approaches that use a global resource like a thesaurus), and *collaborative* are used.

Corpus-specific approaches include the exploitation of statistical measures like term co-occurrence within a document collection to create a data structure (e.g. a thesaurus) that is subsequently used as source for expansion terms. A key issue in this family of techniques, as will be shown in Section 3.2, is to define *when* two terms are to be recognized as related.

Methods based on linguistic analysis usually also make use of dictionaries or thesauri. However, term similarity is not derived by measures like term co-occurrence. Instead,

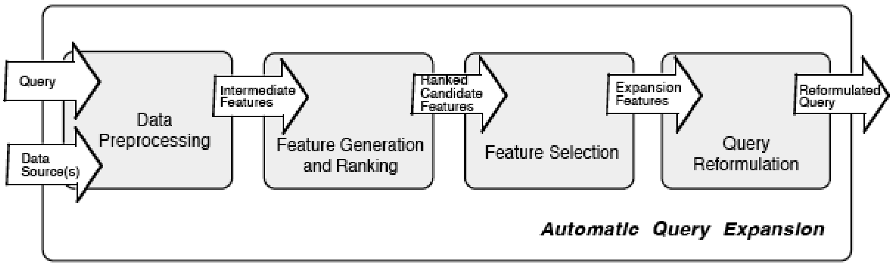


Figure 3.2: Steps in automatic query expansion. Source: [CR12]

morphological, lexical, syntactic and semantic word relationships are used. Approaches based on linguistic analysis are discussed in Section 3.3.

Query-specific approaches pursue a completely different strategy. Here, expansion terms are not directly generated from the user's initial query q. Instead, the set of initially retrieved documents in response to q is used as source for expansion terms (Relevance Feedback, see Section 3.4). The underlying idea is that this set of documents presents a more detailed representation of the user's information need than q itself. A key question in this family of approaches is how to determine which terms from a set of relevant documents characterize these documents best.

Collaborative approaches try to use information gained by the behavior of previous searchers in order to improve the understanding of the intended meaning of a query. Web data, e.g. anchor texts of web pages, are used by several more recent approaches as a source for the generation of expansions.

3.1.2 Applying Query Expansion

As shown in Figure 3.2, an IR system is usually concerned with four "stages" of the QE process: Preprocessing, Generation and ranking of expansion terms, Selection of expansion terms and Query reformulation. [CR12]

Preprocessing Preprocessing encompasses actions that are likely to ensure fast and easy data access in later steps. Common techniques at this stage include tokenization[1], stop word removal, word stemming and term weighting, or even the creation of a thesaurus derived from the text collection at hand. Stop words[2] are words that are too common within the collection to serve as useful discriminators, for example articles, prepositions or conjunctions. Word stemming is the process of finding

[1] Tokenization is the process of splitting text into words, phrases, symbols and other elements.

[2] The term "stop word" was coined by Hans Peter Luhn. A popular list of stop words, still used in software packages such as the Natural Language Toolkit (NLTK) was compiled by Martin Porter in 1980.

the root form (e.g. "automa" is a stem for the words "automation", "automatic", "automating"...) of a word.[3] Term weighting is the process of attaching *weights* to terms in order to reflect their importance in a collection of documents.

Generation and ranking of expansion terms At this stage, triggered by the submission of a user query, candidate terms for query expansions are generated and ranked based on their merit for expansion. Since it is possible that not all terms are used for expansion the ranking information helps in selecting the most important ones. A multitude of expansion term generation strategies have been proposed, which will be discussed in the following sections.

Selection of expansion features At the previous stage a pool of candidate expansion terms has been derived. In this stage, terms finally used for query expansion are selected from the pool of candidate expansion terms. It is difficult to determine an optimal number of expansion terms. On the one hand there is evidence that contradicts the notion of simply adding *all* candidate expansion terms to the query; even the addition of *many* expansion terms may be harmful to retrieval performance, and the performance improvement would be much higher if only the best expansion terms could be selected [Cao+08]. On the other hand [Won+08] and [Ber+08] did not observe a negative impact caused by the massive (i.e. a few hundred) addition of query terms. [Ber+08] states performance of relevance feedback in general increased as the number of feedback documents and the number of expansion terms grew.

Query reformulation The final step is to reformulate the initial query, i.e. adding additional terms to the query. A *weight* can be attached to query terms to further emphasize or deemphasize their importance.

3.2 Corpus-specific global techniques

Common to this family of techniques is the use of a "global" knowledge resource, e.g. a thesaurus. A thesaurus consists of a set of classes, where each class contains a set of closely related[4] terms, sorted based on their similarity of meanings.

As already mentioned, a thesaurus can either be manually created by human editors, which is a potentially expensive process, or automatically derived by parsing the whole text corpus. The latter approach tries to identify terms that are used in similar ways, for example by analyzing their co-occurrence: frequently co-occurring terms are more likely to be related. One of the earliest publications "concerned with the recognition and exploitation of term associations for the retrieval of documents" is that of [GJ62].

Many techniques based on the exploitation of co-occurrence data have been judged as not successful. For example, [PW91] state that "despite the plausibility of this approach to query expansion, the retrieval effectiveness of the expanded queries is often no greater

[3] A standard algorithm for word stemming was presented by Martin Porter in 1980.

[4] That is, closely related in the context of the collection.

than, or even less than, the effectiveness of the unexpanded queries", and conclude that "the weight of the experimental evidence to date hence suggests that query expansion based on term cooccurrence data is unlikely to bring about substantial improvements in the performance of document retrieval systems."

3.2.1 Concept terms

[Qiu+93] mention the two key issues for QE: 1. *selection* of suitable terms, and the 2. *weighting* of selected additional search terms. A naive term-for-term translation, they believe, will not yield favorable results. Therefore the similarity of a term to the *query concept* should be considered, whereas many earlier approaches have treated terms as rather isolated entities. In most cases, potential word ambiguities are resolved by the context. The context is lost when single terms are expanded without considering their query context.

Similarity thesaurus

Terms are selected based on their similarity to the *concept* of the query, and not based on the relation between a single query term and terms of the document collection. In [Qiu+93] a similarity thesaurus, consisting of term-term similarities, is created. The idea behind a similarity thesaurus is that each term in the text collection can be characterized by the documents it appears in.[5]

A term t is indexed by the documents d of the collection:

$$\vec{t_i} = (d_{i1}, d_{i2}, .., d_{in})^T, \tag{3.1}$$

where n is the number of documents in the collection and d_{i1},...,d_{in} are feature weights with respect to term t_i. For all possible term pairs (t_i, t_j) a similarity score $0 \leq SIM(t_i, t_j) \leq 1$ is obtained by applying a similarity measure. In [Qiu+93], the standard scalar product is used. The query is then expanded by selecting the nearest terms for the centroid of the query vector. A more recent effort based on the use of a similarity thesaurus is described in [Zaz+05].

Noun group

In [XC96] the term *concept* is defined as a noun group. A noun group ("phrase") is either a single noun or two or three adjacent nouns. The *context* is defined as the collection of fixed length *windows* that are surrounding the concepts; effective window sizes were found being 1 - 3 sentences long. To every concept a pseudo-document can be associated, that contains the words that occur in every window for that concept in the corpus. For example, the concept *object-oriented* might have the words programming, inheritance, interface and Java occurring frequently in the corresponding pseudo-documents. A concept database is created by all pseudo-documents. The user query is then expanded by the top entries from a ranked list (retrieved via INQUERY ([CCH92]) of *phrasal concepts*. The retrieval performance of this method increased the baseline on TREC3 by 7.8%, on TREC4 by 3.4%.

Noun phrase

[5] As opposed to the more "traditional" way of characterizing a collection's documents by their terms.

In [Liu+04] a new approach for processing noun phrases in documents is proposed. Here, phrases are considered to be more important than individual terms. Therefore, the similarity measure consists of two components, 1. phrase-similarity, and 2. term-similarity. Documents are first ranked by phrase-similarity. If the phrase similarity is identical, documents are ranked according to their term-similarity. Four types of phrases with learned window sizes, i.e. all the words in the phrase are within a window of a given size, are identified in queries: a) Proper names (i.e. names of people, places, organizations, detected by the named entity recognizer and a window size of 0, i.e. adjacent words); b) dictionary phrases (i.e. phrases defined in dictionaries such as WordNet and window size of 15); c) simple phrases (i.e. two to four words in length, no embedded noun phrase, e.g. "school uniform" and a window size of 50); and d) complex phrases (one or more dictionary phrases and/or simple phrases,... "instruments to forecast weather" and window size of 80). In this approach significant phrases are identified first. Proper names and dictionary phrases are assumed to be significant. A simple or complex phrase is significant if the content words within the phrase are highly positively correlated in the collection of documents.

3.2.2 Term clustering

Complete link clustering

In [CY92], a thesaurus based on the result of clustering the document collection using the complete link clustering algorithm is used for QE. The construction of the thesaurus is discussed in detail in [Cro88]. In this approach the document collection is a) clustered via the mentioned complete link clustering algorithm; b) then, the result of complete link clustering algorithm is traversed to generate thesaurus classes; and c) documents and search queries are then augmented by the thesaurus classes.The results of experiments conducted on four standard test collections showed that this approach of thesaurus generation was promising and the retrieval effectiveness was substantially improved. [CY92]

Lexical-co-occurrence

In [SP97], the construction of a lexical-co-occurrence-based thesaurus is described. The pattern of local co-occurrences of each term is represented by a vector; the term similarity (i.e. co-occurrence similarity) can then be measured by comparing the term vectors. Furthermore, the use of thesauri to cluster query terms is suggested with promising results.

Spectral retrieval (curves of relatedness scores)

In [BMW07], clusters of terms are derived with two different approaches. In a supervised approach two terms were put in the same cluster if they both shared the same and most frequent WordNet SynSet. In an unsupervised approach a set of related term pairs is extracted from a collection using a method described in [BM05]. Finally, clusters are derived from the term pairs using a Markov Clustering Algorithm (MCL).

3.2.3 Context vectors

In an approach suggested by [GWR99], words are considered similar if they are used in similar *contexts*. They do not need to co-occur in the same document. For example certain spelling variations ("color", "colour") are not likely to occur in the same document. The highest frequency words are used as context words, as they provide more statistical information in smaller samples. The similarity calculation is performed as follows. First, target words are identified based on their frequency for which pairwise similarities are going to be calculated. For each target word a context vector is generated. The context vector contains information about word occurrences around the target word. After the context vectors for each target word are created, the sum of word occurrences are replaced by mutual information values [CH89]. The mutual information is large whenever a context word appears at a much higher frequency f_{cw} in the neighborhood of a target word than would be predicted from the overall frequencies in the corpus of the context word and target word.

Adding 1. all highly similar words, and 2. a fixed, small number of somewhat familiar words in a "two-tiered" approached proved to be successful especially on a smaller, specialized corpus.

3.2.4 Mutual information

An approach described by [HDG06] extracts informative terms that are most associated to a given query to perform QE. Here, association is not necessarily meant in the grammatical sense of the word. For example, an original query "object-oriented" may be expanded by "inheritance", "Java", "programming language" or "paradigm". A three stage approach is proposed:

Term-term association calculation The statistical relationship between term pairs in the document collection is used for the construction of a thesaurus-like resource. The association A of two terms $t1$ and $t2$ is calculated by integrating three factors: term weight w, the mutual information value I, and normalized distance D between $t1$ and $t2$. The mutual information value is a measurement of the average amount of information shared between two variables. In this case the measurement is achieved by comparing the probability of two terms co-occuring in a document with the two terms occuring independently.

The addition of the distance D is based on the assumption that two terms that appear closer to each other in a document are also more closely associated.

$$A(t_i, t_j) = \frac{(w_i) * I(t_i, t_j)}{D(t_i, t_j)} \tag{3.2}$$

Term-query based expansion Similar to the family of approaches described in section 3.2.1 a term has to be related to the query to be considered for expansion. Accordingly a *term-query* based expansion scheme is proposed to emphasize the correlation of a term to an *entire* query. The result of the calculation is a ranking of all index terms based on their correlation to the query. Top m ranked terms are then used for expansion.

Re-weighting of expansion-terms Finally, expanded terms are re-weighted based on a simple re-weighting scheme. The motive is to attenuate a possibly too strong influence of the expanded terms to the original query, which might lead to query drift. On the other hand, it should also be ensured that the expanded terms have more than a negligible effect.

3.2.5 Latent semantic indexing

Many traditional approaches rely on the presence of *terms* to determine the relevance of a document. This strategy can in a way be considered naive, as the same idea can be projected by using different terms. Therefore, latent semantic analysis aims at discovering the topics within a document. A topic is defined as a set of related words. In an approach described in [PR07] a thesaurus is constructed by applying probabilistic latent semantic analysis. In a first step the probabilistic relationships between the topics and terms, and the topics and documents are obtained. Then, the probabilistic relationships between each of the terms are calculated in order to construct the thesaurus.

3.3 Approaches based on Linguistic-analysis

In this family of methods query terms are expanded based on lexical, morphological, syntactic and semantic word relationships. Usually, a global resource like a thesaurus or a dictionary is used.

3.3.1 Word stemming

Word stemming is the process of reducing different morphological forms of a word to a root form. The arguably most popular stemming algorithms are presented in [Lov68] and [Por80]. The reasoning behind performing word stemming in the field of QE is quite obvious. Often, the similarity between two documents is simply calculated by the number of terms they share. Different morphological forms of words (e.g. "going" and "gone") would not match; reducing them to a common root ("go"), on the other hand, would. Naturally there are some serious drawbacks to this technique. As pointed out in [Hul96], words with different meanings can be stemmed to the same root, whereas words with the same meaning sometimes can not. For example, the terms "general", "generous", "generic" and "generation" are all reduced to the word stem "gener" by the Porter stemmer [Hul96].[6]

An overview of different types of stemming algorithms is presented in [Jiv11] (see Figure 3.3).

While [Har91] found that the the use of general stemmers did not improve retrieval performance, [Kro93] concluded that stemming lead to significant retrieval performance

[6] http://text-processing.com/demo/stem/

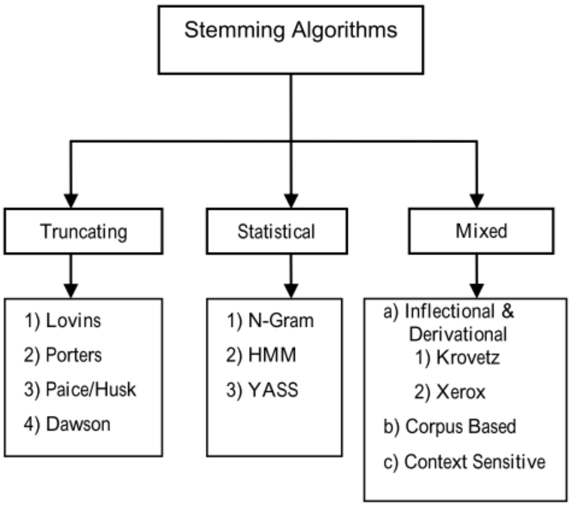


Figure 3.3: Types of stemming algorithms [Jiv11]

improvements. According to [Hul96], the use of "some form of stemming is almost always beneficial".

3.3.2 Ontologies

Ontology[7] can refer to a philosophical discipline, as "the science of what is, of the kinds and structures of objects, properties, events, processes and relations in every area of reality". [Smi08]. In Computer Science an ontology is a formal model of the structure of a system, organizing relevant entities into concepts and relations into a taxonomy of generalization/specialization [GOS03]. Ontologies can be domain-dependent or domain-independent.

Ontologies have been used for QE since around two decades. An exhaustive overview on QE approaches that make use of ontologies is presented in [BMS07].

WordNet

WordNet[8], a popular domain-independent ontology, is a lexical database that groups nouns, verbs, adjectives and adverbs into SynSets, i.e. sets of synonyms with the same meaning, expressing a distinct concept. In the following, a few selected QE approaches using WordNet are presented.

In [NV02], it is argued that the positive effect of expanding a query with synonyms and hypernyms is limited. Instead, the query should be extended with semantic information. This thesis is supported by the notion that most successful QE expansion methods

[7] While the underlying idea reaches back to Aristotle, the term ontology ("ontologia") was only coined in the early 17th century. [Smi08]

[8] https://wordnet.princeton.edu/

add terms based on co-occurrence[9] data. Instead of using co-occurrence data, semantic information can be extracted from an ontology. Several sense-based expansion methods of different complexity were evaluated in [NV02], among them the replacement of terms by their SynSet (based on a Word Sense Disambiguation algorithm discussed in detail in [NV02]), the expansion of terms with direct hyperonyms[10] retrieved from WordNet and the expansion of terms by the SynSets of their WordNet concept definitions (*glosses*). All of these methods were judged as being useful for QE in principle. In accordance with the suggestion that QE methods where terms are extracted based on semantic relationships are most successful, the test results implicated that words from the same semantic domain seem to be the best candidates for expansion.

In [BBAG05], the *document semantic core*, which is a representation of a document's content, is created from a given document. In the first step concepts are extracted from a document. Since each concept can have several meanings, the "best" concept senses are selected in a second step. The document semantic cores are then used to realize a "conceptual indexing" of the collection. It has been shown that retrieval accuracy is improved when conceptual indexing is used in conjunction with traditional keyword indexing.

In [GCH05], an association mining algorithm described in [AS94] to extract a Term Semantic Network (TSN) from the text collection is used. First, a query is expanded via WordNet using hypernym, hyponymy and synonym relations. The information obtained from the TSN is then used by a QE subsystem that performs 1. keyword expansion (add keywords with high confidence and support that were not described in WordNet) 2. keyword filtering (remove keywords with low confidence and support) 3. keyword weighting (e.g. different weighting applied for hypernyms, hyponyms, Synonyms and the Local Context).

Experiments showed that a) the precision improvement of a pure WordNet expanded query is limited; b) WordNet expansion with TSN filtering is much better than both the original query and the pure WordNet expansion; c) TSN expansion's recall and precision are higher than original query and WordNet expanded query, but lower than WordNet-TSN-Filtering expansion; and d) combined queries (WordNet-TSN-Filtering with TSN expansion) provided the best results.

Domain-specific ontologies

A problematic aspect of domain-independent ontologies such as WordNet is caused by their broad nature: term ambiguity. A domain-specific ontology (e.g. for medicine, law, etc.) is likely to extenuate this problem and can lead to increased retrieval performance.

In [FJA05], QE techniques are presented that are based on the use of a domain and a geographical ontology to deal with queries that involve spatial terms (e.g. *castles near Edinburgh*); Nilsson, Hjelm, and Oxhammar use a domain-specific ontology from the Stockholm university domain to perform QE within a cross-language QA system. [NHO05]. Aronson and Rindflesch compared the use of the Unified Medical Language

[9] Terms that are interpreted as pertaining to the same semantic domain.

[10] E.g. the term "vehicle" is a hyperonym for "car", "truck", etc.

System (UMLS) Metathesaurus[11] for QE to a QE approach known as local feedback (where salient terms from initially top-ranked documents are selected as expansion features), concluding that an optimal strategy would be a combination of both techniques. [AR97]. Hersh, Price, and Donohoe observed that the average retrieval performance was not improved by adding synonymous, hierarchical or related terms from the UMLS Metathesaurus unconstrained. [HPD00].

An approach by Jalali and Borujerdi applies term expansion using the Medical Subject Headings (MeSH) ontology[12]. [JB08]. First, the concepts within a user query are identified. To achieve this, noun phrases are extracted from the query and searched for in the MeSH ontology. Concepts with specific meaning are preferred. The query is then expanded by synonyms of identified concepts or by direct descendants of these concepts. Experimental results showed that the average precision was superior compared to the use of a general ontology (WordNet).

Houston, Chen, and Schatz compared three different thesauri (MeSH, UMLS and a thesaurus built from the CANCERLIT document collection) for performing QE in the medical domain. No statistically significant differences in terms of recall or precision were observed between these three thesauri, while the number of overlapping relevant terms was low, suggesting that an approach using a combination of thesauri could be beneficial. [HCS00].

Leroy and Chen conclude that the UMLS Metathesaurus is a useful tool in providing synonyms. On the other hand, WordNet cannot be used "to help bridge the gap from general English (such as patients would use) to specific medical terminology". [LC01b].

3.4 Relevance Feedback

The core idea behind this family of techniques is that the set of documents retrieved from an initial query provides a more detailed description of the query itself. Expansion terms are not selected based on their relationship to one or more query terms. Instead, the most important terms from the set of relevant documents retrieved are selected, i.e. terms that characterize the relevant documents best.

Rocchio is credited with the first formal description of a Relevance Feedback (RF) system [Roc71; RL03]. While it seems that there is no clear answer on the effectiveness of thesaurus based QE approaches [YC11], Relevance Feedback, on the other hand, has been described as one of the most used and most successful approaches used in IR. [Lar09] The relevance of a document for a search can be determined 1. manually, i.e the user has to examine each document and mark it as relevant (positive feedback) or non-relevant (negative feedback); 2. automatically by the IR system (by classifying the top k results as relevant); and 3. implicit via the user's behavior, e.g. by measuring which documents are viewed for how long, etc) [Kak12].

[11] https://www.nlm.nih.gov/research/umls/

[12] https://www.nlm.nih.gov/mesh/

The second approach is also known as Pseudo Relevance Feedback (PRF) or blind feedback. Here, the user's intervention is not required: his/her interaction is replaced with the assumption that the top k ranked documents are relevant. [Lar09] This approach performs well if the top k documents are actually relevant. It, however, is obvious that a poor result of the top k ranked documents will eventually even decrease the expanded query's performance since query terms are added from non-relevant documents. Topic drift is a problematic issue. For example, the top k ranked result documents for an initial search for "earthquake" may all concern the 2011 earthquake in Japan. Terms deemed as important in those documents may be "tsunami", "nuclear", "Fukushima Daiichi", etc., leading the search into a direction that was by not intended by the user. Applying Relevance Feedback is comprised of two main steps: a) adding new terms selected from pseudo-relevant documents, i.e. query expansion; and b) term re-weighting.

3.4.1 Classic approaches

Vector Space Model

The Vector Space Model (VSM) represents documents d and queries q as vectors of n weights w, n being the number of unique terms in the document collection:

$$\begin{aligned} \vec{d} &= (w_1, w_2, ..., w_n) \\ \vec{q} &= (w_1, w_2, ..., w_n) \end{aligned} \tag{3.3}$$

The weight indicates the importance of a term within the document collection.

Rocchio has defined the problem of retrieval as that of defining an optimal query [Roc71], i.e. a query that "maximizes the difference between the average vectors of the set of relevant documents and the set of non-relevant documents" [RL03]. It is obvious that submitting an optimal query is a quite unlikely task to achieve for the standard user. At this point, Relevance Feedback steps in by attempting to move the query vector closer to the mean of relevant documents. This can be achieved by adding and/or re-weighting the query's terms.

Rocchio's formula [Lar09, p.189]) moves the user provided initial query $\vec{q_0}$ closer to or away from the set of relevant and non-relevant documents. The algorithm adds or modifies weights of the inital query vector $\vec{q_0}$ to retrieve a modified vector $\vec{q_m}$. D_r is the set of relevant documents, D_nr is the set of non-relevant documents; α, β and γ are additional weights, emphasizing terms from relevant, and deemphasizing terms of non-relevant documents:

$$\vec{q_m} = \alpha\vec{q_0} + \beta\frac{1}{|D_r|}\sum_{\vec{d_j}\in D_r} - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d_j}\in D_{nr}}\vec{d_j}, \tag{3.4}$$

Term weighting is a critical issue in IR. Much of the success or failure of an IR system depends on it [Pol04]. Many previous studies have shown that term weights are key

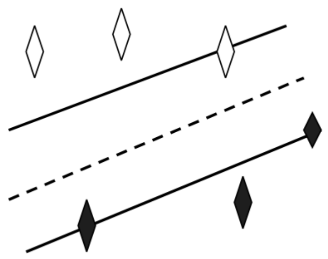


Figure 3.4: The black and white diamonds represent relevant and non relevant vectors. The solid lines are hyperplanes, defined by *support vectors*. One hyperplane is going through at least one relevant vector, another hyperplane is going trough at least one non-relevant vector. A SVM is maximizing the margin between the two hyperplanes.

factors for the performance of IR systems. The importance of a term within a document collection is "measured" by integrating three factors: 1. the term frequency factor tf, counting the number of times a term appears in a document; 2. the inverse document frequency *idf*, calculating a term's importance within a document collection; and 3. a normalization factor.

The idea behind the TF.IDF [SB88] weighting scheme is that frequent terms are less informative than infrequent terms. One problem is that tf will be higher for longer documents. The consequence is that the terms in longer documents will contribute more strongly to the document-query similarity. The use of a *normalization factor* such as Cosine normalization or Pivoted unique normalization [SBM11] attenuates this effect.

SVM as classifier

Drucker, Shahrary, and Gibbon compare a Relevance Feedback system based on the use of a Support Vector Machine (SVM) classifier to traditional approaches such as Rocchio [DSG02]. The SVM model is trained by the user's relevance feedback.

Drucker, Shahrary, and Gibbon state, based on experiments, that an SVM based implementation performed superior to traditional approaches when the initial search result was poor and the topic visibility low.

Another interesting approach based on the use of a SVM classifier is presented in [Xu+03]. They argue that traditionally users label only the most relevant documents and state that this information is not actually informative for the learning system. This is their objection to the Support Vector Model presented in [DSG02]. The use of active learning algorithms such as *SimpleMargin* ([SC00; TK01]) has been proposed. The SVM Active Learning for Relevance Feedback (ActiveSVMRF) implements the *SimpleMargin* algorithm in order to achieve a fast learning rate. The drawback is that the user would be required to read and label many non-relevant documents; to overcome this, a hybrid approach is proposed in which a) a SVM model is learned from an initial query; b) the most likely and the most uncertain documents are presented to the user; and c) a new SVM model is trained based on this information.

Robustness

It has been indicated that local feedback techniques can actually hurt the retrieval performance ([XC96], [RSW99]). The issue of *query drift* has been addressed in many studies. [MSB98] argue that the main reason for query drift is the presence of many non-relevant documents in the top-ranked documents. Therefore, one should aim at improving the precision of top-ranked documents. By assigning a score with regard to the presence of strong additional relevance indicators, the initial retrieved top-ranked list is re-ranked. Experiments showed significant improvements in retrieval effectiveness. [MSB98]

Based on the quality of the initial search, Amati, Carpineto, and Romano propose the introduction of a decision method to control if QE is applied at all. [ACR04] The problem of predicting a poor-performing query is difficult and well known in the field of Query Expansion.[13] A similar approach is proposed in [CTZC04], termed *selective-query-expansion*.

In [Sak00], the concept of Flexible Pseudo Relevance Feedback is presented. They assume that the effect of Relevance Feedback depends on the complexity of the search. Therefore, an estimated optimal number of top-ranked documents as well as an estimated optimal number of expansion terms that is to be extracted from this set of documents are calculated based on the actual query. Experiments showed inconclusive results. Later, this approach was extended by the use of *Selective Sampling*, where documents retrieved in the initial ranked list can be ignored if they are too similar to other documents in that list, looking for non-redundant pseudo-relevant documents instead. [SMK05].

In [TZ06] the importance of *each* feedback document is considered. For some queries not enough relevant documents may be available in the document collection; RF will lead to query drift in these cases. Using external document collections such as Wikipedia in combination to those retrieved from local collection might be an interesting approach. [Voo06; XJW09]

[LZ10] propose a novel positional relevance model for Pseudo Relevance Feedback, where term position and term proximity are considered, i.e. terms in the document that are closer to the query terms receive more weight than those that are far away.

3.4.2 Query language modeling

A Language Model[14] is a probability distribution over the terms of a text. It expresses the likelihood for a given term (or sequence of terms) within a language. For example, a Language Model for a text collection dealing with software development is expected to assign higher probabilities to terms from this domain (interface, abstract, extends, Java...).

[13] Many papers (such as [CTC02], [Huf08]) deal with this topic.

[14] The development of Language Models can be traced back to Andrei Markov (1856-1922), who modeled letter sequences in Russian literature (Markov models) [BLN04].

Term	LM_1	LM_2
a	0.2	0.25
is	0.25	0.25
interface	0.15	0
abstract	0.09	0.08
include	0.05	0.075
programming	0.1	0.0001
language	0.08	0.18
writer	0.001	0.25
prose	0.01	0.1
century	0.0001	0.01

Table 3.1: Probabilities of terms within two different Language Models LM.

Language Models have been used for many natural language processing applications like optical character recognition, machine translation and part-of-speech tagging. Since the late 1990s Language Models have become a popular technique in the field of IR. An early approach was presented by [PC98] to good results. The underlying idea is to build a Language Model LM_d for every document d in the collection. At search time, the probability that LM_d would generate the query q is calculated and used for ranking:

$$score(q, d) = P(q|LM_d) \tag{3.5}$$

Two main problems are stated in [Lóp13]: a) the definition of a Language Model LM_d; and b) the estimation of LM_d based on the document contents. Different models have been used for defining Language Models, such as the original multiple Bernoulli Model (in [PC98]) or the Poisson model (in [MFZ07]). Another important question is the assignment of probabilities to unseen query terms, a problem that is tackled by the use of smoothing methods. In [ZL04] different smoothing techniques are evaluated.

Suppose the term sequence *a programming language is abstract*. Two probabilities P_1 and P_2 can be calculated with the respective term probabilities (see Table 3.1) for both Language Models:

$$\begin{aligned} P_1 &= (0.2 \cdot 0.1 \cdot 0.08 \cdot 0.25 \cdot 0.09) = 3.600000000000001e - 05 \\ P_2 &= (0.25 \cdot 0.0001 \cdot 0.18 \cdot 025 \cdot 0.08) = 7.5600000000000005e - 06 \end{aligned} \tag{3.6}$$

Word dependencies such as "programming language", "object oriented", "New York", "The Mathematical Association of America", etc. can be respected with the use of n-gram models, n being the number of dependent words. A 1-gram model is called unigram, a 2-gram model ("New York") is called bigram, a 3-gram model ("Wolfgang Amadeus

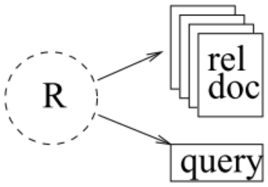


Figure 3.5: An assumed common Language Model R. Source: [LC01a]

Mozart") is called trigram, and so forth. A study presented in [MLS99] has shown small performance gains when using bigrams over unigrams in IR. Song and Croft conclude that word pairs are useful in improving the retrieval performance. [SC99]. However, unigram models seem to be the most widely used Language Models today. [Lar09, p.240]

Other early approaches include work done by Miller, Leek, and Schwartz, where two-state Hidden Markov Models are used ([MLS99]), and by Hiemstra; Berger and Lafferty; Song and Croft ([Hie98; BL99; SC99]).

Mixture Models

In [ZL01] relevance feedback documents are used to re-estimate the Language Model of the query. Two different estimation criteria are investigated: a) a mixture model, where the generative model is created from the query's topic model and the collection's language model; and b) the query model with the smallest average Kullback-Leibler divergence[15] to the feedback documents is selected. Both methods showed good results.

Relevance Models

A different strategy is applied in [LC01a]. Here, an *unknown* relevance model R is assumed, from which the user query Q and a set of relevant documents are considered samples (Figure 3.5). The probability of a word w in R is approximated by $P(w|Q)$, i.e. the probability of co-occurrence of w and Q.

In [MC07], a technique called latent concept expansion is presented, based on the Markov random field model. In contrast to [LC01a] proximity-based features are exploited, for example how often are query terms found in a window of fixed size either ordered or unordered.

Another more recent approach using sentence similarities is presented in [GLJ11]. They propose to add those sentences from the set of top ranked documents that have a maximum term overlap with the query.

Lv and Zhai have compared five different state-of-the-art methods for estimating query language models with pseudo relevance feedback. They conclude that the Mixture Model approach (described in [ZL01]) and a variation on the Relevance Model approach (described in [LC01a]) are most effective. [LZ09].

[15] Kullback-Leibler divergence measures the difference between two probability distributions.

3.5 Collaborative approaches and web data

The underlying idea is to use information gained from the searches of other users. In [CR12] two different approaches are identified. In the first approach, relevant terms are *not* selected from the retrieved documents. Instead, expansion terms are selected from similar *queries* that have been conducted by other users. The second approach selects relevant terms from the retrieved documents of similar past queries.

In [JRM06], the whole query and/or phrases of the query are substituted with terms or phrases retrieved from the search query logs of other users. First, a set of modified queries is generated for an initial query. For example, the initial query "catholic baby names" could be substituted with "catholic names" or with "baby names". Queries tend to consist of several concepts. Therefore, Jones, Rey, and Madani aim to split queries into segments ("catholic", "baby names"). For every segment candidate substitutions are then selected. Finally, a set of alternate queries is obtained, e.g.: "religious baby names", "catholic baby boy names", "catholic unique baby names", etc. [JRM06].

Huang, Chien, and Oyang use the contextual information embedded in the query *session*, i.e. previous search queries issued by the user during his current search. Relevant terms are selected from user queries that co-occur in similar *query sessions* from past searches of other users. [HCO03].

The core idea presented in [Cui+02] considers documents as relevant to a query if these documents have been clicked on in the result list by previous users after a search with *similar* queries. Based on the query logs a probabilistic correlation is established between query terms on the one hand, and document terms on the other hand. This can be exploited for selecting expansion terms from the documents. A drawback of this approach is the fact that documents are considered relevant as soon as the user has clicked on them. However, Cui et al. argue that they expect that most users click on relevant or seemingly relevant documents. This allows them to overcome the issue of lacking sufficient relevance feedbacks.

In [FD97], related terms are extracted from the documents retrieved by past *similar* queries. In [Bil+03], related terms are extracted from past queries associated to the top *k* retrieved documents.

Web data

Anchor texts are short descriptions of an internet destination page. It is assumed that anchor texts are similar to search queries and contain a concise description of the page. In [KZ04] an approach based on the use of anchor texts is presented.

3.6 Summary

In this chapter an overview of various QE strategies has been provided. Global techniques rely on a "global" resource (e.g. a thesaurus) from which the expansion terms are drawn from. The resource is usually automatically generated by determining e.g. frequent co-occurring terms. Approaches based on Linguistic-analysis exploit lexical, morphological, syntactic and semantic word relationships to identify potential expansion terms. The general idea behind relevance feedback is that retrieved (relevant) documents provide a more detailed description of the query. Thus, expansion terms are drawn from these documents. Collaborative approaches, finally, seek to use past similar queries as source from which additional terms are drawn.

CHAPTER 4

Data Acquisition and Preparation

This chapter describes the preparation of the dataset.

4.1 Introduction

Data retrieval and processing is handled by a collection of script files called PyPAIR[1] (see Figure 4.1). PyPAIR is able to perform bulk download of patent applications from the USPTO. SRNT and 892 documents are extracted from all downloaded patent applications. SRNT files are scanned documents. The content of SRNT documents has to be transformed into text. For OCR a 3rd party software is used. Badly processed documents have to be detected and removed. Acquisition and processing tasks are described in this chapter.

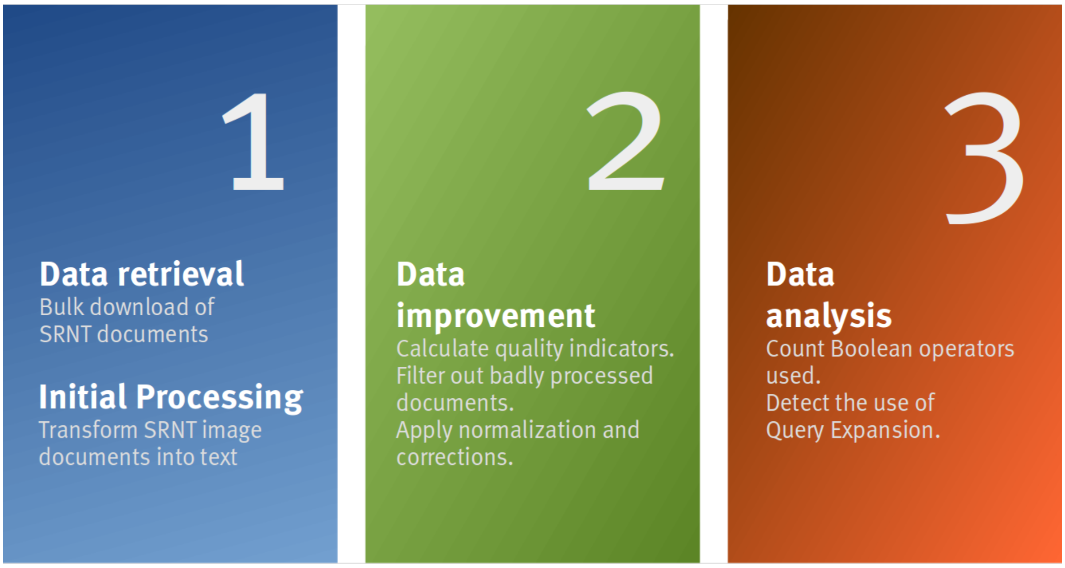
The target is to assemble a dataset of search query logs from a rather large collection of SRNT documents. This is the foundation for data analysis. The main challenges were posed by the rather large amount of data that needed to be processed and the fact that SRNT documents are retrieved as image PDF files. Many SRNT documents contain graphical structures that disturb OCR. These structures have to be removed. Various issues arise from badly scanned documents. Documents are sometimes rotated; there may be stains or even handwritten annotations. In some cases the content of these documents is hard to decipher even for human eyes.

Dependencies, Installation, Configuration

PyPAIR is available at `https://github.com/paper82/pypair`. It is developed in Python 2.7. The following dependencies have to be met:

- Tesseract 3.02
- Leptonica 1.71

[1] "Python Patent Application Information Retrieval"



1
Data retrieval
Bulk download of SRNT documents
Initial Processing
Transform SRNT image documents into text
2
Data improvement
Calculate quality indicators.
Filter out badly processed documents.
Apply normalization and corrections.
3
Data analysis
Count Boolean operators used.
Detect the use of Query Expansion.

Figure 4.1: From retrieval to analysis. Steps 1 and 2 are described in this chapter.

- Ghostscript 9.06
- ImageMagick 6.8.9
- OpenCV

Using the exact same versions of Tesseract and Leptonica as shown in the list above is crucial. Tesseract[2] is an OCR engine that was developed between 1985 and 1996 by Hewlett Packard. In 2005 Tesseract was open sourced. Since 2006, it is developed by Google. PyPAIR is rather tightly integrated with Tesseract.

Tesseract requires Leptonica, an image processing library[3] that offers a wide range of image operations (binary morphology, grayscale morphology, image scaling...). Ghostscript[4] is used for transforming PDF documents into TIFF files. ImageMagick[5] is used for various image processing routines. Finally, OpenCV[6] is used for line detection (discussed below).

Tesseract offers a mechanism called whitelisting to restrict the range of detectable chars. The mechanism is exploited to reduce potential errors during OCR. For example, the "hits" column is restricted to the detection of digits, the date column is restricted to the

[2] https://github.com/tesseract-ocr
[3] http://www.leptonica.com
[4] https://ghostscript.com/
[5] https://www.imagemagick.org
[6] http://opencv.org

detection of digits and the symbols "/" respectively ":". Tesseract whitelist files should be installed with the *install_whitelists* script provided by PyPAIR.

The application handles retrieval of SRNT documents and the preparation of the dataset, i.e. the generation of a set of structured SRNT documents in the form of text. The general process (see Figure 4.2) can be divided into three main parts. First, patent applications are retrieved from the USPTO. SRNT and 892 documents are kept. During the second step (processing), structured text files are generated from every SRNT document retrieved. The third step (improvement) aims at a removing badly processed documents from the dataset. To achieve this, several quality indicators are calculated for every SRNT document. These steps are described in the following sections.

4.2 Data retrieval and initial processing

This section describes the retrieval of SRNT documents and the initial processing steps for generating the dataset.

4.2.1 Data retrieval

Bulk download for USPTO public PAIR data is achieved by using a Google service (`https://www.google.com/googlebooks/uspto-patents-pair.html`). As stated by Google this data is no longer updated: newer applications can now be retrieved directly[7] from the USPTO, following a partnership between Reed Tech and the USPTO. Public PAIR can be downloaded at `http://patents.reedtech.com/Public-PAIR.php`. PyPAIR is not able to perform bulk download of SRNT documents directly from the USPTO.

GSUTIL[8], a tool provided by Google, is used to create a text file containing URL fragments of application files to be downloaded.

4.2.2 Initial processing

In this step all retrieved SRNT image documents are transformed to structured text files. This is the most challenging, time-consuming and critical task.

For every retrieved SRNT PDF document, the following steps are applied:

- convert each page of the document to a TIFF image,
- gather layout information from the currently processed page, i.e. the location and structure of the table containing the search queries; if no table is detected, the processing for the SRNT document is cancelled,
- store table columns as subimages,

[7] `https://pairbulkdata.uspto.gov/`

[8] `https://cloud.google.com/storage/docs/gsutil`

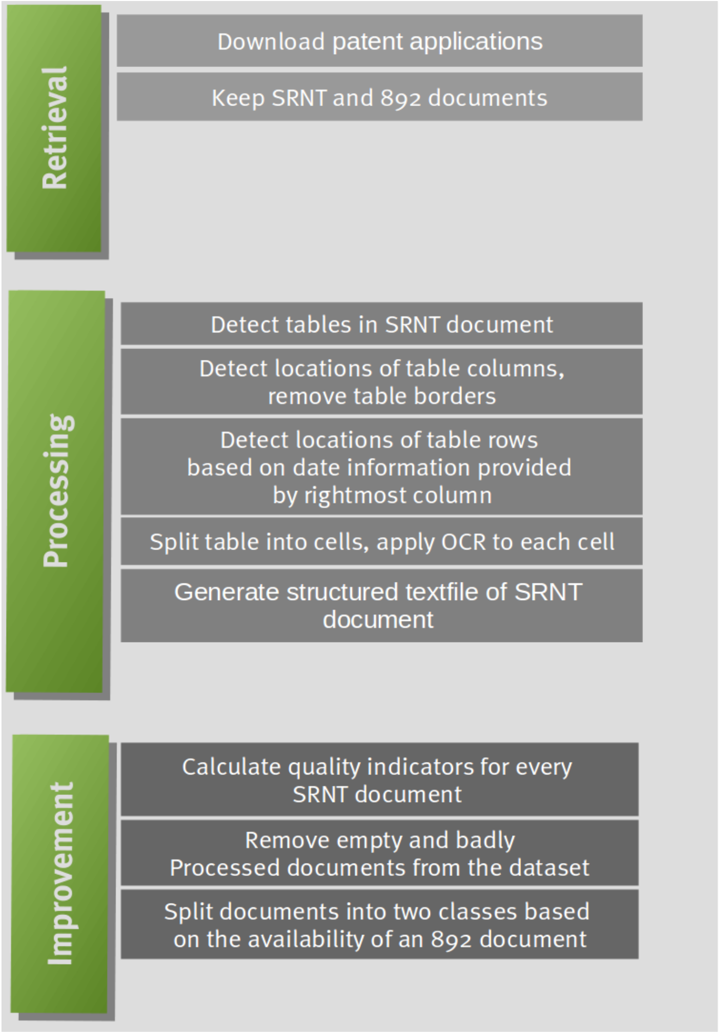
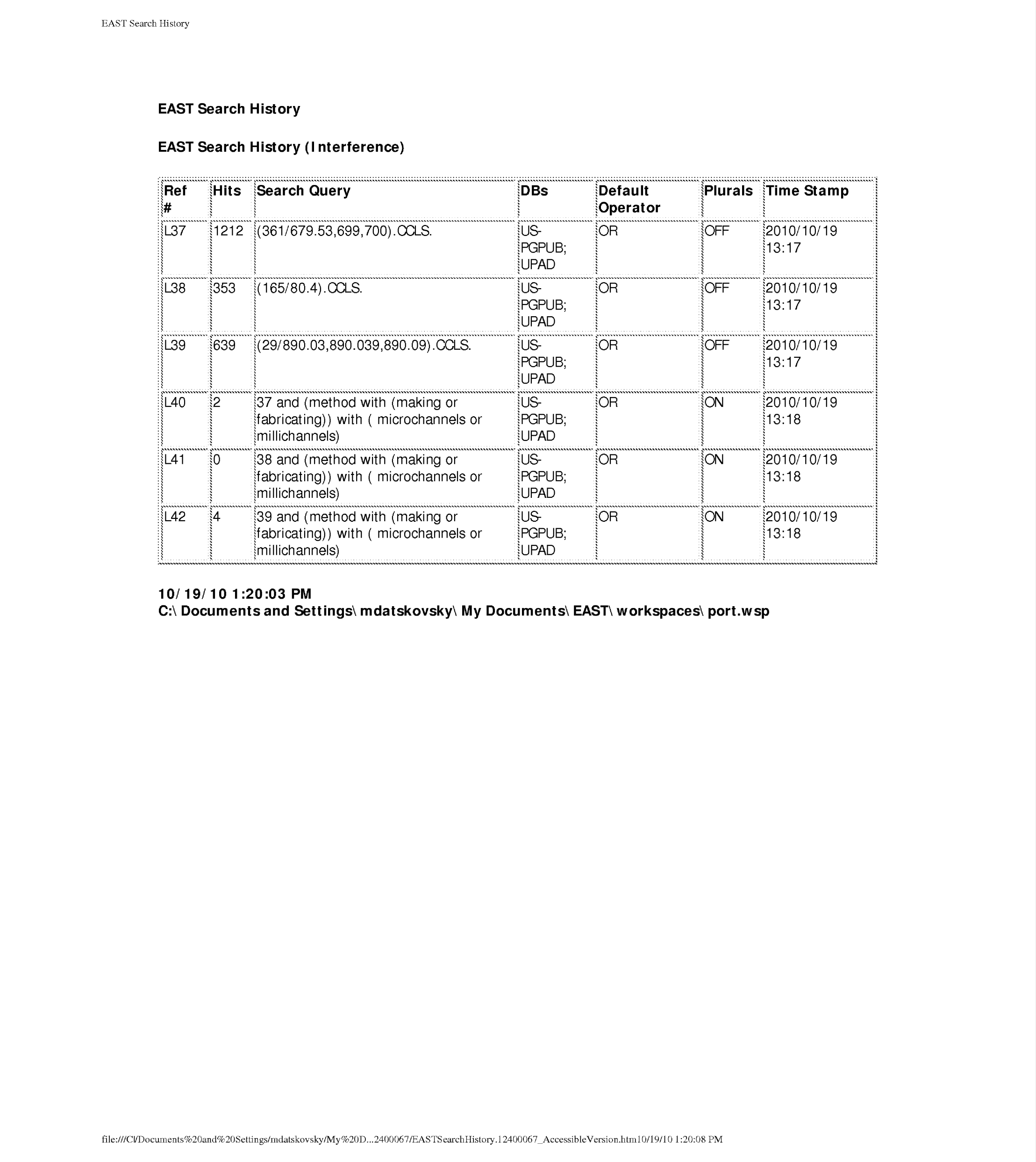

Retrieval
Download patent applications
Keep SRNT and 892 documents
Processing
Detect tables in SRNT document
Detect locations of table columns, remove table borders
Detect locations of table rows based on date information provided by rightmost column
Split table into cells, apply OCR to each cell
Generate structured textfile of SRNT document
Improvement
Calculate quality indicators for every SRNT document
Remove empty and badly Processed documents from the dataset
Split documents into two classes based on the availability of an 892 document

Figure 4.2: Generating the dataset.

- prepare subimages for OCR and apply OCR,
- parse results of OCR and store a text file containing the search queries.

Extracting search table and column dectection It can be seen in Figure 4.3 that it is not necessary to perform OCR across the whole page. The target of this step is to find relevant sections of the page, i.e. to detect the search table and the locations of it's columns.



EAST Search History

EAST Search History

EAST Search History (Interference)

Ref #	**Hits**	**Search Query**	**DBs**	**Default Operator**	**Plurals**	**Time Stamp**
L37	1212	(361/679.53,699,700).CCLS.	US-PGPUB; UPAD	OR	OFF	2010/10/19 13:17
L38	353	(165/80.4).CCLS.	US-PGPUB; UPAD	OR	OFF	2010/10/19 13:17
L39	639	(29/890.03,890.039,890.09).CCLS.	US-PGPUB; UPAD	OR	OFF	2010/10/19 13:17
L40	2	37 and (method with (making or fabricating)) with (microchannels or millichannels)	US-PGPUB; UPAD	OR	ON	2010/10/19 13:18
L41	0	38 and (method with (making or fabricating)) with (microchannels or millichannels)	US-PGPUB; UPAD	OR	ON	2010/10/19 13:18
L42	4	39 and (method with (making or fabricating)) with (microchannels or millichannels)	US-PGPUB; UPAD	OR	ON	2010/10/19 13:18

10/19/10 1:20:03 PM
C:\ Documents and Settings\ mdatskovsky\ My Documents\ EAST\ workspaces\ port.wsp

file:///C|/Documents%20and%20Settings/mdatskovsky/My%20D...2400067/EASTSearchHistory.12400067_AccessibleVersion.htm10/19/10 1:20:08 PM

Figure 4.3: Input image (patent application 12400067).

Detecting a certain number of table borders is a requirement for further processing. If no borders are detected the process is cancelled. In many SRNT documents the table borders are not solid lines but rather made up of small artefacts, which are interepreted as characters by Tesseract. Therefore they have to be removed.

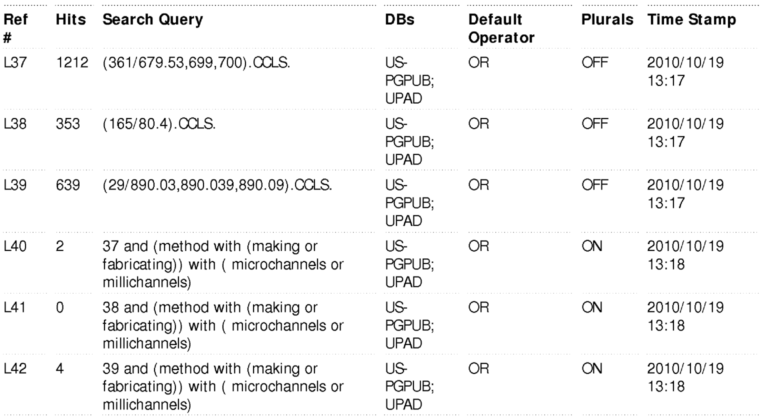
Relevant section detection

The idea behind the algorithm is as follows. First, the average hue for every row of the image is calculated. A value of 255 represents a white row; a threshold of 254.5 is used to add some tolerance for impurities. Every line with an average hue below the given threshold might be relevant, i.e. part of the main document body. The average hue for every row $0...n$ is calculated.

Column detection

Probabilisitc Hough transformation [DH72] is used to detect lines in the potentially relevant sections of the SRNT document page. To improve results, a Canny edge detector is applied to the image before line detection takes place.

Vertical and, if present, horizontal borders must be removed before OCR is carried out. An ideal result of this step is an image that only contains the document section of interest (see figure 4.4). Usually, the tables containing search queries contain seven columns. Some SRNT documents do not contain the "default operator" column.



Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L37	1212	(361/679.53,699,700).CCLS.	US-PGPUB; UPAD	OR	OFF	2010/10/19 13:17
L38	353	(165/80.4).CCLS.	US-PGPUB; UPAD	OR	OFF	2010/10/19 13:17
L39	639	(29/890.03,890.039,890.09).CCLS.	US-PGPUB; UPAD	OR	OFF	2010/10/19 13:17
L40	2	37 and (method with (making or fabricating)) with (microchannels or millichannels)	US-PGPUB; UPAD	OR	ON	2010/10/19 13:18
L41	0	38 and (method with (making or fabricating)) with (microchannels or millichannels)	US-PGPUB; UPAD	OR	ON	2010/10/19 13:18
L42	4	39 and (method with (making or fabricating)) with (microchannels or millichannels)	US-PGPUB; UPAD	OR	ON	2010/10/19 13:18

Figure 4.4: Resulting image after main document section was extracted.

Row detection Horizontal table lines are not always present in SRNT documents. The strategy to identify the locations of table rows, i.e. the different search queries, is as follows: as can be seen in Figure 4.4 a common factor of all search rows is a date string in the rightmost column. This is exploited to discriminate among different rows. The rightmost column is extracted as depicted in figure 4.5 and Tesseract is applied to the extracted subimage.

The BeautifulSoup[9] library is used for parsing the HTML content. Date entries in a format of $YYYY/MM/DD$ are easily detected with a regular expression. The "title" attribute of a HTML node contains layout information, such as the x and y coordinates of an entry in relation to the image dimensions. Once a date entry is found the y coordinate

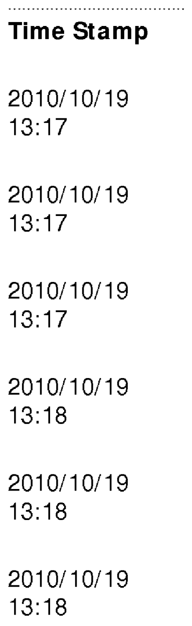
[9] https://www.crummy.com/software/BeautifulSoup/

is extracted from the "title" attribute and stored. This value indicates the top coordinate of the start of a search row.

Applying Tesseract At this point the coordinates of table columns and table rows are known. The search table is split into subimages of table cells. Subimages are inverted first. Then, two morphology kernels are applied via ImageMagick:

1. -morphology close rectangle 1x3: reduces or removes small gaps,
2. -morphology erode:8 square: expands the black areas of the image, i.e. thinning out the letters.

Finally, the image is inverted again. A 100 pixel wide white border is added to the image, since Tesseract might encounter issues when letters are located too close to the border of the image.



Time Stamp

2010/10/19
13:17

2010/10/19
13:17

2010/10/19
13:17

2010/10/19
13:18

2010/10/19
13:18

2010/10/19
13:18

Figure 4.5: The subimage of the time stamp column

It is known which characters a processed cell might contain. For example, the "hits" column only contains digits. By using a Tesseract whitelist, the range of detectable chars is reduced. This, in turn, lowers the possibility of detecting wrong characters (e.g. "O" instead of "0").

Result

Finally, the content of the data structure is stored as a text file. Each search query is stored as a sequence of lines in a key/value scheme (see Listing 4.1); search queries are separated by a blank line.

Listing 4.1: The result of the transformation process

```
Ref: L37
Hits: 1212
Query: (361/679.53,699,700).CCLS.
Dbs: USPGPUB;UPAD
Def_Op: OR
Plurals: OFF
Time: 13:17
Date: 2010/10/19

Ref: L38
Hits: 353
Query: (165/80.4).OCLS.
Dbs: USPGPUB;UPAD
Def_Op: OR
Plurals: OFF
Time: 13:17
Date: 2010/10/19

Ref: L39
Hits: 639
Query: (29/890.03,890.039,890.09).CCLS.
Dbs: USPGPUB;UPAD
Def_Op: OR
Plurals: OFF
Time: 13:17
Date: 2010/10/19

Ref: L40
Hits: 2
Query: 37 and (method with (making or fabricating)) with ( microchannels or
    millichannels)
Dbs: USPGPUB;UPAD
Def_Op: OR
Plurals: ON
Time: 13:18
Date: 2010/10/19

Ref: L41
Hits: 0
Query: 38 and (method with (making or fabricating)) with ( microchannels or
    millichannels)
Dbs: USPGPUB;UPAD
Def_Op: OR
Plurals: ON
Time: 13:18
```

```
Date: 2010/10/19

Ref: L42
Hits: 4
Query: 39 and (method with (making or fabricating)) with ( microchannels or
    millichannels)
Dbs: USPGPUB;UPAD
Def_Op: OR
Plurals: ON
Time: 13:18
Date: 2010/10/19
```

Possible sources for failure

A satisfying result is shown in listing 4.1. There are different circumstances under which the transformation might fail completely or lead to undesirable results.

Main reasons for the failure of obtaining an acceptable result are:

Document layout As described above, the processing algorithm depends on the borders of a table to be present in order to correctly process a document. If no, not enough or too many borders are found, the document can not be processed. The presence of *two* tables containing search queries within the same page is not detected. If these tables are not vertically aligned, portions of text of the second table is likely to be removed by the column removal routine.

Image rotation Image derotation is performed. However, if the image rotation is too pronounced, PyPAIR will fail to assign table cells to the correct column.

Failing to recognize search rows A critical issue occurs when the algorithm fails to detect one or more dates. This leads to a merge of two or more succeeding search queries. A strategy to detect this kind of error is presented in the next section.

OCR issues PyPAIR has to rely upon the quality of the OCR. The quality of OCR depends on the input quality of the SRNT document. In some cases the quality of the SRNT document is not optimal. The contrast of the image might be too low, there are stains, etc. To make things worse, search queries are likely to contain very specific terms, names and wordings. Only certain OCR errors can be automatically corrected.

4.3 Data improvement

This section describes measures for improving the quality of the generated dataset.

4.3.1 Introduction

In the previous section SRNT documents have been retrieved and initial processing has taken place. The dataset is now available in the form of text. The target of this step is to improve the quality of the dataset. This includes the detection and removal of badly processed documents and the correction of various OCR errors.

Obtaining a perfect representation in the form of text of a collection of scanned documents is not realistic. There are cases of errors which can not be corrected without human interference (e.g. a wrongly detected number of hits such as 63 instead of 630); other errors can be detected and repaired.

There are different types of errors that will occur during processing. It is also easy to see that not every fault will have the same impact on the data quality.

More harmless types of errors include misinterpretations of characters by OCR, such as the recognition of "OSPTO" instead of "USPTO" or "c|s" instead of "cls". Other errors are more serious. An example is the aforementioned missing of a digit in the hits column. In such cases it is not possible to perform any automatic correction.

Once all SRNT documents are available in the form of text the following measures are taken to improve the general quality of the dataset:

1. normalization/validation of the data,
2. calculation of quality indicators for every search document,
3. detect badly processed documents by their quality indicators and remove them from the dataset
4. improve the quality of the search query string for the remaining documents by correcting various known OCR mistakes (e.g. the detection of "cc|s" instead of "ccls").

Terminology

For every search query certain additional data is available (e.g. the date and time when a query was issued, the names of databases that have been searched...). The set of search query and additional query data will be refered to as *searchrecord* or *record*. The *fields* of the record are described in the next section. A SRNT document usually contains a series of search records, forming a *searchsession*. From now on the following terminology is used:

Search query / Query Refers to the query string itself.

Search record / Record Refers to the set of all available data for a given search query.

Field Refers to the field or the field's value of the search record.

Search document / (SRNT) document Refers to a set of search records, i.e. a single SRNT document.

Dataset/Document set Refers to the whole set of search documents.

4.3.2 Available fields

For many search fields the data type and/or range of allowed values is known. This knowledge is exploited to validate and normalize the data. The following fields are stored:

Ref The reference field stores a reference under which an already issued search query can be addressed at any later point of the search. Once a search query has been issued it can be reused again by including it's reference in suceeding queries. This avoids having the user to formulate the query again. The usual form of the reference is a number prefixed by a char (S or L) - e.g. $S1$, $S2$. A common error is the detection of the prefix S as digit (5) or the detection of 1 as l and 0 as O.

Hits This field contains the number of results retrieved by a query. An integer typecast is performed, even though Tesseract was instructed for this column to detect digits only. If the typecast fails an empty string is stored.

Query This field contains the Boolean search query provided by the user. A query may contain alphanumeric as well as various special characters. A very frequent issue is the interpretation of the letter l as pipe symbol (|). Several known issues, such as the detection of "c|m" instead of "clm", are fixed. Superfluous whitespace is removed. All letters are lowercased. The unmodified original value is kept and stored in a new field (*Query*(*Orig.*). See Listing 4.2 for an example of improving/normalizing a query.

Listing 4.2: Example for query normalization

```
$(FRAME same (light NEAR2 source) same (SPECTRAL OR co|or)).c|m.$
$(frame same (light near2 source) same (spectral or color)).clm.$
```

Databases This field contains a list of databases (see Table 4.1) that have been searched in by the given query.

In a very simple approach the name of every known database is searched for in the field's value. If the database name is detected, the database is included in the normalized field value (see Table 4.2) . The approach does not consider misspellings such as "JPU" instead of "JPO", etc.

Default Operator (Def_Op) The value of this field specifies the default Boolean operator that is applied whenever no Boolean operator is explicitly set between two keywords. Not every SRNT document includes the default operator.

Table 4.1: Available databases

Name	Database
USPAT	US Patent database (full text)
USPGPUB	US Patent database
USOCR	US Patent database (OCR)
FPRS	French Patent Regristration System
EPO	European Patent Office
JPO	Japan Patent Office
DERWENT	Derwent World Patents Index (DWPI)
IBMTDB	IBM patent database

Table 4.2: Normalizing the database field

Input	Output
USPAT	USPAT
USPAT;IBMTDB;JPU	USPAT;IBMTDB
JPOOSPATEPOIBJTDB;	JPO;EPO;

Plurals This field indicates if plural forms of provided keywords were automatically considered by the search system. The value of this field is either "ON" or "OFF".

Date A valid date in the format YYYY/MM/DD is expected. If a date string is not valid (e.g. 2008/12/32), the value is set to an empty string.

Time A valid time in the format HH:MM is expected. If the value of the field is invalid (e.g. 24:01), the value is set to an empty string.

4.3.3 Indicators of processing quality

Processing issues have already been described. Each issue has a different impact on the quality of the dataset. To improve the quality of the dataset as a whole, badly processed documents are identified and removed. For every search document four quality indicators are obtained. Thresholds are used to decide if a document is kept in the collection.

The following quality indicators have been implemented.

Merged queries A critical error can be observed when row detection fails. In this case, two or more suceeding search queries of a SRNT document are *merged* into one search record. One method of detecting these errors is to examine the value of the plurals field. In affected records the value will be $ONONON$ or $ONOFFOFFON$, etc; affected records can simply be detected by measuring the length difference between normalized field value (e.g. *ON*) and original value (e.g. *ONONON*). The length difference does not directly express how many search queries were merged into one search record. *ON* versus *ONONON* versus *ONOFFOFF* yield different lengths, but all cases represent the fact that three search queries were merged. This indicator stores the number of search records that (are suspected to) contain merged queries.

Invalid date This indicator is the number of search records where the value of field "date" is empty (=invalid).

Invalid time This indicator is the number of search records where the value of field time is empty (=invalid).

Empty queries ratio This indicator is the ratio of search records where the query is empty to the total of search records within the SRNT document.

Particular bad results have been observed for two groups of SRNT documents. In the first group, every second row of the search table has a background texture. Therefore, OCR will fail. The other group contains SRNT documents that have been scanned with considerable degrees of rotation. Image derotation is carried out. However, the removal of vertical and horizontal table borders demands table borders that are paralell to the document edges. If this is not the case it is likely that portions of text are removed together with the table borders. This, in turn, might lead to failed date detection and missing information in other cells.

The following decision rules determine if a SRNT document is kept:

1. the SRNT document must not be empty (i.e. initial processing was successful),
2. indicator MERGED QUERIES must be 0,
3. indicator INVALID DATE is 0,
4. indicator INVALID TIME is 0,
5. indicator EMPTY QUERIES RATIO must be below 0.08 (i.e., not more than ca. 1 out of 13 queries is empty). This threshold was set to a value of > 0 in order to keep otherwise perfectly recognized documents.

4.3.4 Examining processing quality

Of the total of $1,037,420$ downloaded SRNT image documents, $656,221$ were successfully initially processed (i.e. search queries could be extracted).

Of all 656,221 successfully processed SRNT documents, 548,441 (ca. 90%) were not affected by merged queries. Of those, 6,873 documents contained at least one search

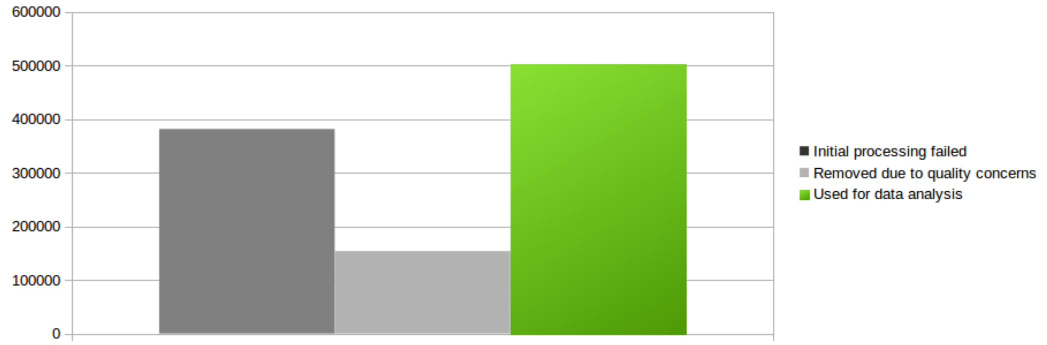


Figure 4.6: Processing result for $1,037,420$ retrieved SRNT documents

record with an invalid date. These documents were removed. From the remaining 541,568 SRNT documents, a rather high amount of 38,715 contained at least one search record with an invalid time value. Finally, from the remaining 502,853 documents, 213 documents (with an EMPTY QUERIES RATIO > 0.8) were removed.

Of 1,037,420 initially retrieved documents, 502,640 SRNT documents are used for analysis (see Figure 4.6).

4.4 Summary

In this chapter the process of assembling a dataset of search query logs from SRNT documents retrieved from the USPTO has been presented. 1,037,420 SRNT documents have been retrieved. After the removal of invalid and badly processed documents, 502,640 SRNT documents are left for analysis. The application PyPAIR, responsible for retrieval, initial processing, calculation of quality indicators and for applying improvements to the obtained data, has been discussed.

CHAPTER 5

Data Analysis

This chapter presents the results of the data analysis.

5.1 Introduction

The examined data set consists of a total of 502,640 SRNT documents (= search logs) with a total of 15,519,715 queries. SRNT documents contain Boolean search queries issued by a patent searcher to a patent search engine. The set of SRNT documents was divided into two groups. For SRNT documents of the first group a 892 document was found. The presence of an 892 document for an SRNT document indicates that relevant patent documents where found during the course of the search. This group is referred to as $SRNT_{892}$. For SRNT documents of the second group no 892 document was found. This group is referred to as $SRNT_{no892}$. $SRNT_{all}$ refers to documents in both groups.

We postulate that for every current invention at least one relevant patent document exists. This implies that the absence of an 892 document indicates the patent search was not conducted well. This thesis examines various quantitative characteristics of Boolean search queries. Of special interest are the differences between both groups of search logs ($SRNT_{892}$, $SRNT_{no892}$), i.e. the use of different Boolean operators, the application of QE, the use of certain search fields, the use of references to address former queries, etc.

In order to determine if a 892 document is available, patent application number and date of SRNT and 892 document filename must match.

For $250,017$ (49.74%) SRNT documents a 892 document was found. For the remaining $251,687$ (50.07%) SRNT documents, no 892 document was found.

Methodology

The general approach was to extract *features* either from an SRNT document (e.g. document length or average query length) or from all queries regardless of their document. The weighted average (avg), the median value, minimum (min) and maximum (max) as well as the variance (var) are then calculated from the obtained features and presented for both document sets ($SNRT_{892}$ and $SRNT_{no892}$).

Data analysis is split into three sections. In the first section various general examinations are made such as: 1. the average SRNT document length, 2. the average query length, 3. the use of parentheses, 4. the use of several search fields available by the EAST search system (e.g. search in claims), 5. the popularity of the various patent databases that are searched, and 6. the use of references to address former queries. The second section examines the use of Boolean operators, e.g. operator popularity and co-occurrence of operators. In the third section the application of QE by the patent searcher is examined, such as the use of the truncation operator and the providing of lists of alternate terms within queries.

5.2 General examinations

In this section several general examinations (such as the average SRNT document length, the average query length, the use of parentheses, the searched patent databases or the use of references to address former queries) are presented.

5.2.1 SRNT document length

The length of a SRNT document is defined by the the number of search queries the document contains. The average SRNT document length is 30.88 (see Table 5.1). The average document length of set $SRNT_{892}$ (average 36.18, median 24.0) surpasses the values of set $SRNT_{no892}$ (average 25.63, median 14.0).

Table 5.1: Number of search queries in SRNT documents

	$SRNT_{all}$	$SRNT_{892}$	$SRNT_{no892}$
avg	30.88	36.18	25.63
median	19.0	24.0	14.0
max	1937	1775	1937
min	1	1	1

Based on the lengths of the obtained SRNT documents it was decided to devise eight length groups (see Table 5.2) for better data visualization. Every SRNT document was then assigned into one length group.

64.17% of very short SRNT documents ($\leq$ 10 queries) belong to $SRNT_{no892}$. Around 50.0% of short SRNT documents (between 10 and 25 queries) fall into $SRNT_{892}$. 60.27% of medium and long SRNT documents (between 25 and 500 queries) belong to $SRNT_{892}$ (see Table 5.2, Figures 5.1 and 5.2). The high amount of documents with a length of 1 seems suspicious at first: an examination of these documents, however, did not reveal

Table 5.2: Length groups of SRNT documents

queries n	$SRNT_{all}$	$SRNT_{892}$	% of class	$SRNT_{no892}$	% of class
$n = 1$	$23,882$	$5,796$	24.27%	$18,390$	77.00%
$2 < n \leq 10$	$135,326$	$51,703$	38.20%	$83,788$	61.91%
$10 < n \leq 25$	$147,183$	$74,817$	50.83%	$72,530$	49.27%
$25 < n \leq 50$	$106,543$	$62,242$	58.41%	$44,434$	41.70%
$50 < n \leq 100$	$641,20$	$40,060$	62.59%	$24,166$	37.68%
$100 < n \leq 500$	$244,54$	$15,306$	62.59%	$9,190$	37.58%
$500 < n \leq 1000$	174	86	49.42%	88	50.57%
$n > 1000$	22	7	31.81%	15	68.181%
Total	15,496,714	9,045,921	58.37%	6,450,793	41.63%

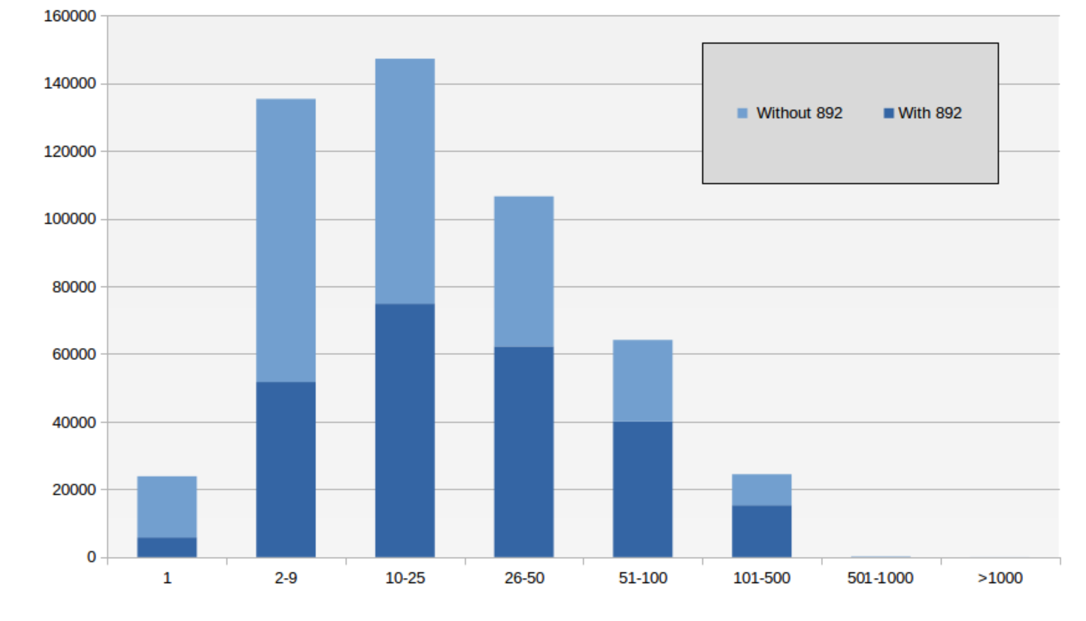


Figure 5.1: Length groups of SRNT documents

faulty processed documents. For 77% of all SRNT documents with one search query, no 892 document has been found.

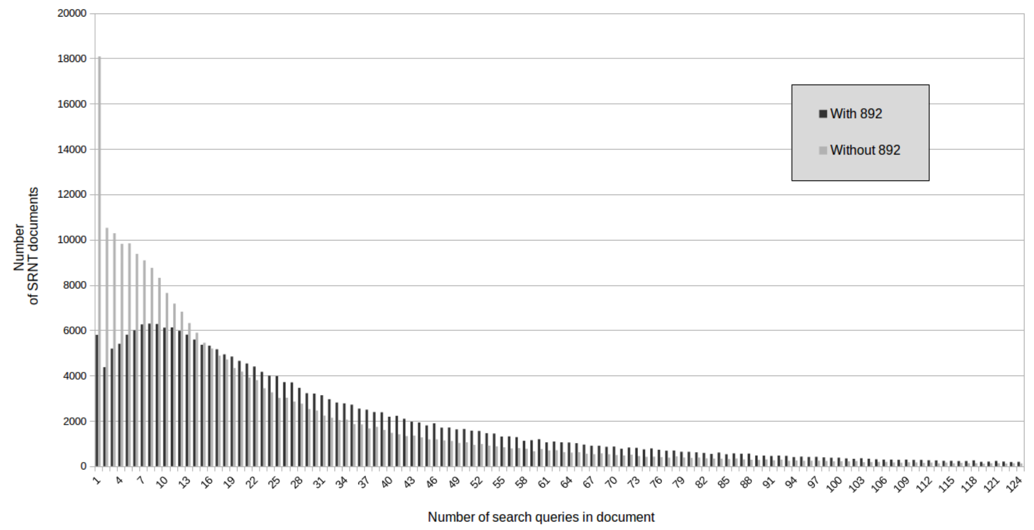


Figure 5.2: Total of SRNT documents with given number of search queries for documents with up to 125 queries

5.2.2 Query length

The length of a search query is defined by it's term count (keywords and Boolean operators). The average query length in documents of $SRNT_{892}$ is 9.55 and 9.12 in documents of $SRNT_{no892}$. The average maximum query length and the average of the variance is higher in documents of $SRNT_{892}$: the average maximum query length is 43.24 in $SRNT_{892}$ and 28.52 in $SRNT_{no892}$. The median variance in query length is 28.48 for documents in $SRNT_{892}$ and 17.51 for documents in $SRNT_{no892}$ (see Table 5.3).

Table 5.3: Search query length by document.

	$SRNT_{892}$	$SRNT_{no892}$
avg	9.55	9.12
median	5.0	5.0
avg of max	43.24	28.52
median of max	21.0	17.0
avg of min	2.12	3.29
median of min	1.0	1.0
median of var	28.48	17.51

To provide a better visualisation of the data at hand, every query was categorized into a length group of very brief (1-4 terms), brief (5-9 terms), medium (10-99 terms), long (100-499 terms) and very long queries (more than 500 terms).

Around 70% of all submitted queries are assigned to the groups of very brief or brief queries with *one* to *nine* terms (see Table 5.4, Figure 5.3). Less than 1% of all queries have a length of more than 100 terms. For a small percentage of queries (0.0037%), the issued query string was detected as empty.

Table 5.4: Length classes of search queries

Query length	$SRNT_{all}$	% of $SRNT_{all}$	$SRNT_{892}$	% of set	$SRNT_{no892}$	% of set
0 (indicates data issue)	583	0.0037	289	0.003	294	0.0045
very brief						
1	3040356	19.61%	1650378	18.24%	1389978	21.54%
2-4	4160264	26.84%	2445724	27.03%	1714540	26.57%
brief						
5-9	4191138	27.04%	2462132	27.21%	1729006	26.80%
medium						
10-19	2689810	17.35%	1622865	17.94%	1066945	16.53%
20-49	1188091	7.66%	716941	7.92%	471150	7.30%
50-99	157955	1.01%	98735	1.09%	59220	0.91%
long						
100-199	49316	0.31%	34523	0.38%	14793	0.22%
200-499	14960	0.09%	11157	0.12%	3803	0.05%
very long						
500-999	3602	0.02%	2728	0.03%	874	0.01%
1000-1999	559	0.0036%	404	0.0044%	155	0.0024%
2000+	42	0.0002%	34	0.0003%	8	0.0001%

Relation between query length and query type

There seems to be a relation between query length and the ratio of digits to all characters of a query. In very brief queries ($n = 1$) as well as long or very long queries, the average ratio of digits to all characters of a query is more than 50.0% (see Figure 5.4). This indicates that in these kind of queries searchers are rather providing lists of patent numbers or patent classification classes.

5.2.3 Use of parentheses

Parentheses encapsulate a group of terms (see Listing 5.1).

Listing 5.1: Example use of parentheses

```
(anti adj vibrati?$2)
and
(applicator or applying or deposit?$3 or
dispens?$3 or rod or roller or roll or cylinder or inject?$3)
and
```

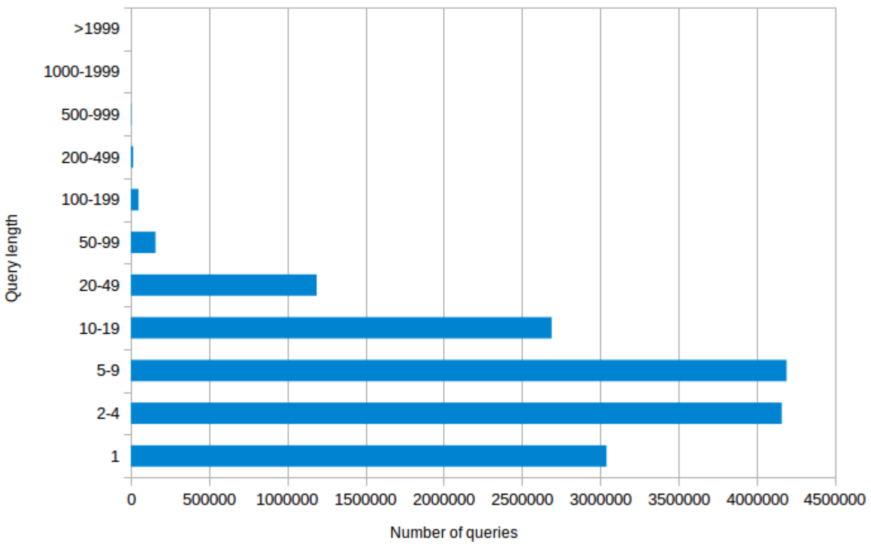


Figure 5.3: Number of queries in length groups

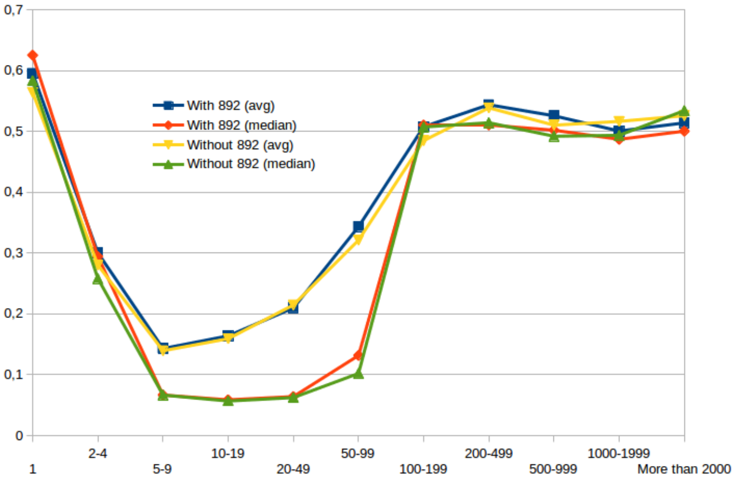


Figure 5.4: Ratio of digits to all alphanumeric characters by query length.

```
(doctor?$3 or blade or -rod)
and
(stage or platform)
and
(fixed near3 plate).clm.
```

In 96.09% ($SRNT_{892}$) and 92.96% ($SRNT_{no892}$) of all documents at least one query contains an opening parenthesis. In these documents, at least one opening parenthesis is present in 60.17% ($SRNT_{892}$; median: 61.11%) and 63.19% ($SRNT_{no892}$; median: 64.94%) of all queries. In queries with at least one opening parenthesis an average number of 2.5 occurrences of opening parentheses is counted (see Table 5.5). Documents in both sets are affected by an OCR issue that leads to the detection of too many parentheses. It is assumed, however, that both document groups are affected with similar frequency.

Table 5.5: Number of open parentheses in queries with at least one opening parenthesis

	$SRNT_{892}$	$SRNT_{no892}$
avg	2.4475	2.5113
avg of median	2.0645	2.2200
avg of max	6.3964	5.2863
avg of min	1.2260	1.4586

Boolean queries can be *nested*, i.e. the use of a pair of parentheses within another pair of parentheses. Here, the nesting level of a Boolean query is calculated as follows. Starting from the leftmost character of a query, all characters are iterated. The nesting level nl is increased whenever an opening parenthesis is seen and decreased whenever a closing parenthesis is seen. The nesting level is the maximum reached value of nl. As shown in Table 5.7 and Figure 5.5 the shares of queries with the same nesting level is nearly identical for both document groups.

Table 5.6: Example for calculating the nesting level.

Query	Nesting level
term1 and (term2 or term3)	1
term1 and (term2 or (term3 and term4))	2
term1 and (term2 or (term 3 and (term 4 xor term5)))	3
(term1 near term2) and (term1 same term3)	1
(term1 or (term2 near term3) and term4)	2

5.2.4 EAST specific searchfields

The EAST search engine allows to specify database fields to be searched. The searcher appends the field code (e.g. '.clm.' to search in claims or '.ccls.' to search in patent classification classes) to a keyword. On average, '.ccls.' is used in 12.43% ($SRNT_{892}$) respectively 17.13% ($SRNT_{no892}$) of all queries in a document. '.clm.' is applied less often in $SRNT_{892}$, where, on average, it is found in 2.72% of all queries of a document

Table 5.7: Shares of queries with given nesting level.

Maximum nesting level	% of all queries in $SRNT_{892}$	% of all queries in $SRNT_{no892}$
0	$43,58\%$	$44,81\%$
1	$38,38\%$	$37,40\%$
2	$13,99\%$	$13,49\%$
3	$3,27\%$	$3,37\%$
4	$0,61\%$	$0,70\%$
5	$0,10\%$	$0,15\%$
>5	$0,06\%$	$0,08\%$

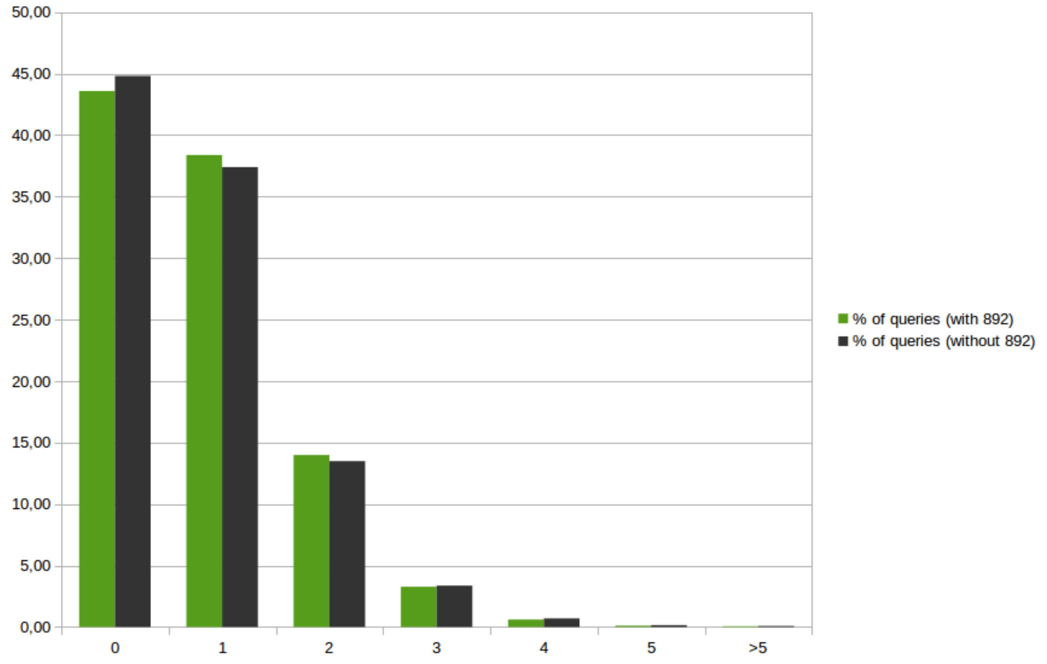


Figure 5.5: Share of queries with given nesting level.

as opposed to 10.29% of $SRNT_{no892}$. The inventor field ('.inv.') is comparatively seldom used. Finally, the patent number field ('.pn.') is used in 12.04% of all queries in a document of $SRNT_{892}$ and in 9.51% of all queries in a document of $SRNT_{no892}$. Results are presented in Table 5.8 and Figure 5.6.

Table 5.8: Usage of search fields in SRNT documents

Search field	abbrev.	% of queries $SRNT_{892}$	% of queries $SRNT_{no892}$
Classification	ccls	12.43%	17.13%
Claims	clm	2.72%	10.29%
Inventor	inv	0.27%	0.41%
Patent number	pn	12.04%	9.51%

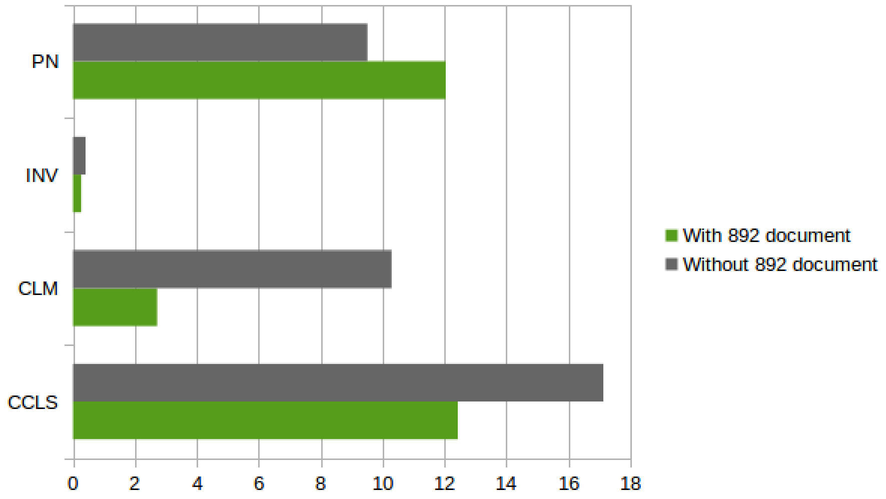


Figure 5.6: Use of east specific search fields (avg. % of queries in a document in which field is used)

5.2.5 Patent databases searched

Search queries can be issued to different patent databases (see Table 5.9 and Figure 5.7). Nearly 100.00% of all queries search patents in the US full text patent database (USPAT). Only 32.41% of all queries of $SRNT_{892}$ and 21.86% ($SRNT_{no892}$) search patents registered in the French Patent Registration System (FPRS).

On average, 4.8 ($SRNT_{892}$) respectively 4.3 ($SRNT_{no892}$) databases are searched in by a query.

5.2.6 Use of references

Once a query is issued to the search engine a *reference* is assigned it (e.g. "S1", "S2", "S3" ...). This mechanism is exploited to reuse a query in a new search query without having to type the already issued query again.

Table 5.9: Databases searched in

Database	% of queries ($SRNT_{892}$)	% of queries ($SRNT_{no892}$)
USPAT	98.00%	96.28%
USPGPUB	93.35%	88.99%
USOCR	55.65%	49.70%
JPO	55.77%	55.38%
IBMTDB	40.11%	38.54%
FPRS	32.41%	21.86%
EPO	57.23%	57.21%
DERWENT	53.02%	51.82%

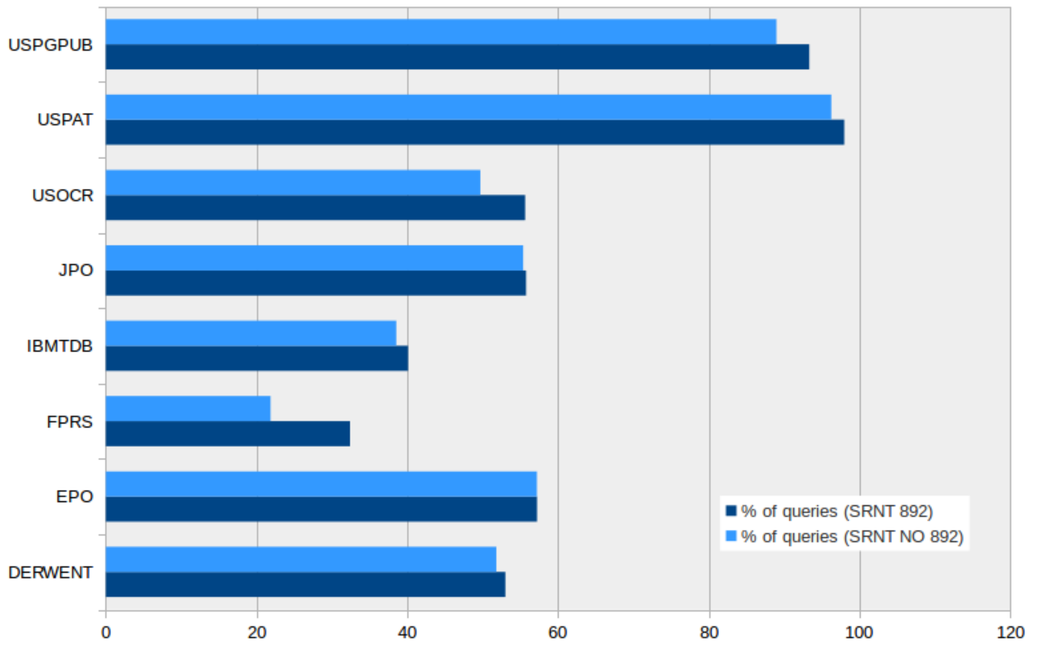


Figure 5.7: % of queries that search patents of given databases

Usually a reference is prefixed by the letter "S" or "L". The following cases are recognized as references:

- character "S" followed by digits ("S1", "S2"...),
- character "L" followed by digits ("L1", "S2"...).

Table 5.10: Number of patent databases addressed to by query.

	$SRNT_{892}$	$SRNT_{no892}$
avg	4.8767	4.3870
avg max	5.7710	5.2219
avg min	3.0629	2.7991
avg median	5.0609	4.5622
avg variance	1.6337	1.5200

At least one query containing a reference is found in 58.50% of all documents of $SRNT_{892}$ and in 45.77% of all documents of $SRNT_{no892}$. On average, 28% of all queries of these documents ($SRNT_{892}$ and $SRNT_{no892}$) contain at least one reference.

5.3 Use of Boolean operators

Boolean operators control which search terms a document must or must not contain in order to be included in the result set. The use of Boolean operators is essential but not mandatory. If omitted, the search system uses the operator specified by the "default operator" field. A default operator of, say, "AND" would rewrite a given query "termA termB" to "termA AND termB". The default operator field was not considered, since not every SRNT document provides this information.

5.3.1 Popularity of operators

It should come to no surprise that OR and AND are the most frequent, XOR and NOT the most seldom used operators. The relative operator popularity is similar for both document sets (see Table 5.11, Figure 5.8). Table 5.11 and Figure 5.8 both show the *share* of operators of each Boolean operator type in the total of all counted operators.

Table 5.12 shows the percentages of queries in which the given Boolean operator type appears at least once.

In roughly 71% of all queries a Boolean operator is used. The AND operator appears in ca. 40%, the OR operator in ca. 27% of all queries. Still, the total count of OR is considerably higher. This indicates that OR is rather repeatedly used in single queries. ADJ, NEAR, SAME, WITH appear in 10 to 20% of all queries (see Table 5.12).

5.3.2 Ratio operator count to query length

For every search query the ratio between operator count and query length was obtained. Results are similar for $SRNT_{892}$ and $SRNT_{no892}$.

Table 5.11: Ratio of occurrences of Boolean operators of given type to the total count of all operators.

Operator	% of all operators ($SRNT_{892}$)	% of all operators ($SRNT_{no892}$)
OR	$48,11\%$	$44,45\%$
AND	$20,65\%$	$24,05\%$
NOT	$0,64\%$	$0,72\%$
XOR	$0,003\%$	$0,003\%$
ADJ	$5,93\%$	$7,06\%$
NEAR	$9,23\%$	$8,87\%$
SAME	$7,08\%$	$6,52\%$
WITH	$8,32\%$	$8,30\%$

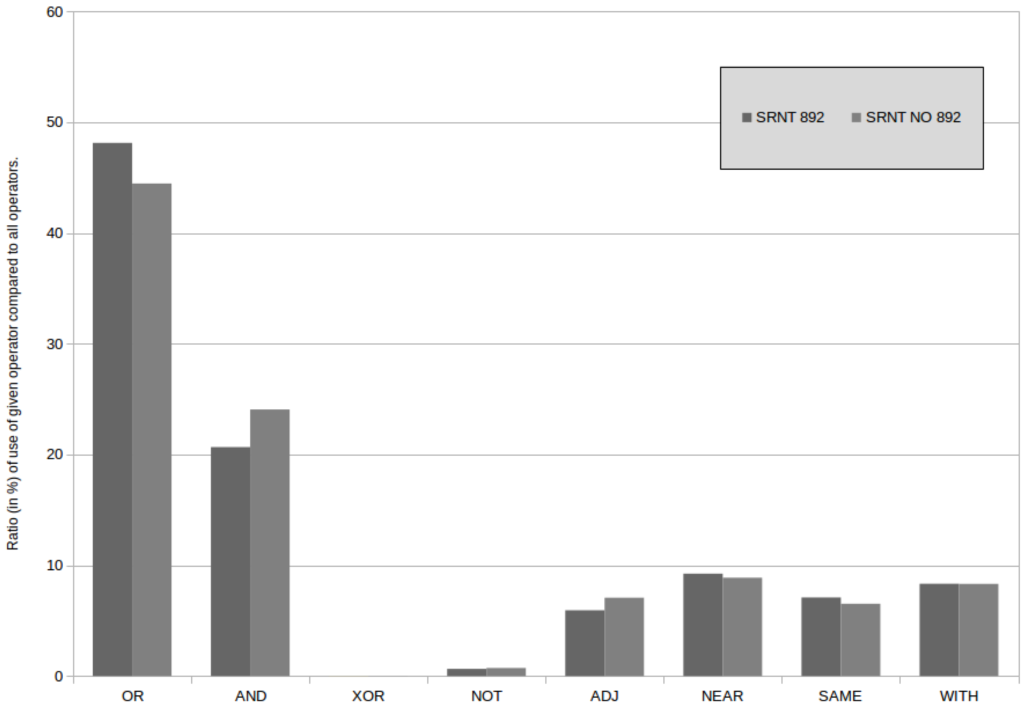


Figure 5.8: Popularity of Boolean operators.

It can be seen (Figure 5.9) that the ratio between operator count and query length remains similar for queries even of very different length, with an average ratio varying usually between 0.3 and 0.4. The ratio of the OR operator to the query length increases steadily (Figure 5.10). Contrary, the ratio of AND to the query length is highest in very

Table 5.12: % of queries with at least one use of an operator of the given type.

operator	in % of queries in $SRNT_{892}$	in % of queries in $SRNT_{no892}$
(any)	72.38%	70.82%
OR	28.65%	26.05%
AND	41.64%	42.01%
NOT	1.74%	1.79%
XOR	0.01%	0.01%
ADJ	11.21%	12.45%
NEAR	18.66%	16.46%
SAME	13.45%	11.84%
WITH	16.01%	14.57%

short (2-4 terms) queries (Figure 5.11). The ratios of ADJ, NEAR, SAME, WITH are highest in short (5-9 terms) queries (Figure 5.12). OR is the only operator of significant use in long and very long queries.

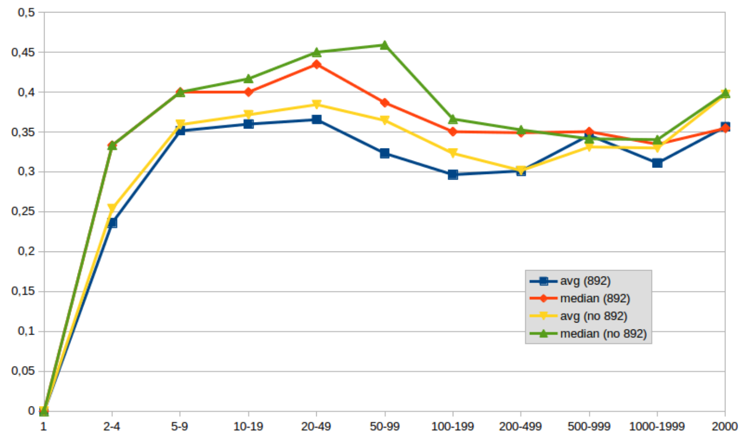


Figure 5.9: Ratio operator count to query length

5.3.3 Co-occurrence of operators

On average, 1.7 different *types* of operators are used in search queries where at least one Boolean operator is present (see Table 5.13). For example, consider the two queries "30/$.ccls. AND (clip$4 OR clamp$4 OR contain$4 OR canist$4) WITH razor WITH cream" and "clipping SAME window AND processing WITH frequency". In both queries, three different *types* of Boolean operators appear.

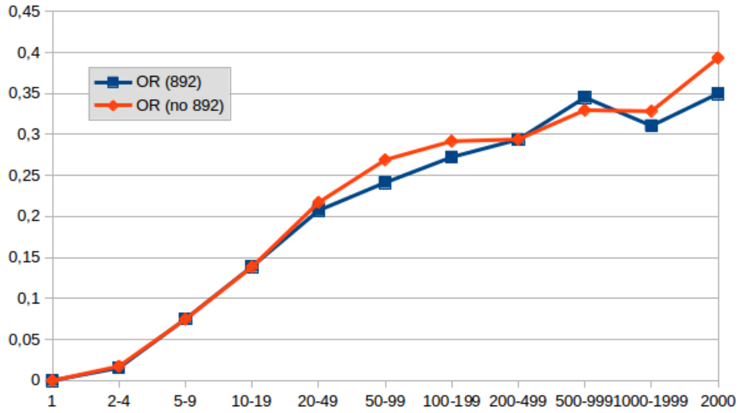


Figure 5.10: Ratio OR operator count to query length

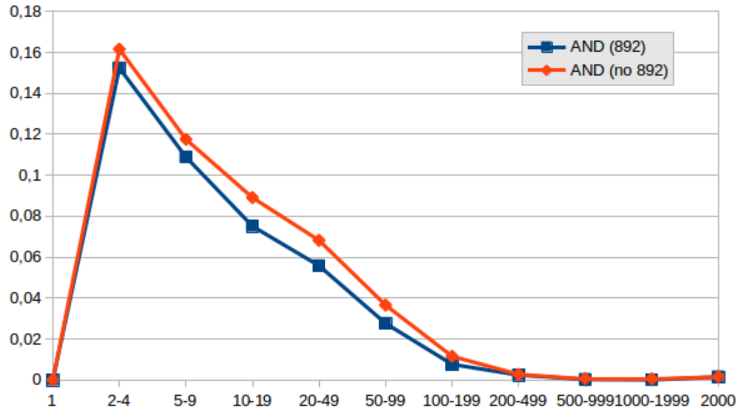


Figure 5.11: Ratio AND operator count to query length

Table 5.13: Co-occurrences of different types of operators.

	$SRNT_{892}$	$SRNT_{no892}$
avg	1.7586	1.7205
avg of max	2.8547	2.5731
avg of min	1.1105	1.1908
avg of median	1.6657	1.6479
avg of variance	0.4723	0.3800

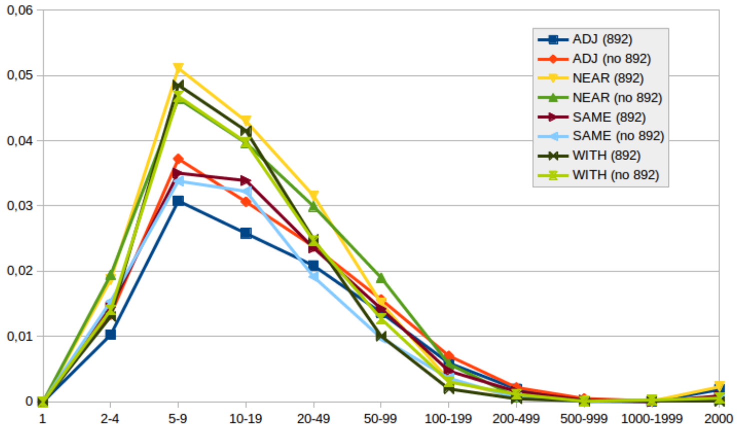


Figure 5.12: Ratio ADJ, NEAR, SAME, WITH operator counts to query length

In both document sets, the most often co-occurring Boolean operators within a single query are (AND, OR), (AND, NEAR), (AND, WITH) for $n = 2$ (Table 5.14), (AND, NEAR, OR), (AND, WITH, OR), (AND, OR, SAME), (AND, ADJ, OR) for $n = 3$ (Table 5.15), n being the number of different Boolean operator types appearing in a query.

In a slightly higher percentage of queries in $SRNT_{no892}$ no Boolean operator is used at all (29.06% to 27.48% in $SRNT_{892}$). In around 35% of all queries in both sets exactly one type of operator is used. In $SRNT_{892}$ co-occurring types of Boolean operators appear slightly more often (see Table 5.16, Figure 5.13).

5.3.4 Use of prescribed words

NEAR and SAME support the prescription of a proximity n, with n being the range of words two terms must appear within in a document for the document to be considered in the result. Specifying prescribed words is common for the NEAR operator. In only 20% respectively 23% of all uses of NEAR, n is not provided. In 25% of all uses of NEAR a proximity of $n = 2$ is specified.

SAME is seldom used in conjunction with a specification of prescribed words. In 98.97% ($SRNT_{892}$) respectively 99.57% ($SRNT_{no892}$) of all occurrences no proximity is provided (see Table 5.17).

5.4 Use of Query Expansion

Patent searchers at the USPTO follow two approaches in how Query Expansion is applied. In the first approach a truncation operator ("\$n") is used, where n specifies the number of

Table 5.14: Co-occurring Boolean operators in queries where exactly two different types of operator appear ($n = 2$).

operators	% of all queries ($SRNT_{892}$)	% of all queries ($SRNT_{no892}$)
(AND, OR)	17.57%	17.14%
(AND, NEAR)	14.45%	14.51%
(AND, WITH)	11.58%	12.45%
(NEAR, OR)	9.47%	7.61%
(AND, ADJ)	9.09%	12.56%
(AND, SAME)	7.07%	6.66%
(WITH, OR)	5.86%	4.84%
(OR, SAME)	4.19%	3.57%
(NEAR, SAME)	3.80%	3.43%
(NEAR, WITH)	3.69%	3.25%
(ADJ, WITH)	2.94%	3.02%
(ADJ, OR)	2.72%	2.96%
(ADJ, SAME)	2.42%	2.73%
(WITH, SAME)	2.35%	2.12%
(NEAR, ADJ)	1.56%	1.63%
sum of co-occurrences with less than 1%	1.15%	1.44%

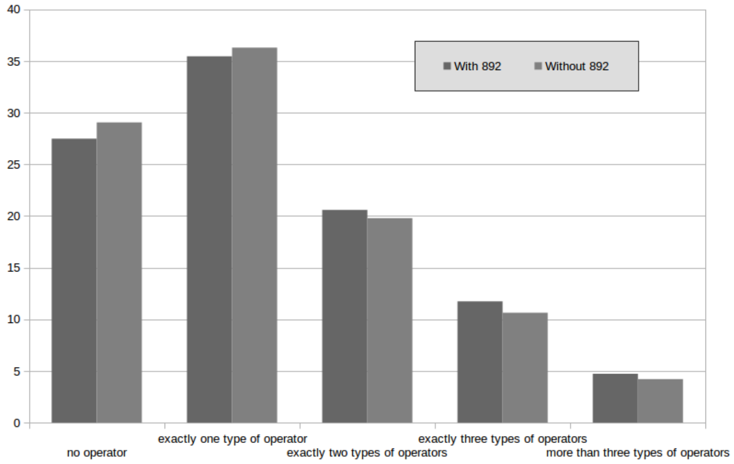


Figure 5.13: Co-occurrences of different types of Boolean operators. (similar results for $SRNT_{no892}$

Table 5.15: Co-occurring Boolean operators in queries where exactly three different types of operator appear ($n = 3$).

operators	% of all queries ($SRNT_{892}$)	% of all queries ($SRNT_{no892}$)
(AND,NEAR,OR)	15.95%	16.36%
(AND,WITH,OR)	14.07%	13.29%
(AND,OR,SAME)	8.71%	7.16%
(AND,ADJ,OR)	7.71%	10.84%
(NEAR,OR,SAME)	5.68%	4.61%
(AND,NEAR,WITH)	5.64%	5.58%
(AND,NEAR,SAME)	5.23%	4.91%
(NEAR,WITH,OR)	5.01%	3.67%
(AND,ADJ,WITH)	4.09%	5.07%
(AND,WITH,SAME)	3.44%	3.44%
(WITH,OR,SAME)	3.31%	2.65%
(AND,NEAR,ADJ)	3.27%	4.08%
(AND,ADJ,SAME)	3.20%	3.74%
(ADJ,WITH,OR)	2.93%	3.00%
(ADJ,OR,SAME)	2.71%	2.80%
(NEAR,ADJ,OR)	2.39%	2.39%
(NEAR,WITH,SAME)	1.63%	1.42%
(NEAR,ADJ,SAME)	1.22%	1.25%
(ADJ,WITH,SAME)	1.07%	1.00%
(ADJ,NEAR,WITH)	1.06%	1.00%
sum of co-occurrences with less than 1%	1.58%	1.64%

Table 5.16: Shares of queries with exactly n different Boolean operator types.

number n of operator types in query	% of all queries in $SRNT_{892}$	% of all queries in $SRNT_{no892}$
(no operators given)	27.48%	29.06%
$n = 1$	35.45%	36.29%
$n = 2$	20.59%	19.79%
$n = 3$	11.73%	10.63%
$n > 3$	4.72%	4.21%

Table 5.17: Use of prescribed words n for NEAR, SAME.

NEAR	$SRNT_{892}$	$SRNT_{no892}$	**SAME**	$SRNT_{892}$	$SRNT_{no892}$
$n = 0$ (not specified)	20.98%	23.85%		98.97%	99.5786
$n = 1$	1.83%	1.21%		0.02%	0.006%
$n = 2$	25.10%	22.74%		0.44%	0.15%
$n = 3$	20.11%	21.08%		0.25%	0.10%
$n = 4$	8.68%	9.58%		0.06%	0.03%
$n = 5$	14.30%	11.82%		0.09%	0.03%
$n > 5$	8.96%	9.69%		0.13%	0.08%

chars for finding term variations. For example, "observ$3" considers documents containing the term "observing" but not "observation". The second approach is to supply a list of alternate keywords encapsulated in parentheses. See Listing 5.2 and Table 5.18 for examples.

Listing 5.2: Example query with manual query expansion

```
(observ$3 identif$3 catch$3 discover$3 find$3)
near10
(similar$3 alikeness association$1 collation$1 similarit$3)
(select$3 pick$3 choos$3)
with
(channel$3)
with
(priority)
near5
(message content data)
and
(flight aircraft)
and
protocol
```

5.4.1 Truncation operator

At least *one* occurrence of a truncation operator is found in 73.71% of all documents in $SRNT_{892}$ and in 64.61% of all documents in $SRNT_{no892}$. In these documents, a truncation operator appears on average in 27.23% of queries in a document of $SRNT_{892}$ and in 26.2% of queries in a document of $SRNT_{no892}$. In queries with at least one occurrence of a truncation operator the average truncation operator count is 2.91 ($SRNT_{892}$; median 2.0) respectively 2.84 ($SRNT_{no892}$; median 2.0).

Table 5.18: Examples for two different types of Query Expansion applied by patent searchers, n being the count of lists of alternate terms respectively the count of truncation operators provided by the searcher.

Query	n of expansion lists	n of truncation operators
(car automobile auto)	1	0
(car OR motion)	1	0
(car AND motion)	0	0
("US12345" | "US23456")	0	0
("US12345" "US23456")	0	0
(car bicycle vehic$3)	0	1
(car NEAR bicycle)	0	0
(car (van pick truck (motor$3 inject$4)))	3	2

Co-occurrence with Boolean operators

In just a small fraction (0.6%) of all queries truncation is used but no Boolean operator. In around 25% of all queries, truncation and at least one Boolean operator co-occur. Slightly more queries (26.91% in $SRNT_{no982}$, 28.45% in $SRNT_{no892}$) use neither. In around 45% a Boolean but no truncation operator is used. Table 5.19 presents the truncation and Boolean operator: 1. queries that contain any Boolean operator and make use of truncation (op+trunc), 2. queries that contain any Boolean operator and make no use of truncation (op), 3. queries that make use of truncation but contain no Boolean operator (trunc), and 4. queries that neither contain any Boolean operator nor make use of truncation (none).

Table 5.19: Query classification based on operator use

	% op+trunc	% op	% trunc	% none
$SRNT_{892}$	26.86	45.61	0.60	26.91
$SRNT_{no892}$	25.01	45.87	0.64	28.45

In Figure 5.14 co-occurrences of different types of Boolean operators with the truncation operator are shown. In more than 50% of queries that use NEAR, SAME or WITH truncation is applied, too.

Prescribed number of characters

As already mentioned above, the truncation operator ("$n") allows the prescription of a number of characters for finding term variations. For example, "observ$3" would

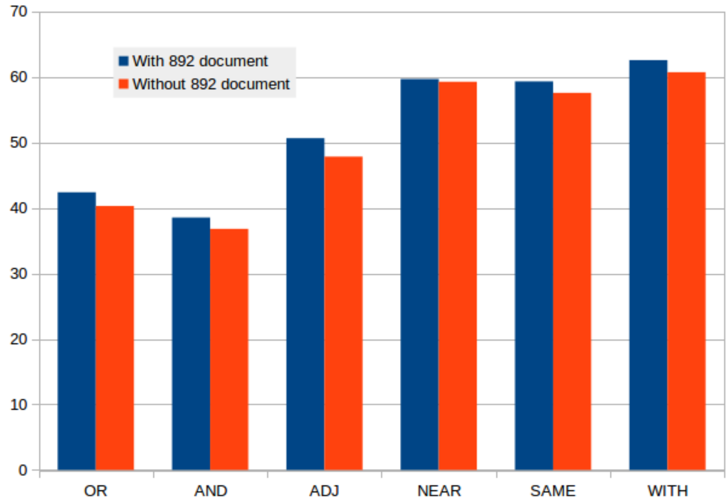


Figure 5.14: Co-occuring of Boolean operator and truncation operator.

consider the term "observing", but not "observation". In nearly 70% of all occurences the prescribed numbers of characters is specified with either 3 or 4 (see Figure 5.15).

5.4.2 Supplying lists of alternate keywords

40.76% of all documents of $SRNT_{892}$ and 42.05% of all documents of $SRNT_{no892}$ contain at least one query where the searcher has supplied a list of alternate keywords. In these documents, lists of alternate keywords are supplied in an average of 45.31% ($SRNT_{892}$) and 49.52% ($SRNT_{no892}$) of queries (see Table 5.20).

Table 5.20: % of queries in which lists of alternate terms are supplied (in documents where at least one query provides alternate lists)

	$SRNT_{892}$	$SRNT_{no892}$
avg	45.31%	49.52%
avg of median	39.98%	044.53%

On average, 2.5 (median: 2.0) of expansion lists are provided in queries with at least one expansion list. 3.5 (median: 3.0) expansion terms are provided for each list by the searcher. In addition, the truncation operator is used in 31.05% ($SRNT_{892}$) respectively 30.48% ($SRNT_{no892}$) of all supplied expansion lists.

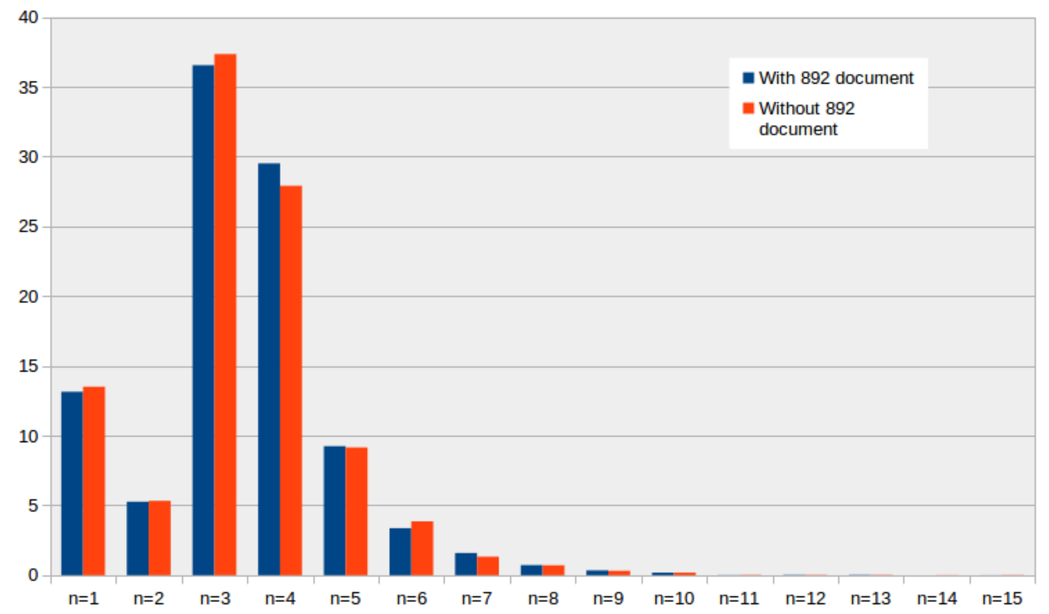


Figure 5.15: Prescribed truncation characters (% to total of all occurrences of the truncation operator)

5.4.3 Summary

In this chapter the results of the examination of more than 500,000 search query logs with a total of more than 15,000,000 Boolean queries obtained from the USPTO have been presented. The dataset was split into two classes ($SRNT_{892}$, $SRNT_{no892}$), based on the presence of an 892 document for an SRNT document.

The examined SRNT documents contain an average of 30.88 queries (median 19.0). The average query length is 9.3 (median 5.0). The most often searched patent database fields are patent classification ("ccls"), patent number ("pn") and claims ("clm"). The fields "ccls" and "clm" appear considerably more often searched in queries of documents of set $SRNT_{no892}$.

On average, 4.5 different databases are searched by a query. The most popular database is USPAT, which is searched in by nearly 100% of all queries. The least popular database is FPRS.

References (to address past queries) are used in considerably more documents of $SRNT_{892}$ (58.50% to 45.77% of $SRNT_{no892}$).

The most popular Boolean operators are OR (roughly 46%) and AND (roughly 22% of all operators); XOR and NOT are seldom used (less than 1% of all operators); ADJ, NEAR, SAME or WITH are used with similar frequency. In roughly 71% of all search queries at least one Boolean operator is present. AND is used at least once in slightly more than 40% of all queries, OR in roughly 28%.

In queries with at least one Boolean operator 1.7 different types of operators are used on average.

Use of Query Expansion

At least one occurrence of a truncation operator is found in 73.71% ($SRNT_{892}$) and in 64.23% ($SRNT_{no892}$) of all documents. In more than a quarter of all queries of these documents a truncation operator appears. In less than 1% of all queries truncation is used but no Boolean operator. In roughly 70% of all usages the number of characters specified by the truncation operator is 3 or 4.

Lists of alternate terms appear in 45.31% documents of $SRNT_{892}$ and 49.52% of documents of $SRNT_{no892}$. In these documents, lists of alternate terms appear in nearly the half of all search queries. On average, the searcher provides 3.5 alternate keywords per expansion list.

CHAPTER 6

Conclusions

In the following, the questions posed in the first chapter of this thesis are answered.

6.1 Results

How many search queries do SRNT documents contain on average? On average, SRNT documents contain 30.88 Boolean search queries. For documents in set $SRNT_{892}$ ("successful" searches) the average length is 36.18, for documents in set $SRNT_{no892}$ ("unsuccessful" searches) the average length is 25.63.

How do patent examiners make use of query expansion? How many synonyms and related terms are used on average to expand a query term? Professional patent searchers are trained in the mechanics of patent searching. Therefore it is no surprise that Query Expansion (be it the use of the truncation operator, be it the supplement of lists of alternate terms) is, in general, thoroughly used: in at least one query of around 40% of all SRNT documents lists of alternate terms are provided. In nearly half of all queries of these documents lists of alternate terms are provided. On average, 3.5 expansion terms are allocated by the searcher.

Are searches with a thorough use of query expansion more likely to find relevant patent documents? Does the number of used synonyms and related terms influence the finding of relevant patent documents? No. The hypothesis that "successful" searches (in the sense that relevant patent documents have been detected) make more frequent use of expansion lists can not be supported at all.

Expansion lists are provided to Boolean search queries with similar quantity by patent searchers in both document sets.

However, we are not able to make an assumption regarding the *quality* of the provided lists, i.e. regarding missing relevant terms. On the other hand, one should note that truncation operators, a popular way of instructing the search engine to perform a rather naive form of QE, are found in more documents of "successful" searches. They appear in nearly 75% of all documents of "successful" and in 64% of all documents of "unsuccessful" searches; in these documents, truncation operators are used on average in more than a quarter of all search queries.

How popular are the various Boolean operators (AND, OR, NEAR, WITH, SAME, ...)? As expected, OR and AND are the most popular Boolean operators; NOT and especially XOR are rarely used. OR is the only operator to usually also appear in long or very long queries. Other findings, such as the ratio of digits to all characters of a query, suggest that very long queries are usually searches for different patent numbers or patent classes. In these queries, keywords are usually simply connected with the OR operator. Roughly 30% of all used operators are either ADJ, NEAR, SAME, WITH. AND is present in more *queries* than any other operator, altough the total count for OR over the whole dataset is higher. Nearly 50% of all used operators are OR operators.

How do various parameters (e.g. average query length, usage of operators) of search sessions and queries differ between $SRNT_{892}$ ("successful searches") and $SRNT_{no892}$ ("unsuccessful searches")? Do they differ at all? Most of the examined features of SRNT documents and Boolean search queries (such as the average query length, the use of parentheses, the nesting level, the number of patent databases searched, quantity and popularity of different types of Boolean operators, etc.) yield very similar results for both document classes.

For few characteristics more pronounced differences were found. These are:

Average SRNT document length The average amount of queries is higher for "succesful" searches (36.18 in documents of group $SRNT_{892}$ and 25.63 in documents of group $SRNT_{no892}$).

EAST specific search fields The claims field ('.clm.') is used far less often in "successful" searches, where it is used in only 2.72% of all search queries compared to 10.29% of all queries of "unsuccessful" searches. The classification field ('.ccls.') is used in 12.43% of all queries of "successful" searches and in 17.13% of all queries of "unsuccessful" searches.

Use of references The use of references to address former queries is more popular in "successful" searches. In 58.50% of all documents of this group at least one query contains a reference. 45.77% of all "unsuccessful" search documents contain at least one query where a past query is reused.

6.2 Accuracy of the results

The underlying data was obtained using OCR. Thus, errors in the dataset are unavoidable. It is impossible to manually verify and correct such a massive amount of data. Certain grave errors (e.g. the "merge" of succeeding queries) are detected automatically. Affected documents have been removed from the dataset. It is assumed, based on observations, that all of these errors appear with similar frequency in both document sets.

6.3 Summary

This thesis has established the assumption that QE is a popular and frequently used strategy within the realm of patent searching. Patent searchers either provide lists of alternate terms or use the truncation operator (to consider term variations). The expectation that successful searches (during which relevant patent literature is discovered) make stronger use of expansions lists was not fulfilled. The truncation operator, on the other hand, is more widely used in successful searches.

Many of the examined features are present with similar frequency in successful and unsuccessful patent searches. More pronounced differences were found in the use of references (to include former queries in a new query) and in the use of specific patent database search fields. For example, the claims field is searched far less often in successful searches.

In general, the differences between Boolean queries of successful and unsuccessful searches are small.

6.4 Outlook and future work

QE in the field of professional patent searching is applied manually by the searcher. The question if automatic QE (performed by the search engine) or even semi-automatic QE (e.g. by presenting alternate terms to the searcher which he might select for inclusion) could be of benefit to a searcher has not been answered by this thesis. A fully automatic approach might be problematic: most studies regarding QE target standard web searches, were only one or two search terms are provided by the user - it has been shown in this thesis that Boolean queries issued by professional patent searchers are considerably longer. Hence, adding terms automatically (and hidden for the searcher) might lead to undesirable effects.

There are, of course, many other factors that influence the outcome of a search, such as the time spent investigating the results, the technological field and it's age, etc. These factors have not been considered in this thesis but should be investigated in future research. Future studies should also consider detecting specific search strategies (e.g. the use of narrowing and expanding queries), i.e. considering differences in a series of queries issued within a search.

APPENDIX A

Using PyPAIR

The provided application encompasses all steps necessary to assemble a dataset of Boolean search queries from SRNT documents. A range of application parameters can be configured in the provided configuration file (*config.py*).

A.1 Preparing the download

The first step is to generate a download file that contains a list of patent applications to be retrieved. For this task, Gsutil is used. The command given in Listing A.1 creates a text file (see Listing A.2) in which each line contains the trailing part of the URL to a downloadable patent application.

Listing A.1: Generation of a download file.

```
gsutil ls -l gs://uspto-pair/applications/126000* > download.txt
```

Listing A.2: Example for the content of a generated download file.

```
uspto-pair/applications/12600072.zip
uspto-pair/applications/12600073.zip
uspto-pair/applications/12600074.zip
uspto-pair/applications/12600075.zip
uspto-pair/applications/12600076.zip
uspto-pair/applications/12600077.zip
uspto-pair/applications/12600080.zip
uspto-pair/applications/12600081.zip
uspto-pair/applications/12600082.zip
```

The configuration file allows to specify the base URL from which patent applications are retrieved (Listing A.3):

Listing A.3: Download specific configurations

```
# The base url for pair download.
# (The full path for a given app number (e.g. "1234567")
# looks like: http://storage.googleapis.com/uspto-pair/applications/1234567.
    zip)
url = 'http://storage.googleapis.com/uspto-pair/applications/'

download_path = '/var/data/srnt/'
download_file = '/home/roland/srnt/retrieval.txt'
```

Download__path specifies the directory under which extracted SRNT and 892 documents will be stored; download__file is the text file generated above (Listing A.1).

A.2 Retrieving patent applications.

The retrieval process is started by invoking *download.py*. Every patent application listed within the download file is retrieved as ZIP file, from which all SRNT and 892 documents are extracted and moved to subdirectories of the download path (see A.4). PyPAIR takes care that not more than around 1000 files are stored in the same directory. Indexed subdirectories are created whenever necessary. Finally, the ZIP file is deleted from the filesystem and it's URL is removed from the download file.

Listing A.4: Download folder hierarchy

```
pypair\download\SRNT\1\SRNT-0001.pdf
pypair\download\SRNT\1\SRNT-0002.pdf
...
pypair\download\SRNT\1\SRNT-999.pdf
pypair\download\SRNT\1\SRNT-1000.pdf
pypair\download\SRNT\2\SRNT-1001.pdf
pypair\download\SRNT\2\SRNT-1002.pdf
...
pypair\download\892\1\892-0001.pdf
pypair\download\892\1\892-0002.pdf
...
```

A.3 Transforming SRNT documents into text files

The transformation of SRNT documents is started by invoking *pypair.py*.

PyPair uses Ghostscript to split a PDF document into a series of TIFF images (Listing A.5). Ghostscript supports various drivers for controlling the output quality. The output driver *tiffg4* (G4 fax encoding, creates black-and-white output) in combination with a resolution of 400x400 dpi was found to yield good results for succeeding tasks.

Listing A.5: Command to split a PDF document into a series of TIFF images.

```
    cmd = "gs -sDEVICE=tiffg4 -r400x400 -dBATCH -dNOPAUSE " +\
          "-sOutputFile=" + pdf + ".page_%d.tiff " + pdf
```

Above command creates one TIFF image for every page of the PDF document (see image 4.3).

Hough transformation is applied to detect table borders. To support Hough transformation, a Canny edge detector is applied to the image beforehand. Parameters for both algorithms[1] can be adjusted in the configuration (see Listing A.6).

Listing A.6: Parameters and their default values for Canny edge detection and probabilistic Hough transform.

```
canny = {
    "threshold1": 80,
    "threshold2": 120,
    "aperture_size": 3
}
hough = {
    "rho": 1,
    "theta": (math.pi / 2),
    "threshold": 2,
    "minLineLength": 100,
    "maxLineGap": 5
}
```

The maxLineGap configuration is used to ignore small gaps in lines; minLineLength introduces a limit in order to prevent that very short lines (e.g. lines that are part of letters) are detected.

PyPair uses Tesseract for OCR.

Listing A.7: Applying Tesseract to the date column

```
tesseract -psm 6  + input.tiff + output -f hocr pairdate
```

[1] See http://docs.opencv.org/modules/imgproc/doc/feature_detection.html#canny and http://docs.opencv.org/modules/imgproc/doc/feature_detection.html#houghlinesp

The parameter psm specifies the page segmentation mode. A page segmentation mode of 6 instructs Tesseract to assume a single uniform block of text. The parameter "-f hocr" instructs Tesseract to output it's result in form of a HTML file. HOCR stands for HTML OCR. In addition to the recognized text the file will contain layout information, i.e. the coordinates of detected chars, lines, paragraphs. The pairdate whitelist is used in order to decrease the risk of recognition errors such as interpreting letter *O* instead of the numeric 0, or *l* instead of 1. For an example of the resulting HTML file see listing A.8.

Listing A.8: Excerpt of the content of a HTML file generated by Tesseract.

```
<body>
  <div class='ocr_page' id='page_1' title='image "/home/honeder/pypair/test/
      tmp.tiff"; bbox 0 0 433 1583; ppageno 0'>
    <div class='ocr_carea' id='block_1_1' title="bbox 0 0 433 1583">
      <p class='ocr_par' dir='ltr' id='par_1' title="bbox 10 0 410 1390">
        <span class='ocr_line' id='line_1' title="bbox 10 0 410 82"><span
            class='ocrx_word' id='word_1' title="bbox 10 0 410 82"></span></
            span>
        <span class='ocr_line' id='line_2' title="bbox 10 186 282 226"><span
            class='ocrx_word' id='word_2' title="bbox 10 186 282
            226">2010/10/19</span></span>
        ...
      </p>
    </div>
  </div>
</body>
```

Listing A.9: ImageMagick command for improving image quality for OCR

```
convert image.tiff -write MPR:source -morphology close rectangle:1x3
         -clip-mask MPR:source
         -morphology erode:8 square
         +clip-mask -bordercolor White -border 100x100 input.tiff
```

A.4 Data normalization

By invoking *normalize.py* the processed dataset is validated and normalized. Fields of known ranges or types of values (e.g. the "plurals" field, the "hits" field, the date and time fields and the list of databases searched) are validated for every query. Invalid values are removed.

A.5 Calculating quality indicators and cleaning the dataset

By invoking *indicators.py* a CSV file ("indicators.csv") is created. This file contains a list of all parsed SRNT documents with their calculated quality indicators. By invoking *goodset.py*, finally, all log files from suspected badly processed SRNT documents (based on their quality indicators) are removed from the dataset.

Bibliography

[ACR04] Giambattista Amati, Claudio Carpineto, and Giovanni Romano. “Query difficulty, robustness, and selective application of query expansion”. In: *Advances in Information Retrieval* 10.X (2004), pp. 127–137.

[AIP13] AIPLA. *Report of the Economic Survey 2013*. Tech. rep. 2013.

[AR97] Alan Aronson and Thomas Rindflesch. “Query expansion using the UMLS Metathesaurus.” In: *Proceedings: a conference of the American Medical Informatics Association* (1997), pp. 485–489.

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. “Fast Algorithms for Mining Association Rules in Large Databases”. In: *Journal of Computer Science and Technology* 15.6 (1994), pp. 487–499. ISSN: 1000-9000. DOI: 10.1007/BF02948845. URL: http://portal.acm.org/citation.cfm?id=645920.672836.

[BBAG05] Mustapha Baziz, Mohand Boughanem, and Nathalie Aussenac-Gilles. “Conceptual Indexing Based on Document Content Representation”. In: *Context: Nature, Impact, and Role. Lecture Notes in Computer Science* 3507 (2005).

[Bel13] Alexander M. Bell. “An Autopsy on Submarine Patents: A Window into Expectations of the World Technological Frontier”. PhD thesis. Brown University, 2013.

[Ber+08] Andrea Bernardini et al. “FUB at TREC 2008 Relevance Feedback Track: Extending Rocchio with Distributional Term Analysis”. In: *Processing* (2008), pp. 1–9. ISSN: 1048776X.

[Bil+03] Bodo Billerbeck et al. “Query Expansion using Associated Queries”. In: *CIKM ’03 Proceedings of the twelfth international conference on Information and knowledge management.* 2003, pp. 2–9.

[BL99] Adam Berger and John Lafferty. “Information retrieval as statistical translation”. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’99* (1999), pp. 222–229. ISSN: 09042512. DOI: 10.1145/312624.312681. URL: http://portal.acm.org/citation.cfm?doid=312624.312681.

[BLN04] Gely Basharin, Amy Langville, and Valeriy Naumov. "The Life and Work of A . A . Markov". In: *Linear Algebra and its ...* (2004), pp. 1–22. ISSN: 00243795. DOI: 10.1016/j.laa.2003.12.041. URL: http://www.sciencedirect.com/science/article/pii/S0024379504000357.

[BM05] H. Bast and D. Majumdar. "Why spectral retrieval works". In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (2005), p. 18. DOI: 10.1145/1076034.1076040. URL: http://portal.acm.org/citation.cfm?id=1076034.1076040.

[BMS07] J. Bhogal, A. Macfarlane, and P. Smith. "A review of ontology based query expansion". In: *Information Processing and Management* 43.4 (2007), pp. 866–886. ISSN: 03064573. DOI: 10.1016/j.ipm.2006.09.003.

[BMW07] Holger Bast, D Majumdar, and Ingmar Weber. "Efficient interactive query expansion with complete search". In: *Proceedings of the sixteenth ACM conference on information and knowledge management* (2007), pp. 857–860. DOI: 10.1145/1321440.1321560. URL: http://dl.acm.org/citation.cfm?id=1321560.

[Cao+08] Guihong Cao et al. "Selecting good expansion terms for pseudo-relevance feedback". In: *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '08* (2008), p. 243. DOI: 10.1145/1390334.1390377. URL: http://portal.acm.org/citation.cfm?doid=1390334.1390377.

[CCH92] J.P. Callan, W.B. Croft, and S.M. Harding. "The INQUERY retrieval system". In: *Proceedings of the third international conference on database and expert systems applications* (1992), pp. 78–83.

[CH89] Kenneth Ward Church and Patrick Hanks. "Word Association Norms, Mutual Information, and Lexicography". In: *Proceedings of the 27th Annual Conference of the Association for Computational Linguistics* 16.1 (1989), pp. 22–29. ISSN: 08912017. DOI: 10.3115/981623.981633.

[CR12] Claudio Carpineto and Giovanni Romano. "A Survey of Automatic Query Expansion in Information Retrieval". In: *ACM Computing Surveys* 44.1 (2012), pp. 1–50. ISSN: 03600300. DOI: 10.1145/2071389.2071390.

[Cro88] C.J. Crouch. "A cluster-based approach to thesaurus construction". In: *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval* (1988), pp. 309–320.

[CTC02] Steve Cronen-Townsend and W Bruce Croft. "Quantifying query ambiguity". In: *Proceedings of the second international conference on Human Language Technology Research -* (2002), p. 104. DOI: 10.3115/1289189.1289266. URL: http://portal.acm.org/citation.cfm?doid=1289189.1289266.

[CTZC04] Steve Cronen-Townsend, Yun Zhou, and W Bruce Croft. “A Framework for Selective Query Expansion”. In: *Proceedings of the thirteenth ACM international conference on information and knowledge management* 3 (2004), pp. 236–237. URL: https://doi.org/10.1145/1031171.1031220.

[Cui+02] Hang Cui et al. “Probabilistic Query expansion using query logs”. In: *11th International Conference on World Wide Web (WWW‘02)* (2002), pp. 325–332. DOI: 10.1145/511446.511489.

[CY92] Carolyn J Crouch and Bokyung Yang. “Experiments in Automatic Statistical Thesaurus Construction”. In: *15th Annual International SIGIR* (1992), pp. 77–88. DOI: 10.1145/133160.133180.

[Dam94] K W Dam. “The economic underpinnings of patent law”. In: *The Journal of Legal Studies* 23.1 (1994), pp. 247–271. ISSN: 0047-2530. DOI: 10.1086/467923.

[DH72] R O Duda and P E Hart. “Use of the Hough transform to detect lines and curves in pictures”. In: *Communications of the Association Computing Machinery* 15.1 (1972), pp. 11–15. ISSN: 00010782. DOI: 10.1145/361237.361242.

[DSG02] H Drucker, B Shahrary, and D C Gibbon. “Support vector machines: relevance feedback and information retrieval”. In: *Information processing & management* 38 (2002), pp. 305–323. ISSN: 03064573. DOI: 10.1016/S0306-4573(01)00037-1.

[FD97] Larry Fitzpatrick and Mei Dent. “Automatic Feedback Using Past Queries: Social Searching?” In: *SIGIR ’97 Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval.* 1997, pp. 306–313.

[FJA05] Gaihua Fu, C. B. Jones, and Alia I. Abdelmoty. “Ontology Based Spatial Query Expansion in Information Retrieval”. In: *Lecture Notes in Computer Science* (2005), pp. 1466–1482.

[FKS07] GS Ford, T Koutsky, and LJ Spiwak. “Quantifying the cost of substandard patents: some preliminary evidence”. In: *Phoenix Center Policy Paper* 30 (2007).

[GCH05] Zhiguo Gong, Chan Wa Cheang, and Leong Hou U. “Web query expansion by WordNet”. In: *Database and Expert Systems Applications* (2005), pp. 166–175. DOI: 10.1007/11827405_37. URL: http://www.springerlink.com/index/6MV29D0K19BCCKMT.pdf.

[GJ62] Vincent E Giuliano and Paul E Jones. “Linear associative information retrieval”. In: *In Studies for the design of an English command and control language system, ESD-TR-62-294, Arthur D. Little, Inc., Cambridge, MA* (1962), pp. 1–42. URL: papers3://publication/uuid/15BAED85-7A03-4B5B-984B-9F7EF6ECD860.

[GLJ11] Debasis Ganguly, Johannes Leveling, and Gareth J F Jones. "Query expansion for language modeling using sentence similarities". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6653 LNCS (2011), pp. 62–77. ISSN: 03029743. DOI: 10.1007/978-3-642-21353-3{_}6.

[GOS03] Nicola Guarino, Daniel Oberle, and Steffen Staab. "What Is an Ontology?" In: *Handbook on Ontologies*. Berlin, Heidelberg: Springer, 2003. DOI: 10.1007/978-3-540-92673-3.

[GW06] F.A. Grootjen and Th.P. van der Weide. "Conceptual query expansion". In: *Data & Knowledge Engineering* 56.2 (2006), pp. 174–193. ISSN: 0169023X. DOI: 10.1016/j.datak.2005.03.006. URL: http://linkinghub.elsevier.com/retrieve/pii/S0169023X05000376.

[GWR99] Susan Gauch, Jianying Wang, and Satya Mahesh Rachakonda. "A corpus analysis approach for automatic query expansion and its extension to multiple databases". In: *ACM Transactions on Information Systems* 17.3 (1999), pp. 250–269. ISSN: 10468188. DOI: 10.1145/314516.314519.

[Har91] Donna Harman. "How effective is suffixing?" In: *Journal of the American Society for Information Science* 42.1 (1991).

[HCO03] Chien-kang Huang, Lee-feng Chien, and Yen-jen Oyang. "Relevant Term Suggestion in Interactive Web Search Based on Contextual Information in Query Session Logs". In: *Journal of the American Society for Information Science and Technology* 54.7 (2003), pp. 638–649.

[HCS00] AL Houston, H Chen, and BR Schatz. "Exploring the use of concept spaces to improve medical information retrieval". In: *Decision Support ...* (2000). URL: http://www.sciencedirect.com/science/article/pii/S016792360000097X.

[HDG06] Jiani Hu, Weihong Deng, and Jun Guo. "Improving Retrieval Performance by Global Analysis". In: *18th International Conference on Pattern Recognition (ICPR'06)* (2006), pp. 703–706. DOI: 10.1109/ICPR.2006.703. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1699302.

[Hie98] Djoerd Hiemstra. "A Linguistically Motivated Probabilistic Model of Information Retrieval". In: *Research and Advanced Technology for Digital Libraries* (1998), pp. 515–515.

[HNR07] David Hunt, Long Nguyen, and Matthew Rodgers. *Patent Searching: Tools & Techniques*. Wiley, 2007. ISBN: 9780471783794.

[HPD00] W Hersh, S Price, and L Donohoe. "Assessing thesaurus-based query expansion using the UMLS Metathesaurus." In: *Proceedings / AMIA ... Annual Symposium. AMIA Symposium* (2000), pp. 344–348. ISSN: 1531-605X.

[Huf08] Claudia Huff. “Query Difficulty for Digital Libraries”. In: *DLib Magazine* 15.9/10 (2008). ISSN: 10829873. DOI: 10.1045/september2009-varalakshmi. URL: http://webdoc.sub.gwdg.de/edoc/aw/d-lib/dlib/september09/varalakshmi/09varalakshmi.html.

[Hul96] David A. Hull. “Stemming Algorithms: A Case Study for Detailed Evaluation”. In: *Journal of the American Society for Information Science* 47 (1996). ISSN: 1098-6596.

[JB08] Vahid Jalali and Mohammad Reza Matash Borujerdi. “The effect of using domain specific ontologies in query expansion in medical field”. In: *2008 International Conference on Innovations in Information Technology, IIT 2008* (2008), pp. 277–281. DOI: 10.1109/INNOVATIONS.2008.4781679.

[Jiv11] Anjali Ganesh Jivani. “A Comparative Study of Stemming Algorithms”. In: *Int. J. Comp. Tech. Appl.* 2.2011 (2011), pp. 1930–1938. ISSN: 2229-6093.

[JRM06] Rosie Jones, Benjamin Rey, and Omid Madani. “Generating query substitutions”. In: *WWW ’06 Proceedings of the 15th international conference on World Wide Web*. 2006, pp. 387–396. ISBN: 1595933239. DOI: 10.1145/1135777.1135835.

[Kak12] Yogesh Kakde. “A Survey of Query Expansion until June 2012”. In: *ACM Computing Surveys* 44.1 (2012).

[Kro93] Robert Krovetz. “Viewing Morphology as an Inference Process”. In: *SIGIR ’93 Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* (1993), pp. 191–202.

[KZ04] Reiner Kraft and Jason Zien. “Mining Anchor Text for Query Refinement”. In: *WWW ’04 Proceedings of the 13th international conference on World Wide Web*. 2004, pp. 666–674. ISBN: 158113844X.

[Lar09] Ray R. Larson. “Introduction to Information Retrieval”. In: *Journal of the American Society for Information Science and Technology* (2009), n/a–n/a. ISSN: 15322882. DOI: 10.1002/asi.21234. URL: http://doi.wiley.com/10.1002/asi.21234.

[LC01a] Victor Lavrenko and Wb Croft. “Relevance-based language models: Estimation and analysis”. In: *Croft and Lafferty [2]* (2001), pp. 1–5. URL: http://boston.lti.cs.cmu.edu/callan/Workshops/lmir01/WorkshopProcs/Papers/lavrenko.pdf.

[LC01b] G. Leroy and H. Chen. “Meeting medical terminology needs - The ontology-enhanced Medical Concept Mapper”. In: *IEEE Transactions on Information Technology in Biomedicine* 5.4 (2001), pp. 261–270. ISSN: 10897771. DOI: 10.1109/4233.966101.

[Liu+04] Shuang Liu et al. “An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases”. In: *Proceeding SIGIR '04 Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* (2004), pp. 266–272. DOI: 10.1145/1008992.1009039.

[Lov68] Julie Beth Lovins. “Development of a stemming algorithm”. In: *Mechanical Translation and Computational Linguistics* 11.June (1968), pp. 22–31. URL: http://journal.mercubuana.ac.id/data/MT-1968-Lovins.pdf.

[LZ09] Yuanhua Lv and ChengXiang Zhai. “A comparative study of methods for estimating query language models with pseudo feedback”. In: *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09* 2.3 (2009), p. 1895. DOI: 10.1145/1645953.1646259. URL: http://portal.acm.org/citation.cfm?doid=1645953.1646259.

[LZ10] Yuanhua Lv and Chengxiang Zhai. “Positional Relevance Model for Pseudo-Relevance Feedback”. In: *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (2010), pp. 579–586. DOI: 10.1145/1835449.1835546.

[Lóp13] J Parapar López. “Relevance-based language models: new estimations and applications”. PhD thesis. Universidade da Coruna, 2013. URL: http://ruc.udc.es/handle/2183/10332.

[MC07] Donald Metzler and W Bruce Croft. “Latent concept expansion using markov random fields”. In: *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 07* 120.8 (2007), p. 311. DOI: 10.1145/1277741.1277796. URL: http://portal.acm.org/citation.cfm?doid=1277741.1277796.

[MFZ07] Qiaozhu Mei, Hui Fang, and ChengXiang Zhai. “A study of Poisson query generation model for information retrieval”. In: *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '07* (2007), p. 319. ISSN: 1595935975. DOI: 10.1145/1277741.1277797. URL: http://portal.acm.org/citation.cfm?doid=1277741.1277797.

[Mgb03] Ikechi Mgbeoji. “The Juridical Origins of the International Patent System: Towards a Historiography of the Role of Patents in Industrialization”. In: *Journal of the History of International Law* 5 (2003), pp. 403–422.

[Miz98] Stefano Mizzaro. “How many relevances in information retrieval?” In: *Interacting with Computers* 10.3 (1998), pp. 303–320. ISSN: 09535438. DOI: 10.1016/S0953-5438(98)00012-5.

[MLS99] David R. H. Miller, Tim Leek, and Richard M. Schwartz. "A hidden Markov model information retrieval system". In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99* (1999), pp. 214–221. ISSN: 1581130961. DOI: 10.1145/312624.312680. URL: http://portal.acm.org/citation.cfm?doid=312624.312680.

[MSB98] Mitra Mandar, Amit Singhal, and Chris Buckley. "Improving Automatic Query Expansion". In: *SIGIR '98 Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (1998), pp. 206–214.

[NHO05] Kristina Nilsson, Hans Hjelm, and Henrik Oxhammar. "SUiS - cross-language ontology-driven information retrieval in a restricted domain". In: *In Proceedings of the 15th NODALIDA conference* (2005), pp. 139–145.

[NV02] Roberto Navigli and Paola Velardi. "An Analysis of Ontology-based Query Expansion Strategies". In: *Information Retrieval* (2002), pp. 42–49.

[OBB82] Robert Oddy, N J. Belkin, and H M. Brooks. "Ask for Information Retrieval: Part I. Background and Theory". In: *Journal of Documentation* 38(2) (1982), pp. 61–71.

[Pat] *Patents*. URL: https://www.monticello.org/site/research-and-collections/patents (visited on 03/02/2016).

[PC98] Jay Ponte and Bruce Croft. "A Language Modeling Approach To Information Retrieval". In: *Proceedings of the 21st annual international ACM SIGIR Conference on Research and Development in Information Retrieval* (1998), pp. 275–81.

[Pol04] Nicola Polettini. "The Vector Space Model in Information Retrieval - Term Weighting Problem Local Term-Weighting". In: *Entropy* (2004), pp. 1–9.

[Por80] M.F. Porter. *An algorithm for suffix stripping.* 1980.

[PR07] Laurence Park and Kotagiri Ramamohanarao. "Query expansion using a collection dependent probabilistic latent semantic thesaurus". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining.* 2007, pp. 224–235.

[PW91] Helen J. Peat and Peter Willett. "The limitations of term co-occurrence data for query expansion in document retrieval systems". In: *Journal of the American Society for Information Science* 42.5 (1991), pp. 378–383.

[Qiu+93] Yonggang Qiu et al. "Concept based query expansion". In: *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '93* (1993), pp. 160–169. DOI: 10.1145/160688.160713. URL: http://portal.acm.org/citation.cfm?doid=160688.160713.

[RL03] Ian Ruthen and Mounia Lalmas. “A survey on the use of relevance feedback for information access systems”. In: *The Knowledge Engineering Review* 18.2 (2003), S0269888903000638.

[Roc71] J.J. Rocchio. “Relevance Feedback in Information Retrieval”. 1971.

[RSW99] SE Robertson, E Stephen, and S Walker. “Okapi/Keenbow at TREC-8.” In: *The Eigth Text Retrieval Conference (TREC-8)* (1999), pp. 151–162. URL: http://trec.nist.gov/pubs/trec8/papers/okapi.pdf.

[Sak00] Tetsuya Sakai. “A first step towards flexible local feedback for ad hoc retrieval”. In: *IRAL ’00 Proceedings of the fifth international workshop on on Information retrieval with Asian languages.* 2000, pp. 95–102. ISBN: 1581133006.

[SB88] Gerard Salton and C. Buckley. *Term-weighted approaches to automatic text retrieval.* 1988. DOI: 10.1016/0306-4573(88)90021-0. arXiv: 115.

[SBM11] Amit Singhal, Chris Buckley, and Mitra Mandar. *Pivoted Document Length Normalization.* Tech. rep. 2011, pp. 21–29. DOI: 10.1007/SpringerReference_101975.

[SC00] Greg Schohn and David Cohn. “Less is More: Active Learning with Support Vector Machines”. In: *ICML ’00 Proceedings of the Seventeenth International Conference on Machine Learning* (2000), pp. 839–846.

[SC99] Fei Song and W Bruce Croft. “A general language model for information retrieval”. In: *Information Retrieval* (1999), pp. 316–321. ISSN: 1581131461. DOI: 10.1145/319950.320022.

[Sha04] Carl Shapiro. “Patent System Reform: Economic Analysis and Critique”. In: *Berkeley Technology Law Journal* 19.3 (2004).

[Smi08] Barry Smith. “Ontology”. In: *The Blackwell Guide to the Philosophy of Computing and Information* (2008), pp. 153–166.

[SMK05] Tetsuya Sakai, Toshihiko Manabe, and Makoto Koyama. “Flexible pseudo-relevance feedback via selective sampling”. In: *ACM Transactions on Asian Language Information Processing* 4.2 (2005), pp. 111–135. ISSN: 15300226. DOI: 10.1145/1105696.1105699.

[SP97] Hinrich Schütze and Jan O. Pedersen. “A cooccurrence-based thesaurus and two applications to information retrieval”. In: *Information Processing & Management* 33.3 (1997), pp. 307–318. ISSN: 03064573. DOI: 10.1016/S0306-4573(96)00068-4.

[Spi11] Roland Spitzlinger. *On the Idea of Owning Ideas: the Philosophical Foundations of Intellectual Property.* VDM Verlag Dr. Müller, 2011. ISBN: 3639228014.

[TK01] Simon Tong and Daphne Koller. “Support Vector Machine Active Learning with Applications to Text Classification”. In: *Journal of Machine Learning Research* (2001), pp. 45–66. ISSN: 15324435. DOI: 10.1162/153244302760185243.

[TZ06] Tao Tao and ChengXiang Zhai. "Regularized estimation of mixture models for robust pseudo-relevance feedback". In: *Sigir* (2006), pp. 162–169. DOI: 10.1145/1148170.1148201. URL: http://portal.acm.org/citation.cfm?doid=1148170.1148201.

[Voo06] Ellen M. Voorhees. "The TREC 2005 robust track". In: *ACM SIGIR Forum* 40.1 (2006), p. 41. ISSN: 01635840. DOI: 10.1145/1147197.1147205.

[Wip12] Wipo. "World Intellectual Property Indicators 2012". In: *World Intellectual Property Organization* 1 (2012). ISSN: 01722190. DOI: 10.1016/0172-2190(79)90016-4. arXiv: 31.

[Wip14] Wipo. "World Intellectual Property Indicators 2014". In: *World Intellectual Property Organization* 1 (2014). ISSN: 01722190. DOI: 10.1016/0172-2190(79)90016-4. arXiv: 31.

[Won+08] W. S. Wong et al. "Re-examining the effects of adding relevance information in a relevance feedback environment". In: *Information Processing and Management* 44.3 (2008), pp. 1086–1116. ISSN: 03064573. DOI: 10.1016/j.ipm.2007.12.002.

[XC96] Jinxi Xu and W B Croft. "Query expansion using local and global document analysis". In: *SIGIR '96: Proceedings of ACM SIGIR Conference* 19 (1996), p. 4. ISSN: 01635840. DOI: 10.1145/243199.243202.

[XJW09] Yang Xu, Gareth J.F. Jones, and Bin Wang. "Query dependent pseudo-relevance feedback based on wikipedia". In: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '09* (2009), p. 59. DOI: 10.1145/1571941.1571954. URL: http://portal.acm.org/citation.cfm?doid=1571941.1571954.

[Xu+03] Zhao Xu et al. "A hybrid relevance-feedback approach to text retrieval". In: *Sebastiani F. (eds) Advances in Information Retrieval. ECIR 2003. Lecture Notes in Computer Science.* Vol. 2633. Berlin, Heidelberg: Springer, 2003, pp. 281–293.

[YC11] Sooyoung Yoo and Jinwook Choi. "Evaluation of term ranking algorithms for pseudo- relevance feedback in MEDLINE retrieval". In: *Healthcare Informatics Research* 17.2 (2011), pp. 120–130. ISSN: 20933681. DOI: 10.4258/hir.2011.17.2.120.

[Zaz+05] Ángel F. Zazo et al. "Reformulation of queries using similarity thesauri". In: *Information Processing and Management* 41.5 (2005), pp. 1163–1173. ISSN: 03064573. DOI: 10.1016/j.ipm.2004.05.006.

[ZL01] ChengXiang Zhai and John D Lafferty. "Model-based Feedback In The Language Modeling Approach To Information Retrieval". In: *Cikm* (2001), pp. 403–410.

[ZL04] Chengxiang Zhai and John Lafferty. “A study of smoothing methods for language models applied to information retrieval”. In: *ACM Transactions on Information Systems* 22.2 (2004), pp. 179–214. ISSN: 10468188. DOI: 10.1145/984321.984322.

[ZTL00] Jason Zien, John Tomlin, and Joy Liu. *Web Query Characteristics and their Implications on Search Engines*. Tech. rep. 2000.