Stockholm University

Department of Linguistics

A Vector Space analysis of Swedish patent claims, does decompounding help?

Linda Andersson

ABSTRACT

In the present study, a comparison between three different subsets of patent claims against an entire collection of 30,117 claims was performed in order to find similarities. A Vector Space Model was used as the retrieval model to calculate similarity between claims. The study was performed in a traditional laboratory environment. The Vector Space Model was evaluated with and without an additional morphological decompounding module. The decompounding module was implemented in two different ways (an exhaustive method and a more restricted method). The results indicate that decompounding will influence the performance of the retrieval model in a positive way. However, the sublanguage of patent claims and the errors made during the optical character recognition process were harmful towards the overall performance of the retrieval model.

D-Level Thesis
Advanced Course in Computational Linguistics
May 2009
Supervisor: Magnus Sahlgren

Table of Contents

Table of Figures

Table of Appendices

1. Introduction

The idea for this study was born on a trip to a birthday party for a relative of mine. During small talk in the car, I understood that my cousin Lisbeth Andersson worked as a patent engineer at the Swedish Patent and Registration Office. Since I am very interested in all sorts of information processing, I started to ask her about her experience of information seeking and search tools within her work. I learned that they use an advanced search tool with many search possibilities, such as a complex truncation function. After a while, we started to ponder upon developing an Information Retrieval system (IR system) that could automatically indicate if a new patent document was similar to others, as well as indicate the difference between the new documents and those that have been suggested to be similar. We agreed that such a system could very well be a useful complementary tool for the patent engineers working with the search process or the sifting process.

My idea was first to carry out a similarity comparison as well as a difference analysis of patent documents, an embryo to the IR system mentioned above. However, such a twofold analysis surpasses the reasonable scope of a thesis at the D-level and therefore I decided to narrow the analysis to cover only similarity comparison. Although the narrow scope, the task has offered a great deal of work due to the condition of the raw data, the large amount of the data (approximately 11,000,000 tokens) and the sparseness of linguistic tools for this type of domain (especially for Swedish). But I did not choose to narrow the scope further since I wanted to investigate the entire Information Retrieval process (i.e. from raw data to a full scale evaluation of an IR system).

Since Swedish morphology offers novel problems for Information Retrieval, the inclination towards morphological compounding being one of the main problems (Hedlund*, et al.* 2001), (Karlgren 2005), I chose to carry out my analysis with three test settings – one setting with the addition of a decompounding module at the pre-processing level provided by the parser (Functional Dependency Grammar parser), one setting with another decompounding module consisting of two external algorithms which select decompounding suggestions directly from the morphological analyzer, and eventually one setting without the addition of a decompounding module at the pre-processing level.

The Swedish Patent and Registration Office provided me with a collection containing 30,327 patent documents. More specifically, the documents used in my study consist of **patent claims**. Patent claims are short descriptions of the invention that is to be patented. Therefore, the patent documents will henceforth be referred to as patent claims. This specific collection of claims were well suited for research in Information Retrieval since the claims were already manually classified in categories in terms of subject field and subcategories in such a way that the most fine-grained subcategories contain only 3 – 4 claims in average. This simplifies the evaluation of the study. However, the extended use of multi-classification in this type of classification system (one claim can be classified by more then one classification code, see further section 2.4.2) made it more difficult to choose a good query set and to evaluate the performance of the retrieval model.

1.2 Purpose

The purpose of this study was primary to examine if it was possible to use a general automatic retrieval method in order to discover similarities between Swedish patent claims. A secondary purpose was to find out whether the addition of a morphological decompounding module at the pre-processing level improves the result.

With the study I hoped to answer the following three questions:

- Is there enough content in a patent claim to perform automatic similarly calculation with a Vector Space Model?
- Does a decompounding module help finding similarities between Swedish patent claims?
- What is significant for those claims used as queries (or search topics) that generate good average precision values?

Furthermore, this thesis aims to explore several aspects of the Patent Retrieval domain from different scientific fields, which are involved in the retrieval process. Therefore, the terminology and examples used in this thesis are kept as explicit as possible so that librarians, as well as linguists, computer scientists and researchers within the Intellectual Property domain will benefit from it.

1.3 Outline of this thesis

Section **2 Background** contains

- a short overview of the Information Retrieval domain,
- a discussion on general linguistic problems within Information Retrieval, (i.e. normalization, tokenizing, morphological analysis and parsing)
- a discussion on Information Retrieval performed on corrupted data,
- an overview of statistical approaches to analyze texts for retrieval,
    - a description of the Vector Space Model
- the evaluation measurements,
- a general discussion on the Patent Retrieval task:
    - features of patent, patent classification systems, related work,
- an outline of Swedish morphology issues for Information Retrieval, especially features concerning compound words.

In section **3 Method** the study is described in terms of

- the original material,
- how the material was turned into a test collection,
- external software,
- normalization issues,
- how the decompounding was done,
- retrieval implementation,
- how the selection of query set (henceforth search topic set) was done,
- modification of the evaluation measurements.

Section **4 Result** consists mainly of statistical data, in charts and tables, of all measurements used in this study. The main purpose of the charts and the tables is to illustrate the characteristics of a search topic set or of a specific search topic. In the last sub-section a comparison of what distinguishes the three search topic sets from each other is explored, as is the question what parameters that could influence a search topic's ability to capture relevant claims.

In section **5 Discussion and future**, important aspects of this study are discussed, such as natural language processing tools for the patent domain, optical character recognition issues, and Patent Retrieval issues.

2. Background

2.1 Information Retrieval – an overview

In 1948 the first transistor was constructed at Bell laboratories. The invention enabled a commercial use of computers. As for Information Retrieval (IR), the literature for the time period is largely about hardware and the technical capability of automatic document searching and processing (Salton 1987). In the late 1950s the literature was still most concerned with whereabouts of punched card and micro film equipment. However, between 1957 and 1959, H. P. Luhn published a series of revolutionary papers within text processing. According to Salton (1987), Luhns ideas formed the beginning of the computer age in this field. At that time, the notion that keywords for documents had to be chosen by human specialists was common ground among researchers.

One of Luhn's ideas was that the computer was good not only for matching and sorting, but it could also be used for analysing the content of written text. Luhn concluded that, by using the frequency (later known as *term frequency*) and the location of a word, one could automatically index a text through automatic term weighting. In the late 1950s Luhn presents an automatic model to find the ideal set of index terms without human intervention. The automatic model applies **Zipf's law**, which says that the ability of a word to characterize a text is proportional to the words frequency in the text (Zipf 1949). Zipf's law is the *Principle of Least Effort*, which in the context of text production basically states that a writer tends to elaborate on an important notion by repeated use of a lexical unit (keyword) once chosen for the important notion.

> *The justification of measuring word significance by use-frequency is based on the fact that a writer normally repeats certain words as he advances or varies his arguments and as he elaborates on an aspect of a subject. This means of emphasis is taken as an indictor of significance.*
>
> (Luhn 1958 cited in: (Schultz 1968, p. 119))

Luhns assumption was that the frequency of a word in a particular text should yield some essence of text content. Luhn established that not only the most frequent words, like 'the' or 'of', are bad index terms, but also words with very low frequency (Schultz 1968). Therefore, Luhn suggested that the deciles containing the lowest and the highest frequency words of a document should be removed in the indexing procedure. A more common way to remove the frequent word is to use a stop list. Luhns work has been fundamental for automatic text processing and has paved the way for later work in the area (Salton 1987).

Another bright star in the IR heaven is Calvin Mooers. Between 1960 and 1980, he forecasted that machines were to be used for storing very large collections of millions of documents and that searches would be carried out interactively in communication between the user and the systems. He forecasted the online retrieval and both the interfaces and the flexibilities of a user-system. Furthermore, he was the man who coined the term *Information Retrieval* (Saracevic*, et al.* 1997).

Nowadays, many advanced models are used within IR, and there are various ways of displaying the output of a search session. According to Baeza-Yates and Ribeiro-Neto the classical models in IR is the Boolean model, the Vector Space Model VSM and the probabilistic model (Baeza-Yates and Ribeiro-Neto 1999).

The Boolean model is based on the set theory and Boolean algebra (i.e. `AND`, `OR`, `NOT` …). The drawback with this model is that it is based on binary decision criteria: if a search term appears in a document than the document is considered relevant for the query, disregarding the importance of the word's content mirroring power in different documents. Widdows (2004, p. 145) uses a dictionary to exemplify the problem with the Boolean model: "…a dictionary will probably contain most common words that aren't proper name, but the dictionary isn't relevant to every user whose query contains any of these words". Even though the Boolean model has its drawbacks, it is the most used model within the patent domain (Intellectual property domain), since the model will generate a high recall, if the query (by the expert) is well formed (van Dulken 1999). Why is high recall important in the patent domain? Inventions that could be patented should show novelty, uniqueness, should be unseen of, and most importantly, should not be patented by someone else already.

In the 1970s when the document collections grew bigger (however considered small compared to today's standard) the users experienced that the Boolean model returned too many documents and left the users to wade through a large amount of documents in order to find documents that actually were relevant (Widdows 2004). The users requested a new model that could indicate the retrieved documents in descending order of relevance for a query. One of several models developed during the 1970s was the Vector Space Model. The Vector Space model makes use of geometry, where words are being represented by vectors. The coordinations of these vectors are values mirroring the importance of a word in a particular document (Widdows 2004). (See section 2.2.1 for a more exhaustive presentation of the Vector Space Model, since this model is used in this study). Another well-liked model is the Probabilistic model which tries to define the subset of the document collection that would be relevant to a query (Baeza-Yates and Ribeiro-Neto 1999). The key in the Probabilistic model is the observation of the distribution of terms between the relevant documents and the non-relevant documents for a specific search topic (Sparck*, et al.* 2000).

Before further elaboration on Information Retrieval, some terminological clarifying is necessary. A **document** is generally referred to as a unit of text indexed in an IR system (Jurafsky and Martin 2000). The documents that an IR system contains are called a **collection.** A **term** corresponds to a lexical item – it could be a word or a phrase (a sequence of words) that occurs in the collection. A **query** is a set of terms (in which each term is called a **search key**) that represents the information needs of a user and it is matched against the terms in the index-file. In the present thesis I will use the term **search topic** instead of query (except in section 2.2.1 due to the mathematical expressions), since search topic is a more extended term for query and is more frequently used within the Patent Retrieval community. An **index** summarizes document contents by collecting **index terms** from each document in the collection (Baeza-Yates and Ribeiro-Neto 1999). Especially nouns are considered to be good index terms, since they, as (Baeza-Yates and Ribeiro-Neto 1999, p. 24) points out, are easier to grasp semantically.

2.1.1 General linguistic problems connected with IR

An ideal IR system (for patents) would be a system that could understand the users' real needs and only retrieve the information that is relevant, or as (Hedlund*, et al.* 2001, p. 149) express it:

> *An ideal case in searching would be that a search would give all the relevant documents of a database, ranked in an order of descending relevance, and none of the irrelevant documents.*

However, this is seldom the case. The following quotation seems to be more down to earth within Patent Retrieval for a novice user:

> *The information you have is not what you want.*
> *The information you want is not what you need.*
> *The information you need is not available.*
> (Lancaster and Warner 1993, p. 62)

The quotation is drawn from the Proceedings of *The 1986 Clinic on Library Applications of Data Processing*. The entire conference was devoted to how to build a more user-friendly system. Many suggestions on how to make a system more user-friendly was explored at the conference, some good and some not so good. One writer suggested that the IR-problem would be solved simply with the personal computer coming into use.

However, it has later been shown that it is not that simple. Hedlund*, et al.* (2001) give a detail list on linguistic problems within IR:

- The problem with **the selection of alternative concepts and search keys** is associated with the fact that different writers have a variety of synonyms to choose from and different way to express the same thing. This causes problems when one tries to retrieve all relevant documents.

- The problem referred to as **the morphological variation of search keys** accounts for the span of morphological complexity that natural language could have and is therefore linked to the matching process. To retrieve a document the search key has to be identical with the index term for that document. If a word has for example different singular and plural forms in orthography, it means that those documents containing only plural forms of the search key will not be retrieved, if the search key is in singular, and vice versa. Orthographically compound words are another problematic area, which also encompasses morphological variations. Hedlund*, et al.* (2001, p. 149-150) define a compound word as "a word formed by two or more components that are spelled together". To explore the nature of compound words and how compounds differ, see section 2.5.1.

- The problem with **referred and omitted search keys** encompasses anaphoric and elliptical keys. In the example 'Lisa is eating a banana. She likes to eat bananas', the pronoun 'she' is anaphoric – it refers back to the proper name 'Lisa'. In the sentence 'Oscar likes bananas and apples too', information is omitted (the second clause is stripped

of subject and verb). The problem occurs for complex compounds like *iron and steel industry* as well. To omit information in that way is called ellipsis.

- The problem of **search key ambiguity** is associated with the words being polysemic or homographic. A polysemic word has more than one related sense or sub sense, as for example 'crown' meaning "a crown on a kings or a queens head", "someone who rules", "title for winning", "top part of head or hat", "cover for tooth", "top part of hill", "top part of tree" or "unit of money" (Rundell and Fox 2002). Homographs are orthographically identical words with completely different meanings, for example the word 'log' meaning both "a piece of wood" and "a written record on things that happen". Polysemic and homographic ambiguity in language entails that irrelevant documents will be retrieved.

All of the above listed linguistic features generate search results that will contain irrelevant documents as well as overlooking relevant documents. My experiments were performed on a highly demanding domain when it comes to style and vocabulary. Larkey (1999) declares that to the retrieval and classification processing the vocabulary of the patent domain gives several mismatch problems. For instance, inventions that are similar could contain very different terminology, this could actually be an attentive strategies by inventors, "…some inventors intentionally use non-standard terminology so their invention will seem more innovative and to prevent search systems from finding prior art" (Larkey 1999, p180), the term 'prior art' roughly meaning "already existing invention". Moreover, the legalistic language, both the idiosyncratic legal style and terminology, used in claims differ from patent documents in general and the internal style of the patent at hand, as Larkey writes, "Idiosyncratic legal styles and terminology can lead to spurious similarities between patent based on style rather than content." (Larkey 1999, p180).

2.1.2 The normalization process

Before the actual indexing takes place, a few normalizing processes have to be performed, such as tokenization, identification of lemma[1] or stemming, decompounding, parsing and use of stop list.

The first process is tokenization – What should be considered a token? Ahlgren (2004) gives a very detailed description of the problems regarding identification of a token and what should not be considered a token when normalizing a text material. Ahlgren also gives account for what will be lost if one or another analysis is chosen. A token is "a non-empty string of characters"(Ahlgren 2004, p16). ASCII characters as punctuation marks or spaces are considered non-tokens. Also digits can be considered non-tokens, as they have a poor discriminative ability. However, as Ahlgren points out, the alphanumerical string 'U2' (a rock group) will then be diminished to only U. With regard to the collection in my study, all digits were eliminated, since most of them either refer to a particular section in an image, or they constitute a part of an enumeration. Unfortunately, this means that instances like '90-prisma' ('90-prism') and '3-väteatom' ('3 hydrogen atom') are lost. Another problem is the question whether the hyphen should qualify as token separator. Using the hyphen as a token separator will yield higher recall values and not using the hyphen as a token separator will yield higher precision values (Ahlgren 2004).

[1] *lemma* consider to be a word's base form (e.g. *bok* 'book', boken 'the book', böcker 'books' 'the books')

The next process concerns the dilemma of identifying the words' stems. There are two different ways of solving this problem – one is to use a stemmer and the other one is to identify the lemma forms of the words. The difference between these two techniques is that a lemma-identifier checks the lemma candidates against a lexicon and will not yield a nonsense word, while the stemmer guesses the word through pattern analysis and may very well come up with a nonsense word (Dura 1998). A lemma-identifier also referred as a morphological analyzer will identify the word by returning the word's Part-of-speech (POS). Since the word forms are often ambiguous in their POS and it is normally the context that makes them unambiguous (Schmid 1994). An additional process has to be put into use to make the word form unambiguous. A Part-of-speech tagger makes use of the context to automatically predict the POS of a specific word.

There are several methods to automatically predict the POS, such as rule based models, and probabilistic and neural network models. In my study SWETWOL performed the morphological analyse of each word and a Functional Dependency Grammar (FDG) parser was used to disambiguate the words. In the Patent Retrieval domain some studies have used a porter stemmer (Fall*, et al.* 2003) and TreeTagger [2] (Nanba 2007). For Japanese the morphological analyzer ChaSen[3] is used (Fujii 2007), (Nanba 2007).

By using a stemmer or a lemma-identifier the recall will automatically increase, because the terms will become more general and cover different inflectional forms. This is also true (if dealing with highly inflectional languages like Swedish) when a decompounding module is used. The terms will become more general. For example, the words 'student' ('student') and 'revolt' ('revolt') are more general index terms than the compound 'studentrevolt' ('student revolt') (Hedlund*, et al.* 2001). To index every word in a document is not advisable both regarding to storage and speed. Certain words are usually removed from the document by a stop list. A stop list strips the documents from function words as prepositions, conjunctions and highly frequent words in the collection.

2.1.3 IR performed on OCR corrupted data

Since the electronic media have become more present and standardized it has been necessary to transfer older non-electronic documents to electronic format (Beitzel*, et al.* 2003). The material I got from Swedish Patent and Registration Office are typewritten texts that have been OCR-processed (processed by Optical Character Recognition) in France in the beginning of our decade. The optical character recognition errors (OCR error) could affect the performance of the Information Retrieval model since the algorithm is based on exact matches between search terms and text (Vinciarelli 2005),(Beitzel*, et al.* 2003). The OCR errors are caused by graphical similarities (Nylander 2000). Generally, OCR-systems have error rates at about 1–2% of printed character text. If the image is not clear enough the OCR-device either generates a default character (for example ~) or it generates a wrongly identified character or string. Usually the OCR errors are divided into two primary groups – a **non-word error** is a character string not being a word of the language and a **real-word error** is a character string being a word of the language but not corresponding to the original text.

[2] For further discussion see http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger
[3] For further discussion see http://chasen.aist-nara.ac.jp/

A smaller amount of OCR errors do not generally affect the IR models performance, but if the collection contains few documents, or the document at hand is short, the performance will decrease considerably (Beitzel*, et al.* 2003). Retrieval of short OCR text documents will be both unstable and unpredictable (Mittendorf and Schäuble 1996).

There have been different solutions to cope with corrupted data texts, for example a modification of the retrieval model used so that the model also handles noisy characters (Mittendorf and Schäuble 1996), (Beitzel*, et al.* 2003). Over the years there have been different solutions for reducing the OCR errors during the post processing. These programs are mainly lexicon based. Nylander (2000) gives account for two different post processing methods for finding errors in optical scanned Swedish text without the use of a lexicon. The first method consists of rules based on n-gram statistics from a training set, and the second method uses a graphotax (i.e. rules for acceptable letter sequences in a language) created from Bengt Sigurd's model[4] of Swedish phonetics.

In summery, the material in my study is not only written in a highly inflected morphological language, having the characteristic of patent genre concerning vocabulary and style. It also contains OCR errors. All of these features will affect the performance of the IR model.

2.2 A statistical approach to analyse text contents for retrieval.

In the traditional Information Retrieval systems, the index is based on the words within each document (Baeza-Yates and Ribeiro-Neto 1999), (Jurafsky and Martin 2000). These methods are not concerned in which order words are written, the syntax is not of importance (Jurafsky and Martin 2000, pp. 646-654). For example, the sentences 'I see what I eat' and 'I eat what I see' mean the same thing in these systems. This is often referred to as **bag-of-words** methods. Every document could be looked upon as a bag-of-words. In fact, the only thing to consider is to choose the right words from the bag, and the ideal case is that the chosen words capture a document's content. Different statistical measurements or indexing methods are used in this selecting process to find the ideal set of index terms. Each word within a document will be assigned a value that reflects how important it is for the content of the document. This is called **Term Weighting** (Wolfram and Zhang 2008), (Jurafsky and Martin 2000).

Most statistical indexing methods start with calculating word frequency in the collection at hand, both within a specific document and on the entire collection. As I mentioned in the introduction text to section 2, it has been established that the distributional pattern of word types in written text is irregular (Zipf 1949), (Schultz 1968). Words or terms that occur in few documents are considered to be more valuable owed to their ability to distinguish documents from one another than terms that frequently occur in several documents (Salton and McGill 1983), (Wolfram and Zhang 2008).

The **Term Frequency** (TF) is used to compute the frequency value for each word in a specific document. This frequency value is supposed to reflect how salient a word is for the semantic context of the document (Manning and Schütze 2002), (Wolfram and Zhang 2008).

[4] For further discussion see Sigrude B. 1965 *Phonotactic Structures in Swedish* Lund University. Lund

The **Document Frequency** (DF) is used to compute, for each word type in a document collection, how many documents in the collection that contains the word. If a word occurs in few documents, the word is a good **discriminator**.

The terms that occur in few documents are regarded as being more informative of a text's content (Manning and Schütze 2002), (Wolfram and Zhang 2008). The **Inverse Document Frequency** (IDF) uses this phenomenon to extract a word's capability of describing the content of a text. IDF thus calculates distribution pattern of terms across a collection.

Figure 1: IDF formula

$$idf_i = \log(N / df_i)$$ (Manning, *et al.* 2008, p. 112)

$\mathbf{df_i}$, the total sum of documents where word type $_i$ occurs
N, the total sum of documents in the collection

The most widespread term weight method to derive efficient term weights is to use a combination of two different measurements – TF and IDF (Jurafsky and Martin 2000), (Wolfram and Zhang 2008). The combination of the IDF and TF is the term weighting method used in the present study.

2.2.1 Vector Space Model

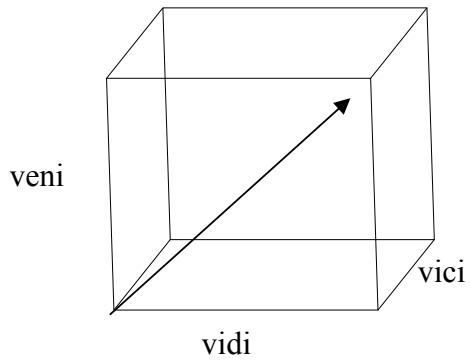
In early IR-literature, Vector Space Models (VSM) were associated with data structures but were not considered a formal and logical retrieval tool (Wong and Raghavan 1984, p. 169). Wong and Raghaven points out that the first time VSM was explicitly described as useful for indexing was by Salton, Wong and Yang in 1975, in an article titled *A vector space model for atomic indexing*. However, already in 1971 in the book *The Smart Retrieval System – Experiments in Automatic Document Processing*[5] the VSM was described and implemented in the SMART system; "… the SMART system treats the terms as a set of orthogonal vectors." (Wong *et al.* 1985, p. 18).

Salton, *et al.* (1975) claimed that it is possible to represent index terms as dimensions in a space. Each term is interpreted as stretching out along one (unique for that particular term) direction, or **dimension.** A document could be represented as a vector describing the defined space (Salton, *et al.* 1975). The minimal vector is the vector representing a document or a query consisting of one single index term. The theoretically possible number of terms (or dimensions) is infinite. An alternative approach in trying to explain what is at stake here is to consider each index term as represented by a vector with a unique direction, the document as the set of the documents' term vectors and, finally, the vector representing the document as the scalar product of the documents' term vectors (Baeza-Yates and Ribeiro-Neto 1999, p. 41ff.). The length of the term vector corresponds to the frequency of the term within the document.

[5] Salton, G. 1971 *The SMART Retrieval System — Experiments in Automatic Document Processing*. Prentice-Hall, Inc.

If a document contains three terms, say ‘veni’, ‘vidi’ and ‘vici’ the document vector would be three-dimensional (see figure 2).

Figure 2: a three-dimensional vector



Thus, the documents and the query are represented by vectors and these vectors contain the terms within the documents and the query, or more accurately – the features of the vectors contain term values within the collection (Jurafsky and Martin 2000, p. 647). This could be formalized as follows:

Figure 3: term features of document vectors

$$\vec{d}_j = (t_{1,j}, t_{2,j}, t_{3,j}, \ldots, t_{N,j})$$

$$\vec{q}_k = (t_{1,k}, t_{2,k}, t_{3,k} \ldots, t_{N,k})$$

(Jurafsky and Martin 2000, p. 647)

$\vec{d}_j$ represents a single document within a collection

$\vec{q}_k$ represents a single query

The **t** features represent the **N** terms within the whole document set (collection) of which $\mathbf{d_j}$ is a member. In case $\mathbf{d}_j$ does not contain a particular term $\mathbf{t_n}$ the value for this term will be 0.

The VSM, in IR, consists of a moderate amount of retrieval methods (Schäuble 1997). Each method contains an indexing method and a retrieval function. As mentioned in the previous section (see 2.2) the indexing method uses a term weighting technique that reflects the importance of a term for analyzing a specific document (Jurafsky and Martin 2000), (Wong and Raghavan 1984). The retrieval function uses the indexing values in computing the similarity between documents and query. The similarity equation consist in computing terms that are common for both query and a certain document, by assigning 0 for absence and the corresponding weight values for the presence terms. The similarity measure is obtained by computing the product of the features of the query vector and the features of the vector for each document in the collection. This could be accomplished by following similarity metric:

Figure 4: formula for similarity calculation with weight values

$$sim(\vec{q_k}, \vec{d_j}) = \sum_{i=1}^{N} w_{i,k} * w_{i,j}$$

(Jurafsky and Martin 2000, p. 650)

$\mathbf{w_{i,j}}$ corresponds to the weight term (i.e. TF-IDF) $_{\mathbf{i}}$ in document $_{\mathbf{j}}$
$\mathbf{w_{i,k}}$ corresponds to the weight term $_{\mathbf{i}}$ in query $_{\mathbf{k}}$

The similarity formula that I have presented disregards the different lengths of the query and the document. For an account of the length one has to normalize[6] this variable and it can be done by dividing each of the dimensions with the overall length of the vector (i.e. the query vector and the document vector), which can be described by the following formula:

Figure 5: cosine normalization factor

$$\sqrt{\sum_{I=1}^{N} W_{i,j}^2} * \sqrt{\sum_{I=1}^{N} W_{i,k}^2}$$

(Jurafsky and Martin 2000, p. 651)

By inserting the normalizing formula one computes the cosine value for the angle between two vectors. If two identical documents were to be computed their vectors would get the cosine value 1. Eventually, the complex formula is presented in figure 6:

Figure 6: formula for similarity calculation with cosine normalization

$$sim(\vec{q_k}, \vec{d_j}) = \frac{\sum_{i=1}^{N} w_{i,k} * w_{i,j}}{\sqrt{\sum_{I=1}^{N} W_{i,j}^2} * \sqrt{\sum_{I=1}^{N} W_{i,k}^2}}$$

(Jurafsky and Martin 2000, p. 651)

In this study, I used only a combination between TF and IDF (i.e. $\mathbf{w_{i,k}}$ and $\mathbf{w_{i,j}}$) as term weighting method together with the Cosine Retrieval Method (se figure 6). The Cosine Retrieval Method consists of a similarity equation with the cosine as normalization factor. This method has performed fairly well in many cases (Jurafsky and Martin 2000), (Schäuble 1997), (Baeza-Yates and Ribeiro-Neto 1999). (For a mathematical example from the collection see section 3.6, figure 20)

[6] Note that this is another normalization concept than the character normalization that has to be done prior to carrying out linguistic analyses of text.

As the reader already may have guessed, each document in the collection will have a cosine value showing its similarity, or closeness, to a search topic (query). Either a threshold or a cut-off value could be used. Jurafsky and Martin (2000, p. 651) suggest that the threshold based on the similarity value be set to 0.80 or 0.90. However, the threshold, based on the similarity value, will differ from collection to collection as it depends very much upon what is being analyzed since the document in theory can differ infinitely from the search topic (Baeza-Yates and Ribeiro-Neto 1999). For instance in my study the highest values of similarities range from 0.05 to 0.9 and the average similarity is 0.2. Since the similarity values differ very much between a search topic and another, I have chosen to use the 100 highest similarity values retrieved for each search topic as the cut-off.

The cosine normalization is a very exact normalization factor, i.e. a document containing (x,y,z) will have the same score as a document containing (x,x,y,y,z,z) (Lee*, et al.* 1997), (Widdows 2004). Mittendorf writes that the cosine normalization also will be affected by the OCR errors:

> "Cosine normalisation is a bad choice for normalisation if you want to retrieve corrupted data, since all terms in a document are used for the normalisation and the *garbage features* [sic] which are introduced by the recognition process have high inverse document frequencies and therefore can destroy any similarity between the perfect normalisation factor and the noisy normalisation factor. "
>
> (Mittendorf and Schäuble 1996, section 6)

Lee*, et al.* (1997, p. 69) propose an alternative normalization factor using only the number of terms in the document and not in the search topic (query) – a more simple normalization factor then the cosine normalization factor.

In general, if high precision (see section 2.3) is desired, a higher threshold should be set. If high recall (see section 2.3) is desired, a lower threshold should be set. One disadvantage with VSM, and other retrieval methods using statistical data, is that it is presumed that authors use the same words to describe similar concepts (Jurafsky and Martin 2000), (Chen*, et al.* 2003). As I have mentioned earlier, the difference in terminology could make similar inventions seem different as the author may try to use unusual terms in order to make the invention more unique (Larkey 1999). Another problem, which is a question of implementation, is that VSM used on a large collection will take time and be memory demanding (Fall*, et al.* 2003). Furthermore, when larger amount of new data enter the collection everything has to be re-calculated all over again.

2.3 Traditional Evaluation Measurements for IR

The effectiveness of the IR systems is estimated by traditional measurements, i.e. recall and precision. The recall value corresponds to the relation between the number of relevant documents retrieved by the system, and the absolute number of relevant documents within the document collection (Jurafsky and Martin 2000). The precision value corresponds to the relation between the number of relevant documents retrieved by the system, and the total number of documents retrieved, i.e. how many of the retrieved documents were relevant documents? Relevance in this context is defined through human judgement.

However, the recall and precision measures are not useful in all retrieval experiments. Systems rarely retrieve documents weighted by 1 or 0 for relevance (i.e. "this is relevant, while this is not"). Within IR, relevance does not have a complementary but a scalar function. Usually the documents are ranked internally as to how relevant they are to the search topic (query) (Jurafsky and Martin 2000), (Manning*, et al.* 2008). Therefore, both recall and precision have been subject to modifications to give better answers to the effectiveness of IR systems. I will only present the measurements used in this study. There is an abundance of measurements with the purpose of establishing how good a retrieval system is.

Recall(N) measures the recall value for N retrieved documents. Recall is obviously an appropriate evaluation measurement when all relevant documents are important to find, for instance in patent and law retrieval (Buckley and Voorhees 2000).

Average Precision (AP) measures the mean of the precision scores obtained after each relevant document is retrieved (Buckley and Voorhees 2000). The value zero is used as an indicator that a relevant document is absent from the retrieved set (Baeza-Yates and Ribeiro-Neto 1999). Average Precision emphasizes the relevant documents with a prominent position in the ranking list of retrieved documents.

Mean Average Precision (MAP) is the standard measurement in the TREC community (Manning*, et al.* 2008). MAP gives a single value that indicates how well the method performed in general over all search topics. To calculate MAP the average precision from each search topic is required (Manning and Schütze 2002).

The method of performing laboratory experiments on test collections and compare the effectiveness of different retrieval models is well-established within IR community (Buckley and Voorhees 2000). Buckley and Voorhees established that an experiment has to be carried out with at least twenty-five to fifty queries in order to be reliable. The authors found that a relationship between the error-rates of the measures did increase when the number of queries decreased.

The evaluation measurement within Patent Retrieval, the modified traditional recall and precision measurements are used (Iwayama*, et al.* 2003a). Kando (2000) gives an account for a discussion on the Patent IR Challenge within the NTCIR project. The aim of the discussion was to explore the characteristics of the patent document, particularly those features that affect the relevance judgment by real users. Therefore, several groups of patent engineers and other professionals occupied with patent document processing were invited to brainstorm (at the ACM SIGIR 2000 Workshop on Patent Retrieval) on what should be evaluated from the users' point of view.

The result of the brainstorm showed that both high recall and high precision are strongly requested in Patent Retrieval. The specialists also concluded that images in patent documents are important to judge the relevance, particularly in the domain electric/machinery/computer industries. The structure of patent documents is overall important, but various parts of a patent document could be important in different stages of the patent application processes.

The evaluation measurements in my study are **recall(100)**, **fallout**[7]**, average precision** and **mean average precision**. In NTCIR[8]-4 and NTCIR-5 the performance of each participant's system were evaluated with Mean average precision as a standard (Fujii, *et al.* 2004), (Fujii, *et al.* 2005). A different evaluation measure, combinational relevance, was used/introduced to evaluate one of the sub task *passage retrieval*. Combinational relevance compute all retrieved passage for each document and compared with the already established relevance passage in a document.

2.4 Patent Retrieval

The following sections (in 2.4) give an overview of research done in the Patent Retrieval community and different aspects connected to the Patent domain.

2.4.1 Features of patent documents

Patent documents are associated with several interesting characteristics such as huge differences in length, strictly formalized document structure (both semantic and syntactic), acronyms and new terminology (Larkey 1999), (Mase, *et al.* 2005), (Krier and Zaccà 2002). It has been noted that many vague, general terms and non-standard terminology are used in patent documents to avoid narrowing the scope of the invention, unlike the stylistic techniques developed within other genres like newspapers and scientific articles.

A cross-genre study, by Iwayama *et al.* (2003), established that Japanese patent documents were approximately 24 times longer than newspaper articles, and a patent application could contain up to 20 times more different words (word types) than a newspaper article. Even Japanese patent attorneys themselves mean that Japanese patent claims are difficult to read (Shinmori, *et al.* 2003). Researchers connected to World Intellectual Property Organisation (WIPO) concluded that the average word frequency fluctuates strongly in both English and German patents (Fall, *et al.* 2003), (Fall, *et al.* 2004).

Sheremetyeva *et al.* explore genre-specific norms for patent texts and illustrates how patent texts and especially claims are composed (Sheremetyeva, *et al.* 1996), (Sheremetyeva 2003). Patent genre language is generally looked upon as a union of legislative sublanguage and the sublanguage of the domain of the invention. The claim should have a good conceptual, syntactic and stylistic/rhetorical structure. "A claim must be composed so as to make patent infringement difficult" (Sheremetyeva 2003, p. 67). A patent claim typically consists of a single sentence, which on the other hand can be several pages long! Furthermore, words used in patent are more abstract and creative than those used in research paper in order to widen the scope of the claim (Nanba, *et al.* 2008).

More research on how special the patent genre could be compared to other genres is needed, as well as research on the internal characteristics of the patent genre (Mase, *et al.* 2005). Furthermore, there is the problem with images – should or should not an image be analysed?

[7] The **Fallout** measures how many of the retrieved document that is of a non-relevant nature among the retrieved (Salton and McGill 1983).

[8] The abbreviation NTCIR should be interpreted as National Institute of Informatics/National Center for Science Information System (NACSIS) Test Collection for Information Research

One argument for using image analysis as a complement is that in many patent domains almost every patent contains an image.

The manner in which a Swedish patent applications or documents should be structured is regulated by law (Patentlagen 1967)[9]. Patentlagen states that a patent application should contain a description of the invention and if possible drawings/images. The application should also include distinct information about what the applicant wishes to protect with the patent claim. The description should be clarifying enough as to permit a professional execution of the invention. The application should also have a technical summary of the patent, but this section is not to be used for the patent.

A European patent application is protected within the Swedish Constitution (Patentlagen 1967), but it should be noted that in the first section of the 82nd paragraph, it is stated that the claim must be translated into Swedish.

2.4.2 The patent classification system

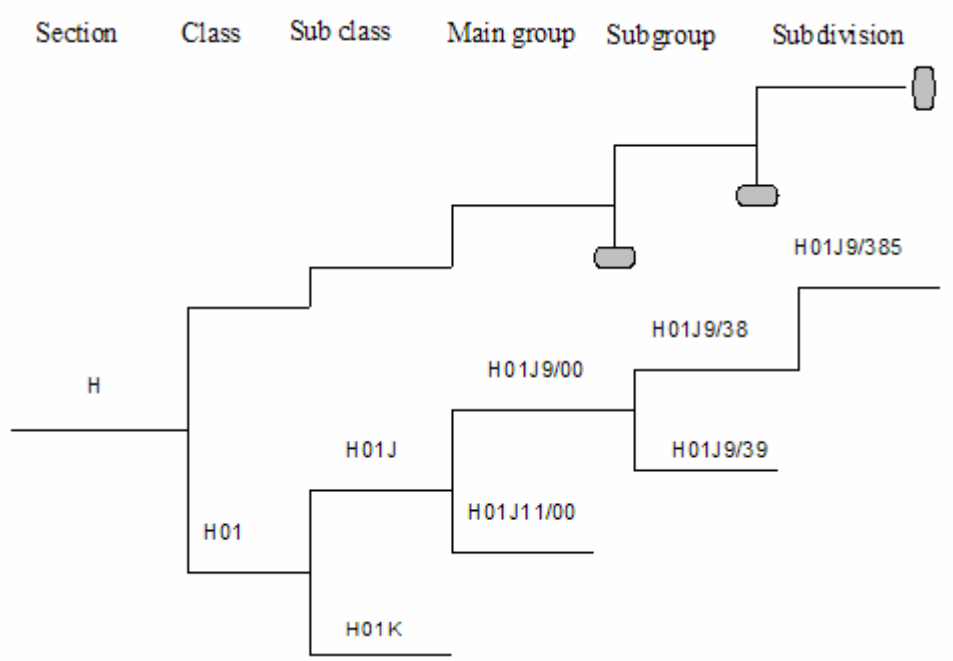
The first classification system for granted patent was developed in France in 1791 and it had an alphabetic structure (WIPO 2004). Since then, many classification systems (also known as intellectual classification schemes (Krier and Zaccà 2002)) have been generated foremost on a national level, but also more trans-nationally. The **International Patent Classification** (IPC) system was developed in 1971 (WIPO 2004a). The administration and development of the IPC is done by the WIPO. The patent claims used in my study are classified by both IPC and the ECLA, which is a European classification system based on IPC.

Each granted patent is given significant symbols, which relate to the technical field or fields it is created for (WIPO 2004). Via these symbols, searchers and industries can keep track on and are able to retrieve documents that relate to the patent in question. IPC is primarily developed to offer a good search tool, and the system is designed to classify any kind of technical invention.

The IPC consists of sections, classes, subclasses and groups. The seventh edition of the IPC has 8 sections, 120 classes, 630 subclasses and almost 69,000 groups with sub-divisions (Fall*, et al.* 2003), (WIPO 2004). Figure 7 shows the structure of the system.

[9] (82 § patentlagen [1967:837])

Figure 7: example of hierarchal structure of IPC



The symbol for 'section' is a capital letter from A through H:

A (*HUMAN NECESSITIES*)
B (*PERFORMING OPERATIONS; TRANSPORTING*)
C (*CHEMISTRY; METALLURGY*)
D (*TEXTILES; PAPER*)
E (*FIXED CONSTRUCTIONS*)
F (*MECHANICAL ENGINEERING; LIGHTING; HEATING; WEAPONS; BLASTING*)
G (*PHYSICS*)
H (*ELECTRICITY*)

Classes consist of two digits following the section symbol (e.g. H 01 – *BASIC ELECTRIC ELEMENTS*). The subclasses are expressed by a capital letter and this letter comes after the class symbols (e.g. H 01 J – *ELECRIC DISCHARGE TUBES OR DISCHARGE LAMPS*). There are two kinds of groups – main groups and sub-groups. The symbol for 'main group' is an integer followed by a slash and the number 00 (e.g. H 01 J 9/00 – *Apparatus or processes especially adapted to the manufacture*). A 'sub-group' has the 'main group' number and an integer counting from 01 at the right hand side of the slash, (e.g. H 01 J 9/38 – *Exhausting, degassing, filling or cleaning vessels*). A third digit at the right hand side of the slash should be interpreted as a further subdivision (e.g. H 01 J 9/385 – *Exhausting vessels*).

One has to keep in mind that a patent can be indexed in two entirely different sections – so called multi-classification. This is because the invention deals with a larger system/component/construction, and at the same time it can be specifically categorized on one or several smaller details of the construction. From library classification systems we are used to one item – for example a book – being classified by only one code, since the code is referring to where one can find the item in the physical room (i.e. at what shelf the item is to be found in the library). The multi-classification technique generally used in patent

classification systems make the patent classification systems differ from the library classification systems, since each item can be classified by more than one code.

The first idea with the IPC actually was that one invention should have only one classification code at the lowest level (main and sub group), since it was developed as a paper search file system (Bruun 1999). However, when the IPC became computerized, multi-classification was encouraged, since the searchers limit their search within pre-specified areas with the help of classification codes in order to avoid too many non-relevant documents (Bruun 1999). In that way, the characteristics of IPC is much like the library subject heading[10] system rather than a library classification system. In a subject heading system one item will be assigned more than one subject heading, since the system has the purpose of describing the item's content with a few terms.

Kier and Zaccá explain the nature of a patent classification system as following:

> *In fact, an intellectual classification scheme like the IPC or ECLA can be seen as a technical language into which during the classification process any document, whatever the original language, is translated.*
>
> (Krier and Zaccà 2002, p. 188)

2.4.3 Problematic features in Patent Retrieval

Mase*, et al.* (2005) addresses four main issues related to Patent Retrieval when developing tools for this domain, (some of these problems are generally connected with IR, see section 2.1.1). The list below is a rendering of the issues addressed in the article.

1. How to extract appropriate terms
    a. Which part of a text should be analyzed – the entire text or a particular part of the text?
    b. What kinds of terms should be extracted – nouns, verbs adjective or others?
    c. How should terms peculiar to patents be collected?
2. How to assign an appropriate weight to each term
    a. What clues should be used – TF, IDF, document length, co-occurrence of terms or text structure?
    b. How much weight should be given?
3. How to treat allomorphs[11] and synonyms
    a. How should the dictionary be constructed – corpus-based or rule-based and manually or automatically?
    b. How should the dictionary be used – category-dependent or independent?
4. How to retrieve the relevant patent documents
    a. What part in a text should be used as a retrieval target?
    b. How should a relevant score be calculated?

(Mase*, et al.* 2005, p. 189)

[10] Dodd, S., Library terms in English and Swedish, http://www.ub.uu.se/bibliotekstermer/?Swe=ä, [visited 2008-12-01]
[11] Allomorphs, a variant of morphemes (e.g. kasta-**r** 'throw', sjung-**er** 'sing')

The issues most closely explored in my study are 1a, 1b (interpreted as morphology in general), 2a and 4a. I investigated Swedish claims as both search topics and target collection and as term weight method I used a combination of TF and IDF.

Mase*, et al.* (2005) state that TF is not appropriate for a Japanese query since the query, in general, will be too short if only claims are used. Another reason is that some terms are used repeatedly since patent is a legal document and should therefore not be ambiguous (Mase*, et al.* 2005). For instance, pronouns and reference terms such as "it" and "that" do not occur in the claim section of patent documents (Mase*, et al.* 2005, p. 192). As a matter of fact for the Swedish patents collection, I have been working with, confirms that the (referential) use of elliptic versions of exhaustive compound terms are significantly sparse. In the patent texts, even a compound term as long as 'polaritetsväxlingsdetekteringsorgan' ('polarity shifting detection instrument') will be written out (almost) every time

The same term is used repeatedly and redundantly and therefore the TF value will become higher than normal. All of these findings seem fairly language independent. However, even if the terms get higher TF values the important term describing the invention tend to have a lower TF value, since these terms are usually mentioned only in the last phrase of the claim (Mase*, et al.* 2005).

Nakatani*, et al.* (2002) concluded that the most effective search was achieved when title, abstract and claims were used all three in combination. However, there will still be problems with this combination due to the inclination by inventors to intentionally use non-standard terms in order to make the invention more innovative and diminish the search system's chances to find prior art (Larkey 1999).

Nakatani*, et al.* (2002) also identified a demand for applications which are able to find similar documents with high recall and precision accuracy. However, the retrieval models (as VSM) and term weighting method, as TF and IDF, are not yet suitable for patent documents because of the uniqueness of the documents regarding complex expressions, coined words and variation in term distribution (Nakatani, *et al.* 2002). The work with finding the best way to find similarity between patents is still in progress. However, since it could take up to two years to complete a Swedish application (Hansen and Järvelin 2000), an assisting tool which could correctly identify similar patents and show the differences would be very useful.

2.4.4 NTCIR

Several international workshops have been held that relate to my study, seven of them being Japanese initiatives (NTCIR-1, NTCIR-2, NTCIR-3, NTCIR-4, NTCIR-5, NTCIR-6, NTCIR-7). The other workshops closely related to my study are in a start up phase (the 1st and the 2nd Information Retrieval Symposium – (IRFS2007 and IRFS2008))[12] and the 1st International CIKM Workshop on Patent Information Retrieval (PaIR'08)[13]).

[12] IRF, Information Retrieval Facility Symposium 2007 and 2008, http://www.ir-facility.org/symposium [re-visited 2008-12-01],
[13] PaIR'08, 1st International CIKM Workshop on Patent Information Retrieval, http://www.cikm2008.org/workshops.php and http://www.ir-facility.org/events/pair08 [re-visited 2008-12-01],

The workshops NTCIR-2, NTCIR-3 were held in a time span from May 2000 to December 2003. These workshops constituted the first serious attempts to explore Information Retrieval for patent documents, and the expression **Patent Retrieval** was established in the IR-community (Iwayama*, et al.* 2003).

Since 2003, the NTCIR provides a large test collection (NII Test Collection) consisting of Japanese patent documents and English abstracts. This test collection has made it possible to researchers from different scientific fields to systematically evaluate methods, particularly within IR text-summarisation and classification (Fujii*, et al.* 2007) . The NII Test Collection contains approximately 3,500,000 documents in Japanese and the time span is over ten years.

In the NTCIR-3 workshop a test collection for a so called Technology Survey (a cross-genre study) was established. The aim of a Technology Survey is to find patents related to a certain technical field. The NTCIR-3 search topics consisted of newspaper clippings which were related to a specific technology. The search topics were divided into the following fields (or combination of fields) – Description, Narrative, Article and Supplement. The overall best MAP value 0.2660 was obtained when the log(tf).idf+dl[14] was used as retrieval method, the search topic was a combination of the fields Description and Narrative and the target collection was the entire patent document collection. When claims were used as target collection the best MAP value 0.1182 was obtained by the same retrieval model. The second best model was the probabilistic model BM25 with 0.2503 (when the entire collection was the target collection) and 0.1129 (when only claim section was the target collection) (Iwayama*, et al.* 2003).

At the NTCIR-4 and NTCIR-5 workshops the focus was on *Invalidity Search* (Fujii*, et al.* 2005), (Fujii*, et al.* 2007). The purpose of Invalidity Search is to compare the search topic, containing rejected applications from the Japanese Patent Office, against already existing patents (i.e. prior art), in order to explore the possibility of automatically finding invalid patent applications. Each rejected patent has been given a so called citation mark (a reference note which indicates the presence of prior art related to the patent).

The Invalidity Search is closely related to my study, since the search topic in Invalidity Search is the claim of the rejected patent. Figure 8 shows the MAP values for the Invalidity Search performed at NTCIR-4 and NTCIR-5 workshops.

Figure 8: MAP for mandatory runs in Document Retrieval Subtask at the NTCIR-5

Table 1. MAP for mandatory runs in Document Retrieval Subtask.

NTC-4-A		NTC-4-B		NTC-5-A		NTC-5-B	
Run ID	MAP	Run ID	MAP	Run ID	MAP	Run ID	MAP
HTC10	.3048	HTC10	.2506	RDNDC505	.1949	RDNDC505	.1619
RDNDC501	.2672	RDNDC501	.2369	HTC12	.1944	HTC12	.1573
ricoh3	.2444	ricoh2	.2035	IFLAB1	.1916	IFLAB1	.1539
IFLAB1	.2137	IFLAB1	.1615	ricoh3	.1766	ricoh3	.1447
kle-patent1	.1445	kle-patent1	.1573	kle-patent1	.0786	kle-patent1	.0757
JSPAT2	.1083	JSPAT2	.0772	JSPAT1	.0683	JSPAT1	.0548
TUT-K1	.0989	TUT-K1	.0768	TUT-K1	.0348	TUT-K1	.0283
# of Topics	31	# of Topics	34	# of Topics	619	# of Topic	1189

(Source: (Fujii*, et al.* 2005, p. 6))

[14] $(1 + \log(f_{q,t})) \times idf_t \times (1+\log(f_{d,t})/(1+\log(avef_d)) \times (1/(avedlb+S\times(dlb_d-avedlb))$ for notation see appendix 3

Run ID is the research team. The letters A and B stand for relevance; where A is a rigid value (i.e. only relevant documents are judged relevant) and B is a relaxed value (both relevant and partly relevant documents are considered relevant). In the NTCIR-4, the relevancy was judged by a human assessor and the citation mark. In the NTCIR-5 the citation mark was the only assessor. In my study, I used the IPC classification code as assessor (See section 2.4.2 and 3.7).

At the NTCIR-6 workshop, Mase and Iwayama (2007) concluded that claims are not sufficient for collecting patents with high similarity. Mase and Iwayama also concluded that both text analysis and retrieval algorithm should be modified to the technical field and the intention of the search topic. Mase and Iwayama conducted the study on a Japanese test collection. I conducted my study on Swedish patent claims, both as search topics and target collection, and I will also show that Mase and Iwayama findings are true also for Swedish claims, even when a decompounding module is used in the pre-processing stage.

The aim with NTCIR-7 workshop was to develop techniques for effective retrieval and classification of patents and research papers with the IPC instead of the F-term[15] system, which was used in previous NTCIR workshops (Nanba, et al. 2008). The participant teams were asked to submit a ranked list of 1000 IPC codes for each search topic. The formal run consisted of 897 search topics and the average of IPC codes assigned to each search topic was 2.3 codes. There were four different subtasks:

- classification of Japanese research papers using patent data written in Japanese –a cross-genre study,
- classification of Japanese research papers using patent data written in English – a cross-lingual study as well as a cross-genre study,
- classification of English research papers using patent data written in English,
- classification of English research papers using patent data written in Japanese,

The best result for both the Japanese and English cross-genre study as well as the cross lingual study exceeded a MAP value of 0.4. Now, this was a cross genre study and – as Nanba *et al.* (2008, p. 326) write – terms used in the patent domain differ from the terms used in research paper. For example, the hyperonym for the scholarly term "machine translation" is "natural language processing" but in the patents the terms used are "automatic translation" or "language translation". Subsequently, to obtain terms both occurring in patents and in research papers the terms were retrieved from both research papers and different parts of the patent. A combined hypernym-hyponym-based method and citated-based method was used to obtain the patent term as paraphrase of a given scholarly term (Nanba, *et al.* 2008).

[15] A Japanese patent classification "File forming term system".

2.4.5 Other research projects within Patent Retrieval

The importance of patent processing has recently been recognised by Information Retrieval and natural language researchers (Fujii, *et al.* 2007). In this section I will refer to other research study with a different focus but still within the patent domain.

At the ACME SIGIR 2000 and the ACL 2003 workshops a Swedish study was presented (Hansen and Järvelin 2000). The study was conducted by researchers at the Swedish Institute of Computer Science (SICS) in cooperation with the Swedish Patent and Registration Office (SPRO), and focuses on how information seeking is related to the work-task for the patent engineers. While many of the IR studies are undertaken within a controlled laboratory environment and with controlled variables and with the researchers merely simulating information needs, this study wanted to explore a real work task where information seeking was an ongoing process. The authors argue that in order to understand the performances of information seeking and retrieval one has to take in a broader perspective and not just be content with studies in laboratory environments.

For instance, the main search tools in the patent domain have been the classification systems, and the other search tools, such as the keyword search systems have primarily been complements (Bruun 1999). However, in some fields of technology the classification search strategy has been substituted with other search tools, but the classification is still the main search tool in general. Only using a few key words will generally generate too many non-relevant documents. Therefore, the search strategies used in Patent Retrieval differ from the Internet searching where the users generally only write in 3 words (van Dulken 1999). The search strategy within the patent domain is rather a combination of both the classification's codes and keywords combined with Boolean syntax (Bruun 1999), (Lyon 1999), (Krier and Zaccà 2002). One has to keep in mind, that selecting appropriate keywords (exhaustively finding all synonyms and phrases describing the same concept of a patent) is a difficult task – it is "sometimes similar to gambling" according to Lyon (Lyon 1999, p. 90).

In the Swedish study, by Hansen and Järvelin (2000), ten patent engineers at SPRO were closely observed by researchers at SICS during a five-week period from May to late June 2000. The study was conducted in five phases – an introduction seminar, a pre-interview session with open-ended questions, diary writing, observation sessions and a closing seminar. Hansen and Järvelin (Hansen and Järvelin 2000) concluded that a patent engineer search task and work consisted of different levels of collaborative activities, both individual and group activities. In a later article, in 2005, Hansen and Järvelin (2005) elaborated on the different aspects of collaborative activities within the patent domain.

In 2003 WIPO started a research project to develop an automatic categorization assistance program – Classification Automated Information System (CLAIM). The aim of the CLAIM-project was to develop an automatic categorization assistance program for several languages. The first study was performed on the English data set in 2003 (Fall, *et al.* 2003). In June 2004 the WIPO-alpha collection extended to five languages – English (497,135 documents), French (832,449 documents), Spanish (198,805 documents), Russian (182,227 documents) and German (238,903 documents) (WIPO 2007). The English and the German classification studies were test cases in order to develop the automatic categorization assistance tool.

In the English study, two automatic categorization methods were used – the Rainbow package (implements of naïve Bayes, k-Nearest Neighbors (k-NN) and Support Vector Machine

algorithms), and the machine learning architecture SNoW (Sparce Network of Winnows) (Fall*, et al.* 2003). The English data set of the WIPO-alpha collection contains documents from different countries worldwide. The language of the documents is English and the documents have been converted into electronic format via optical character recognition. According to the author, a far-reaching automatic and manual checking of the OCR-files resulted in few cases of errors in the final collection.

The title and the claim were indexed to describe the invention/patent. The index vocabulary of title and claim yielded 25,000 200,000 words respectively. The next 300 words comprising the titles, inventors, applications, abstracts and descriptions resulted in an index of about 200,000 words. This index was later reduced to 150,000 for computational efficiency. The outcome of the normalizing process was that the length of a patent document ranged from a maximum of 6,200 words to a minimum of 275 words. The authors observed that class C07 "Organic chemistry" was the foremost contributor to the diversity of the vocabulary. This class contained tens of thousands of DNA sequences.

The evaluation was performed on class level and subclass level. The best performance on the class level was when the first 300 words were indexed. The authors observed that the claim section overall generated poorer results than the other index alternatives even though the vocabulary size was similar. Overall, the support vector machine algorithm outperformed the Naïve Bayes, the k-NN and SNoW algorithms, especially at the subclass level. At the class level the performance was more uniformed

In the German study a stemmer was used in the pre-process – the stemmer dealt with most of the German suffixes (Fall*, et al.* 2004). However, no decompounding module was used even though the authors considered the German patent collection a more demanding test case than the English one. Not to implement a decompounding module was a strategic choice allowing language-independency and a relatively quick implementation into a system.

The German vocabulary was larger than the English when all the words were indexed. Over 850,000 different words were established in the first-300-words index. The authors used their own categorization tool instead of the SNoW. This tool contained a k-NN algorithm and a Linear Least Square Fits Method (LLSF). The evaluation revealed a fairly good performance in comparison with the former English patents classification, in spite of the different morphological characteristics of the two languages (Fall*, et al.* 2004).

In 2008, there are several on-going projects to explore the automatic processing of the patent genre ranging from retrieval to re-classification and mapping articles to a specific patent. Wanner*, et al.* (2008) explore a semantically oriented patent processing service, as opposed to the usual textual surface where the user has to "hypothesize how surface textual clues reflect the content" which is usually a time-consuming process, and the outcome can not be guaranteed to satisfy the users information need. The authors claim that if the patent material were specified in an explicit and unambiguous semantic representation this would make the retrieval, classification and validation (both machinery and human) more straight forward. The system will make use of state-of-art semantic web content representation to do content distillery and semantic patent search.

2.5 Features of Swedish morphology, which could affect the performance of IR systems

Hedlund*, et al.* (2001) identifies five features of Swedish morphology that they consider to affect IR systems performance:

- Swedish is rich with homographic word forms, e.g. the noun 'dom', meaning 'cathedral' as well as 'verdict/judgment'. This word type could very well be a good candidate for index term, independent on what it means. According to Hedlund *et al.* ((Hedlund*, et al.* 2001, p. 154) cited Karlsson 1994[16]), around 65 percent of the words in a Swedish running text are homographs. Furthermore, Hedlund *et al*. (still cited same author) establishes that the homographs amount to no more than 15 percent in a Finnish text, and to 50 percent in an English text.

- Swedish nouns are classed into two grammatical gender categories – **neuter** and **uter**. These gender features could in fact help resolving problems with homographs, since homographs usually belong to different genders (Hedlund*, et al.* 2001, p. 152). Also anaphoric resolution algorithms could use gender features for linking pronouns to their nominal correlate.

- Swedish nominal morphology is fairly rich. Adjectives and articles agree in gender, number, and definiteness with the nominal they modify. Definite articles are suffixed to their head. Nouns are usually sub-categorized into five declinations, along with their plural suffixes. Umlaut inflections are frequent, e.g. singular 'broder' ('brother') and plural 'bröder' ('brothers'). Genitive case is expressed by means of a suffix (i.e. –**s**). Due to the rich inflectional morphology, the indexing and matching methods used in IR are insufficient for Swedish (Hedlund*, et al.* 2001, p. 151). Inflection hampers pattern matching, and as a consequence also the weighting processes.

- Complex (multi word) noun phrases are less common in Swedish than in English, the Swedish equivalents forming compound units. Multi-word phrases are generally difficult to deal with in IR. The Swedish counterpart – the compound – is highly frequent, especially the subclass **productive compounds** (Ekeklint 2001), (Järborg 1998). According to Karlgren (2005) 10 percent of the words in Swedish running text are compounds, and in a search topic every second search term, considered to be important for the topic, was a compound. Karlgren (2005, p. 111) mention that compounding is as productive process and "…new compounds can be formed on the fly or ad-hoc proposes to treat topical elements in the discourse at hand". Furthermore, languages where the compounding process results in an orthographical unit (closed compound) could well conceal crucial elements inside a compound (Karlgren 2005).

- Change of Part-of-speech (POS) class by derivation is extremely productive in Swedish. There are different derivation suffixes, which very often determines the Part-of-speech category of the word (e.g. 'lära-re' ('teatch-er'), 'fri-het' ('free-dom') and 'mål-n-ing' ('painting' (as noun)) (Hedlund*, et al.* 2001).

[16] For further discussion see Karlsson, F. (1994) *Yleinen kielitiede*. Helsinki: Yliopistopaino [General linguistics]

Hedlund *et al.* (2001) recommend that a morphological analysis program be used to normalize the text, in the indexing as well as in the search stage. A disambiguation module for homographs might also be useful in Swedish IR systems. They also concluded that more research on the Swedish language for IR is needed. According to Teleman, *et al.* (1999, vol. 1 pp. 20-21) approximately 20 million persons have a basic knowledge of Swedish[17].

2.5.1 Compound Word Features

Before elaborating on compounds, I will unfold some morphological terms.

Swedish morphological units can be subdivided into **free morphemes** and **bound morphemes**. Free morphemes, or **root morphemes** (henceforth I will refer free morphemes as root morphemes), are independent words with independent meaning. Free morphemes usually belong to the open word classes – nouns, adjectives, verbs. Bound morphemes are not complete words, one could say that they modify the root morphemes and give them a more or less different meaning. This is why they are also called **grammatical morphemes**. The category of bound morphemes can be further divided into **derivative morphemes**, **inflective morphemes** and **interfix morphemes** (Malmgren 1994), (Hedlund*, et al.* 2001), (Bauer 1994). Roughly speaking, a derivative morpheme alters the POS class identity of a root morpheme, whereas an inflectional morpheme modifies a root morpheme with regard to definiteness, number, gender, case, comparison or tense. The interfix morpheme constitutes a class of grammatical morphemes that is important for compounding in Swedish, although it is far from always applied. The interfix morpheme is the glue between two words. There are five graphemes representing the interfix morpheme – a, o, u, s, e – all of them originally genitive markers (Teleman*, et al.* 1999).

Figure 9: example of morpheme analysis

barn|s|lig ('childish')	root morpheme + interfix morpheme + derivative morpheme
av|led|ning ('derivative' (n))	derivative morpheme + root morpheme + derivative morpheme
barn|s|lig|are ('more childish')	root morpheme + interfix morpheme + derivative morpheme + inflective morpheme
av|led|ning|en ('the derivative')	derivative morpheme + root morpheme + derivative morpheme + inflective morpheme

Malmgren (1994) offers the following definition of a compound word:

> *A compound word is a word which can be split into at least two word-like units, both of them containing at least one root morpheme* [author's translation]
>
> (Malmgren 1994, p. 32)

Hedlund*, et al.* (2001, p. 149) offers an explanation of the difference between compound and phrase usually adopted within IR:

> *"... a **compound** is defined as a word formed by two or more components that are spelled together. The term **phrase** is used for the case where the constituents are spelled separately"*

English morphological theory identifies compound structures in English, albeit the words are not joined together, for example 'window cleaner', 'emergency sail change' or even

[17] Including the population of Denmark, Norwegian and the geographic part of Finland where Swedish is spoken; also including student at universities taking courses in Swedish as foreign language.

'supermarket parking lot attendant'[18], the motive for this being phonological (Spencer 2001). Swedish compounds also demonstrate phonological characteristics. The Swedish equivalent to Spencers term **compound stress** is **sammansättningsbetoning** (Riad 1997).

The most common compounds in Swedish are combinations of noun plus noun, adjective plus noun, and verb plus noun (with descending frequency). Hedlund*, et al.* (2001, p. 154) even regards particle verbs (i.e. verb plus adverb or preposition) as compounds. The combination verb plus verb is very rare (Malmgren 1994). Within the class of noun compounds there are combinations with proper names and other encyclopaedic units, for example 'mellanösternspecialist' ('Middle East specialist'), 'Hultsfreds-biljetter' ('Hultsfred tickets' tickets for a rock music festival) and 'Björnborgväska' ('Björn Borg bag'). This kind of compounding is quite common in Swedish (Järborg 1998).

2.5.2 Compounding and IR

Compound words are easier to detect than phrases but, on the other hand, they have to be decomposed for IR to be effective (Hedlund*, et al.* 2001). Karlgren *et al.* (2004) claim that it is important to decompound a compound into its parts, because compound parts are likely to be content bearing (e.g. diamantgruva 'diamond mine'). On the other hand, it is contra-productive to split a compound in case the splitting (decompounding) causes a complete loss of content (e.g. 'riksdag' ('Parliament of the Kingdom'), for which a splitting would yield the words 'rich' (instead of 'kingdom') and 'day'. To find correct splitting is in many ways similar to sense disambiguity and it is important for many language applications such as grammar checking, retrieval and machine translation (Sjöbergh and Kann 2004).

How to choose the correct segment split is an on-going quest. Sjöbergh and Kann (2004) developed a statistical approach that manages to find 99% of all the compounds and 97% of them were correct. Sjöbergh and Kann (2004) test collection consisted of 50,000 words from the Stockholm-Umeå Corpus (Ejerhed*, et al.* 1992). 1,300 out of 3,500 compounds in the test collection were ambiguous (could be decompounded in more than one way). As a base Sjöbergh and Kann (2004) used an individual word list (consisting of words that can not be part of a compound,) a last part list (consisting of word that can be an end part of a compound or and independent word) and a first part list (consisting of modified stems that can be the first or the middle part of a compound).

Sjögbergh and Kann method to decompound was reused by Dalianis in 2005 on web material. The material consisted of nine Swedish public websites all in all containing 100,000 documents. The search engine obtained 64 percent more relevant hits when the compounds were split in queries (Dalianis 2005). However, Karlgren (2005) writes that a retrieval system for a compounding language such as Swedish should ideally split both when indexing and when processing the search terms. Both Karlgren (2005) and Dalianis (2005) agree that the two folded problem "which compound should be split and how" need more research?

To perform compound analysis is a question of judgment involving both knowledge of language and understanding of the context at hand (Karlgren 2005). The meaning of a compound can not always be predicted by its parts. In such cases one has to simply know the

[18] Examples are from Johnson, M., Handouts for class 2002, CG41 Morphology, the structure of words, http://www.cog.brown.edu/~mj/classes/cg41/handouts/wk02a.pdf, 2003-10-15

meaning of the compound to understand the word. These types of compound words are called **opaque compounds** or **exocentric compounds** (Ekeklint 2001). The Swedish word 'jordgubbe' ('strawberry') is a typical opaque compound, the separate components 'soil' and 'old man' not offering a sensible clue to the meaning of the word. This type of compound should preferable not be decompounded.

The **productive compounds** mentioned in section 2.5 are very often new words that a writer creates for a specific context (e.g. indexeringsmetod ('indexing method')) and should preferable be decompounded (Ekeklint 2001). But there are also compounds in this class, which are high frequency words in every day spoken language (e.g. lastbil ('truck')) (Ekeklint 2001) and as a results these high frequent compounds are consider a single item/lexeme in a language (Karlgren 2005).

As mentioned above Sjögbergh and Kann use a statistical method to select and split compounds based on pre-generated lists. Dalianis (2005) suggests another method to split and select compounds. A compound should be split in two parts and the right most part should be the longest. The interfix morpheme 's' should be used as a split marker. By only using the 's' as a split marker the compounds that are allowed to be decompounded are those constructed with a 's' as interfix morpheme. Using this approach in the patent collection at hand the compounds 'styrhylsa' and 'styrorgan' both meaning 'guiding sleeve' will not be decompounded. Meanwhile the 'planterings**s**organ' and 'planterings**s**enhet' both words meaning 'a planting unit' will be allowed to be decompounded since the 's' marker is present. However, the condition that the right most part has to be the longest will not be fulfilled.

Karlgren (2005) addresses the two folded problem differently by conducting statistical observation on the CLEF (Cross-Language Evaluation Forum) collection of 140,000 Swedish news articles. One of Karlgren's finding was that the left element of the compound could be useful in a query expansion. Karlgren exemplifies with the compound 'diamantgruva' ('diamond industry') where "…"diamant" is likely to be a useful query expansion, far more than other "industri" would" (2005, p.114). If we were to implement this approach in the patent collection at hand the compounds pairs'styrhylsa', 'styrorgan' and 'planteringsorgan', 'planteringsenhet' would only add 'styr' ('control') and 'plantering' ('planting') to the query.

Another method could well be to use linguistic information. The productive compound is constructed so that the last component of the compound gives the semantic hyperonym to the word (e.g. 'löv**skog**' ('deciduous **forest**')). Sometimes the last component of a compound is used elliptically. That is, once a compound word has been introduced, the first component can be omitted, implied, later in the text in order to smooth out the reading (e.g. 'a/the deciduous forest' can subsequently be referred to as 'the forest') (Ekeklint 2001), (Hedlund*, et al.* 2001, p. 152).

When splitting Swedish compounds one has to know where the parts start and stop (Dura 1998). It can be quite difficult to split compounds at the right place. Sometimes compounding produces ambiguity, such as when the interfix morpheme meets with an identical letter in one or both of the surrounding words, see figure 10. Swedish spelling conventions do not allow for more than two identical letters following each other. Word-final gemination of a letter will be reduced to a single letter if, when compounding, the geminate sequence meets with a word starting with the same letter.

Figure 10: example of decompounding analysis

Word: glassko	
Interpretation	**Translation**
glas|sko	glas-shoe
glass|sko	ice-cream shoe
glass|ko	ice-cream cow

Sometimes bound morphemes coincide with homographic root morphemes. The complex compound 'självständighetsförklaring' ('Declaration of Independence'), for instance, could (erroneously) be split into five root morphemes in two different ways:

Figure 11: example of decompounding analysis

Parts of the compound	**Erroneous interpretation 1**	**Erroneous interpretation 2**	**Reasonable interpretation**	
själv	root morpheme	root morpheme	root morpheme	word
ständig	root morpheme	root morpheme	root morpheme	
het	root morpheme	root morpheme (hets ('baiting'))	derivative morpheme	
s	Interfix morpheme		interfix morpheme	
för	root morpheme	root morpheme	derivative morpheme	word
klar	root morpheme	root morpheme	root morpheme	
ing	derivative morpheme	Derivative morpheme	derivative morpheme	

The analysis of the second root morpheme is still uncomfortable 'ständig', even in the reasonable interpretation of the compound. The morpheme is in fact in itself a complex word formed by the Old Swedish 'ständ' (standing) and the (German) derivative suffix 'ig' (SAOB[19]), also displaying an umlaut phenomenon.

Another phonological convention when building compounds in Swedish is that word final vowels which do not belong to the main stress syllable in the word are omitted and substituted by an interfix morpheme or by the "empty morpheme" (Teleman*, et al.* 1999, vol. 2 §31-§38). When decompounding compounds, the left part of the compound may therefore actually constitute a non-word, and this has to be taken into consideration when using decompounding in IR.

[19] Svenska Akademien, Svenska Akademiens ordbok, http://g3.spraakdata.gu.se/saob/ (entries: STÄNDIG; STÅND), [re-visited 2007-11-01], http://g3.spraakdata.gu.se/saob/ (entries: STÄNDIG; STÅND)

3. Method

In the appendix 2 a flow chart of the entire study is presented.

3.1. Material – the collection of claims

In 2003 I received 30,327 claims from the Swedish Patent and Registration Office. But the actual number was 30,217 since there were 110 doubles in the material. The claims were stored on an unusual memory unit, which is used only in main frame environment. To extract the material I got help from Volvo Information Technology in Gothenburg.

However, the memory unit was not the only unusual phenomenon in this material. The material was also encoded in EBCIDIC instead of the more common code scheme ASCII. So, before I was able to view the material I had to convert the material to ASCII. First, I tried to use standard modules for this task, but they only gave spurious results, so I had to write my own converter. The conversion of the material from EBCIDIC encoding to ASCII encoding made images impossible to analyze, since there was no one-to-one matching for characters other than alphanumerical characters. The presence of images was indicated by a large amount of noisy characters. Since the image information became corrupted during the conversion from EBCDIC to ASCII I decided to remove all image information.

The claims in the collection proved to be distributed in all IPC sections as can be seen in figure 12. The date of the claims seemed to range from October 1991 to March 1993 according to the XML-tag. But during the preprocess I found that the claims could be registered as far back in time as 1975, (I even found a patent claim from 1915).

Figure 12: the IPC distribution of the claims in the collection

Sections	Classes	Sub class	Group/subgroups	Number different of claims
A HUMAN NECESSITIES	15	82	3396	5258
B PERFORMING OPERATIONS; TRANSPORTING	34	164	8444	11204
C CHEMISTRY; METALLURGY	19	93	7116	7368
D TEXTILES; PAPER	8	37	1052	1396
E FIXED CONSTRUCTIONS	7	31	1631	2200
F MECHANICAL ENGINEERING; LIGHTING; HEATING; WEAPONS; BLASTING	17	95	4481	5639
G PHYSICS	13	74	3407	4192
H ELECTRICITY	5	48	3579	4003
SUM	118	624	33106	41260

The IPC information was extracted from esp@cenet (http://ep.espacenet.com/) in 2004

Note that the resulting number of claims in the figure will not coincide with the number of claims in the collection. This is due to the extensive use of multi-classification (one claim can be classified by more then one IPC code, see section 2.4.2).

The claims were scanned and digitalized in France during the late 1990s or early 2000s, and there should have been 99.9% accuracy in letter identification. However, at a very early stage I found that several of the words in the claims were not correctly identified in the OCR-process. An extensive semi-manual OCR-correction was performed on the material to reduce the OCR-errors.

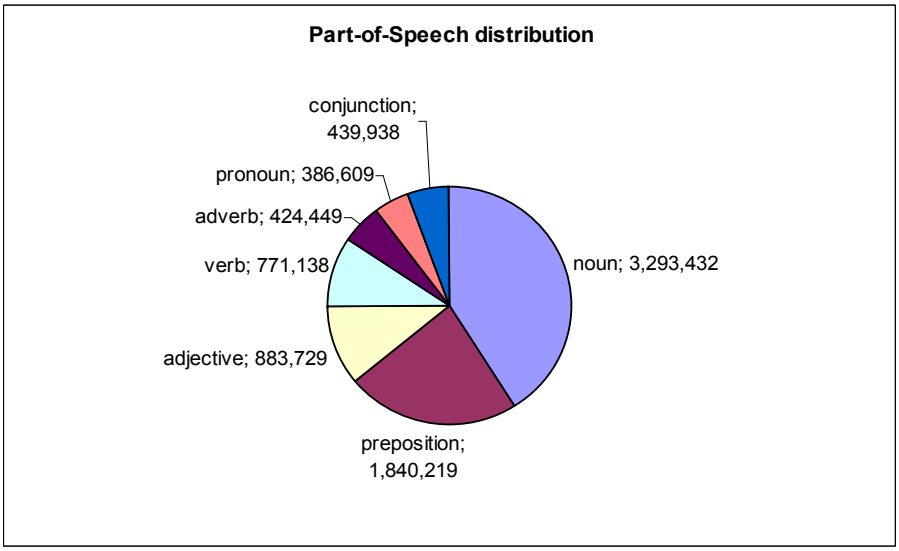
The collection contains approximately 11,000,000 tokens (only counting digits and letter units) and the total number of compounds is 1,070,972 according to the Functional Dependency Grammar Parser. The average value and the median differ distinct for document length, sentence length, and frequency occurrences;

- the average length of a claim is 320 tokens and the median is 262 tokens, the longest claim contains 4,450 tokens and the shortest only 3 tokens,
- the average sentence contains 70 tokens and the median is 16, the longest sentence contains 565 tokens and the shortest only 1,
- the average lemma occurrence in 11 claims and the median only occurs in one.

The claim that only contains 3 words is classified in section C and it consists of the words 'förening' ('union'), 'med' ('with'), 'formel' ('formula') and an image of a molecular structure. As the chemical compound is represented as an image, the essential words in this claim will not be available for my similarity computation. It is difficult to determine how many of the 7,368 claims classified in section C that display the essential chemical compound as an image. In some cases the alphanumerical characters in the molecular structures seem to have been identified in the OCR-process, and in some cases not.

According to the Functional dependency parser the most frequent POS is a noun (3,293,432). Figure 13 shows the distribution of POS (only: noun, verb, adjective, pronoun, conjunction, preposition, and adverb).

Figure 13: Part-of-Speech distribution in the collection



3.1.1 Examples of claims from the collection

Figure 14 and figure 15 illustrate an example of retrievable patent claims in the collection. These two different claims are classified by the same IPC code A01C11/02 (i.e. transplanting machines for seedlings). The first claim (figure 14) was selected as a search topic and the second claim (figure 15) generated the highest similarity value for this search topic. The mathematical calculation between these two claims is shown in figure 20 (in section 3.6). (For the English abstracts of the two claims and relevant pictures see appendix 1.)

The italic and bold words are still OCR errors present in the two claims, errors consisting of both non-word errors and real-word errors.

Figure 14: claim 436822

<startpatent>
patentid(DNUM= 436822 DATE=19910218)
1. Anordning vid planteringsmaskiner, företrädesvis sådana för plantering av skogsplantor, kännetecknad av kombinationen av ett vid planteringsoperationen roterande planteringsorgan (1-8) och ett enda, till ***planteringsorgaflet*** via ett i alla riktningar elastiskt eftergivande element (10), företrädesvis ett gummielement, anslutet markberedningsorgan (12, 13). 2. Anordning enligt patentkravet 1, kännetecknad därav, att planteringsorganet har formen av två relativt smala, lämpligen böjda, i användningsläge ***nedat*** konvergerande och i sin nedre ände varandra korsande skär (4a, 4b) vilka är ledbara *sa* att de från nämnda verksamma läge kan svängas upp till ett overksamt läge genom att nämnda böjda skär är förda genom öppningar (3) i ett med ett roterbart planteringsrör (1) förenat styr- och formningsorgan (2) och lagrade vid en i förhållande till nämnda planteringsrör förskjutbar hylsa (8). 3. Anordning enligt patentkraven 1-2, kännetecknad därav, att markberedningsorganet (12, 13) uppvisar en nedre skäryta (13) vilken lutar nedåt i riktning ut från planteringsorganets centrum och att styr- och formningsorganet (2) är koniskt med ungefärligen samma vinkel som skärytans lutning sa att sagda skäryta huvudsakligen kommer att utgöra en förlängning av den nedre ytan av sagda styr- och formningsorgan. 4. Anordning enligt patentkraven 1-3, kännetecknad därav, att det eftergivande elementet med sin ena ände är förbundet med styrhylsan (8) t.ex. via en vid dennas mantelyta anbragt fästplatta (9) sträcker sig huvudsakligen vinkelrätt mot sagda styrhylsa och i sin andra ände är ansluten till en huvudsakligen vertikal fästplatta (11) hos markberedningsorganet (11-13). 5.Anordning enligt patentkravet 1, kännetecknad därav, att i planteringsorganet är förskjutbart anordnad en planthållare (15), kopplad för rörelse med ett i planteringsorganet ingående organ (8) för påverkan av planteringselement (4a, 4b), att i sagda planthållare ingriper ett spärrorgan (19), verkande mot översidan av plantans rotklump och i avsikt att förhindra en sagda planthållare befintlig planta att förflytta sig i riktning ***uppat***, och att sagda spärrorgan i samband med planteringsorganets återgång till utgångsläge för ny plantering bringas ur ingrepp med plantan ifråga. 6. Anordning enligt patentkraven 1-5, kännetecknad därav, att spärrorganet (19) är anslutet till en vridbar axel (21) vilken står under inverkan av en torsionsfjäder och vilken i samband med en planteringsoperations avslut- ande medelst en vid nämnda axel (12) fäst anslagsdel (23) kommer till samverkan med ett vid anordningens stationära del fäst medbringarorgan (24), vilket lämpligen för säkerställande av ingrepp är koniskt utformat.
<endpatent>

Figure 15: claim 408121

<startpatent>
patentid(DNUM= 408121 DATE=19910318)
1. Planteringsaggregat, speciellt för plantering av skogsplantor, ***k å n fl e t e c k n a-t*** av minst en planteringsenhet (11) med organ för tillförsel av plantor till ett ***väntelåge*** (18) för plantering, organ (24,26) för tillförsel av en förutbestämd ***jordmångd*** förbi ***vänteläget*** och i riktning mot ett lämpligt planteringsläge och styrorgan (21,16,17) för styrning av jordtillförseln på ett sådant sätt, att jorden kommer att omsluta plantans rotklump, vilka styrorgan utgörs av en ovanför vänteläget anordnad falljordsdelare (21), som delar den nedfallande jorden i två på var sin sida om plantan nedfallande delar, och två vid var sin sida om vänteläget belägna vingar (16,17), som ***år*** anordnade att styra in den nedfallande jorden i riktning mot plantans rotklump. 2. Aggregat enligt patentkravet 1, kännetecknat därav, att styrorganen (21,16,17) är inrättade att styra jordtillförseln så, att den för varje tillfälle nedfallande jordmängden helt eller delvis åstadkommer plantans frigörande från vänteläget. 3. Aggregat enligt patentkravet 1 eller 2, kännetecknat därav, att.planteringsenheten (10), jordtillförselorganen (24,26) och styrorganen (21,16,17) är sammanförda till ett såsom en enhet till en lämplig basmaskin anslutbart aggregat, varvid basmaskinen lämpligen är anpassad att bära ett jordmagasin (10) och ett plantförråd. 4. Aggregat enligt något av patentkraven 1-3, kännetecknat därav, att det ***år*** försett med ett flertal planteringsenheter med tillhörande jordtillförselorgan, styrorgan etc., varvid det inbördes avståndet mellan planteringsenheterna företrädesvis är lätt variabelt. 5. Aggregat enligt något av patentkraven 1-4, kännetecknat därav, att varje planteringsenhet uppbärs av en i basmaskinen eller ett stativ företrädesvis allsidigt ledbart lagrad-pendelarm (14), vars undre ände eventuellt kan vara anordnad att släpa mot marken. 6. Aggregat enligt något av patentkraven 1-5, kännetecknat därav, att pendelarmarna (14) är fjäderpåverkade (15) i riktning mot ett neutralläge, varvid fjädrarna lämpligen har progressivt tilltagande fjädringskarakteristik. 7. Aggregat enligt något av patentkraven 1-6, kännetecknat av i framryckningsriktningen efter planteringsenheten anordnade avstryknings- och tilltryckningsdon (23,20). 8. Aggregat enligt något av patentkraven 1-7, kännetecknat därav, att tilltryckningsdonen utgörs av snedställda valsar (20), som är belägna på ömse sidor om det plan, i vilket planteringsenheten rör sig under framryckningen. 9. Aggregat enligt något av patentkraven 1-8, kännetecknat därav, att i framryckningsriktningen framför varje planteringsenhet är anordnat ett lämpligen frånslagbart eller tillfälligt ***undansvångbart*** markberedningsdon (22). 10. Aggregat enligt något av patentkraven 1-9, kännetecknat av förreglingsdon, som förhindrar jordtillförsel respektive frigöring av en planta i vänteläge om planteringsenheten intar ett för plantering olämpligt läge.
<endpatent>

The two claims (in figure 14 and figure 15) have some common terms, for example nouns as, ‘plantering’ (‘plant’), ‘skogsplantor’ (‘forest plants’), ‘rotklump’ (‘plant root ball’), ‘patentkraven’ (‘the claims’), the last word not being important for the context.

The content bearing words are hiding in productive compounds such as

- ‘markberedningsorgan’ (in claim 436822, figure 14) and ‘markberedningsdon’ (in claim 408121, figure 15) both words meaning ‘soil preparation unit’,
- ‘styrhylsa’ (in claim 436822, figure 14) and ‘styrorgan’ (in claim 408121, figure 15) both words meaning ‘guiding sleeve’,
- ‘planteringsorgan’ (in claim 436822, figure 14) and ‘planteringsenhet’ (in claim 408121, figure 15) both words meaning ‘device for planting or a planting unit’.

Synonyms are also frequently used by the authors of the claims. For instance in claim 436822 (figure 14) the author addresses the invention as ‘anordningen’ (‘the device’) while in claim 408121 (figure 15) the author addresses the invention as ‘aggregatet’ (‘the aggregate’). For more information about the two patent claims shown in figure 14 and 15, see appendix 1.

3.2 Morphological analyser used in this study.

SWETWOL is a program that performs automatic morphological analysis of Swedish text (Karlsson 1992). The program encompasses an entire description of Swedish inflectional morphology and was created to be used as a tool for basic morphological analysis in information indexing, retrieval and machine translation. The number of entries in SWETWOL is 48,000 and it constitutes the base forms of the core vocabulary of Standard Swedish. The vocabulary does not include productive compounds and derivatives. The theory which this program originates from is a two-level model using both lexicon and rules (Koskenniemi 1983). The model is implemented as small finite state automata. One of the limitations, according to Koskenniemi, is a restricted adequacy in the infixion and reduplication handling.

For instance, SWETWOL is able to identify compounds and mark the word boundaries with special signs but the interfix morpheme, which is used in production of creating new compounds, is only identified. The program does not remove the Swedish interfix morpheme nor does it restore the lemma that could have been compromised in the compounding process, all according to the Swedish phonological rules. (For a more exhaustive explanation of Swedish morphology concerning compound see section 2.5) SWETWOL marks different boundary strength with different signs – vertical line for weak compound borders ‘upp|visa’ (‘display’) and number sign for strong compound borders ‘bad#yta’ (‘bath surface’) (Aasa 2004), (Volk 1999).

SWETWOL only analysis compounds and could give several readings for ambiguous compounds. The analysis of unambiguous compounds causes no problems for SWETWOL. The compound ‘stålplåtmantel’ (‘steel plate casing’), for instance, is correctly decompounded as ‘stål#plåt#mantel’. However, for a compound such as ‘munstyckslängdaxel’ (‘nozzle longitudinal axis’), SWETWOL generates two suggestions, as shown in figure 16.

Figure 16: example of decompounding analysis

munstyckslängdaxel ‘nozzle longitudinal axis’				
1st lemma	2nd lemma		3rd lemma	4th lemma
mun ‘mouth’	styck (stycke) ‘part’	s (interfix morpheme)	längd ‘length’	axel ‘axis’
mun ‘mouth’	s (interfix morpheme)	tyck (tycka) ‘uphold’	slängd ‘clever’ or ‘thrown away’	axel ‘axis’

The first analysis above is the correct one. The problem is that SWETWOL does not show precedence to any of the analysis. For the purpose of my study, the analysis has to be deterministic, that is, the lemmatization process should not generate but one analysis for each lemma. Therefore, I have added two algorithms to the morphological analysis:

- a weighting and counting of different types of segmentation symbols attached to morphological analysis of compounds (Volk 1999), and
- a matching algorithm for elliptical uses of compounds, a heuristic decompounding algorithm (Andersson 2003).

The first compound analysis algorithm used to disambiguate the output from SWETWOL, henceforth referred to as **Volks algorithm**, contains three rules (Volk 1999):

1. If SWETWOL generates several decompounding suggestions and only one is a regular noun and the other is a derivation compound – select the regular noun!
2. If the decompounding suggestion contains both strong and weak boundaries (i.e. # and |), assign 4 points to a strong boundary, 2 points to a weak boundary.[20] The compound that gets the lowest rate will be selected. The idea is that the segmentation variant with the least internal complexity is to be preferred.
3. If two or more decompounding suggestions share the same lowest rate, a pair collection is used, where bad and preferred segment pairs are collected.

The algorithm gave 90% improvement when used on a German corpus (Volk 1999). However, since the bad/preferred distinction is manually executed, implementation of the third rule is not possible with the present study.

An alternative solution to the problem of decompounding suggestion is the heuristic algorithm. This algorithm was first explored with the intention to split compounds with low document frequency (Andersson 2003). The assumption was that it could be possible to exploit the ellipsis phenomenon (i.e. once a compound word has been introduced, the first component can be omitted subsequently) as a decompounding condition. If the rightmost part of a compound is present in the same text as the compound itself, then we assume that it is a productive compound with low term frequency (TF). The primary intention with the algorithm is to give the rightmost part a higher frequency value in the indexing, providing for more effectiveness in the search results.

The algorithm can be formally described thus:

> Take document *j* and sort all words *w* in *j* so that the shortest word is in the top of the list and then descending or use a reverse function so that the word order in document *j* becomes reversed. Then for all *w* in *j* test if w_x contains w_y as the right most part and if this is true then cut w_x by the number of w_y from right.

In my study the algorithm was only allowed to split the compounds that SWETWOL did not give a deterministic output, and that the first two rules of Volks algorithm were not able to disambiguate. Otherwise, the heuristic algorithm would yield several spurious decompounding suggestions.

Another problem, which I already have mentioned, is the interfix morphemes themselves. A part of a compound, providing it is not the right-most part, could be of the kind that it is not considered a lemma or a base form in Swedish. The compound 'orrspel' ('black cock's courtship display') for example, consists of the parts '*orr' ('black grouse') and 'spel' ('play'), where 'spel' is a lemma form and '*orr' deviates from the lemma form 'orre', lacking the last letter **e**. SWETWOL will not recognize '*orr' as a Swedish word. The interfix morphemes cause problems too. Three material related examples are 'gat**u**#väg**s**#korsning' ('road crossing'), 'sid**o**#väggs#murningen' ('lay of bricks on a side wall'), kedj**e**#sömnads#maskin ('machine of chain needlework'). To get a Swedish lemma form, one has to first identify the interfix morpheme (the bold letters in the examples above). In the above examples the part 'gatu' has to be converted to 'gata' ('road') to be accepted, 'kedje' has to be converted to 'kedja' ('chain') and 'sido' has to be converted to 'sida' ('side'), and the '**s**' morphemes in all examples have to be removed.

[20] and, in the original German implementation, 1 point to a derivation boundary (a boundary not adapted in the Swedish version)

3.3 The Parser used in this study.

As I have mentioned earlier, the word forms are often ambiguous in their Part-of-Speech (POS) and it is normally the context that makes them unambiguous (Schmid 1994). I used a Functional Dependency Grammar (FDG) parser to disambiguate words (Tapanainen 1999), (Tapanainen and Järvinen 1997), (Voutilainen 2001).

The FDG-parser is a computerized implementation of Tesniére's Dependency Theory[21]. One important element in the theory is the notion of syntactic-semantic nuclei instead of words. A nucleus could consist of several words (not necessarily adjacent to eachother), only one word or a part of a word. For example, 'would like' is a nucleus in the sentence 'What would you like me to do?' (Tapanainen 1999, p. 5).

The FDG-parser aims at being a full scale parser, (from tokenization to full syntactical dependency tree structure), for Swedish. The main component is the morphological analyser which is an extended version of Koskenniemi's two-level formalism (Voutilainen 2001), corresponding to SWETWOL. The parser makes use of a large lexicon, the morphological analyser, and a guesser for unknown words. The disambiguation function in the parser consists of hand-coded contextual constraint rules. The FDG-parser also selects one alternative when it comes to disambiguate compounds.

In Voutilainen's (2001) informal evaluation of the FDG-Parser (version 30.4.2001), the parser's ability to identify and link subjects, objects and subject complements to their regents (main verbs), was investigated. The test consisted of 6,149 words, 406 sentences from newspaper articles from *Hufvudstadsbladet* and *Dagens Nyheter*. The evaluation gave the precision ratios 98% for subject identifying and linking, 95% for object identifying and linking and SC 97% for subject complement identifying and linking, and recall ratios for S 92% for subject identifying and linking, 90% for object identifying and linking, 95% for subject complement identifying and linking.

Such good results would be out of range in my study. Both constraint grammar and dependency grammar has their main emphases on the lexicon (Ericksson and Gambäck 1997). Patent texts display very special sublanguages and special stylistic formats, and are quite different from news article texts. A platform which is presented in the project SVENSKA at SICS (Ericksson and Gambäck 1997) would have been a useful tool to test performance of different POS-taggers and parsers[22]. For instance, a subset of my collection has some common features with the bioinformatics field, and Gawronska and Erlendsson (2005) have presented a parser based on Categorial Grammar and Referent Grammar with good performance in the bioinformatics field. Over 70% of the sentences were parsed correctly when the parser was tested on a biological abstract from PubMed consisting of 14,090 words.

In addition, my material contained numerous OCR errors, which results in a disproportionate use of the guesser. For instance, when using the FDG parser on my material the conjuction 'eller' ('or') sometimes was identified as a verbal compound and decompounded to 'el#le' ('electrical smile'). Although these occurrences were rather sparse, the result was that the

[21] For further discussion see Tesniére, Lucien. 1959. Éléments de syntaxe structruale. Librairie Klinck-sieck, Paris.

[22] The aim with the SICS project was to develop a Swedish computational linguistic toolkit, with several possibilities to use different linguistic resources to a minimal cost.

entire phrase tree structure collapsed. Also the decompounding module, the FDG-parser provides for, created a lot of nonsense decompoundings. The electronics term 'rasterspår' ('screen trace') was analyzed as 'rast#erspår', the string '*erspår' being a non-word. The correct decompounding should be 'raster#spår'. The FDG parser performed inconsistent analyses, assigning different POS analyses to the same token. For instance, the word 'transportrullar' (transport cylinder) was repeatedly analyzed as both verb and noun within one and the same claim, when the correct assignment always is noun. Inconsistency in the POS and in the decompounding analyses affects the weight and retrieval method used in this study. For an example of a parsed sentence, see appendix 4.

3.4 Normalization issues

As I discussed in section 2.1.2 there are different ways to identify a token. Apart from the space character and the punctuation marks I regard xml-tags such as
 as token separators. I have chosen different strategies considering the question if the hyphen should qualify as a token separator or not, depending on the characters surrounding the hyphens.

Removing all hyphens *correctly* in the collection was not doable since chemical terms are very frequent in the collection. Chemical terms such as 'p--tenoyl-a-etyl-fenylättiksyra' ('p - tenoyl-a-ethyl-phenylacetic acid'), 'n--kloroetyl-n-dimetylamin' ('n - chloroethyl-n-dimethylamino') and 'dietyl--metyl--{--tenoyl-m-tolyl}' ('diethyl - methyl - (- tenoyl-m-tolyl)') are often or almost every time written with several hyphens. Therefore, the conclusion not to consider the hyphen a token separator is correct thinking. On the other hand, the compound parts in these terms are very much content bearing for the patent and should be separated in order to find similar inventions. An additional problem is the many marginal-splitting hyphens also present in the collection. In case the hyphen is a marginal-splitting hyphen the hyphened token should be considered a single token. Thus, the question arises as how to deal with the different types of hyphens.

I preserved the hyphens in the Baseline setting and decompounded the hyphenated terms when the decompounding modules recommend it in the other two decompounding settings.

To solve the marginal-splitting-hyphen issue I created an algorithm that uses the tag
:

> if word *x* is directly followed by a hyphen followed by one or several
 tags, then merge *x* and the next word.

Figure 17: example of different hyphened tokens in the collection

Raw data	Baseline setting	Type of hyphen
bränsle-huvudströmningspassage	bränslehuvudströmningspassage	Hyphenation caused by newline. The token is a four composite part compound.
för-a	föra	Hyphenation caused by newline. The token should be (and is) regarded as a single token.
förh-indras	förhindras	
fm-mottagare	fm-mottagare	Hyphenation according to Swedish hyphenation rules.
etyl-p—pyridylkarbonyt-hydratropat	etyl-p—pyridylkarbonyt-hydratropat	Hyphenation based on the chemical structure of the compound.
i-öppet	i-öppet	Probably OCR error.
som-bildar	som-bildar	
och-vilket	och-vilket	

The patent genre causes many domain specific hyphenation problems, in particular relating to electronic devices. Developing an algorithm which is able to distinguish between correctly hyphenated words such as 'OCH-krets' ('END circuits'), 'OCH-funktion' ('END function'), 'i-delen' ('the I divece') and 'f-lera' ('the f mud') and OCR errors such as '*och-vilket' ('*and who') and '*och-innehåller' ('* and content') is not a trivial issue. There are also other frequent terms in which upper case marking is salient (e.g. CHz-grupp 'CHz unit', D-kapsel 'D capsule', JK-minnesorgan 'JK memory unit') and it is difficult to find a good method which single out the important upper case units from the merely stylistic instances. I did not perform any normalization on these words. Those analyzed by SWETWOL were decompounded in the two decompounding settings.

Hyphenated tokens involving digits (e.g. 0,2-procentig '0.2-percentage', 90-prisma '90-prism', 17-ställigt '17-set', h2-elektrod 'h2 electrode', 3-väteatom '3 hydrogen atom') cause problems in the pre-processing stage. I removed all digits, since it is too difficult to determine when a digit is content bearing or not.

3.4.1 Corrupted data issues

As I mention in previous sections my material contained several OCR errors. The OCR errors could be divided into two groups real-word errors (e.g. OCR interpretation 'år' ('year') for 'är' ('are/is'), 'bargasen' ('?gas of bar') for 'bärgasen' ('carrier gas'), 'Därvid' ('at it') for 'David' (proper noun) and 'Liar' ('scythe') for 'Har' ('have/has')), and non-word errors (e.g. 'Mäldinlovt' for 'Mäldinlopp' ('grist inlet') and 'tillf5rselkälla' for 'tillförselkälla' ('supply source')).

The OCR error rate of the collection presumably exceeds 1–2%, given the massive correction work I had to perform. Not correcting the OCR errors would have seriously deteriorated the performance of the retrieval method.

For instance, SWETWOL decompounded OCR error affected compounds such as the OCR interpretation '*transportorrullar' for 'transportörrullar' ('conveyor cylinder') and generated nine different analyses, all of them creating spurious senses, as shown in figure 18.

Figure 18: example of decompounding analysis

Decompounding analysis of erroneous OCR interpretation		Spurious sense
1	transport#orr#rulle	transport#black grouse#cylinder
2	trans#port#orr#rulla	trance#gateway/doorway#black grouse#roll/weel
3	trans#port#orr#rulle	trance#gateway/doorway/black grouse#cylinder
4	trans#por#torr#rulla	train oil/whale oil#pore#dry#roll/wheel
5	trans#por#torr#rulle	train oil/whale oil#pore#dry#cylinder
6	Tran#sport#orr#rulla	train oil/whale oil#sports#black grouse#roll/wheel
7	Tran#sport#orr#rulle	train oil/whale oil#sports#black grouse#cylinder
8	Tran#spor#torr#rulla	train oil/whale oil#spore#dry#roll/wheel
9	Tran#spor#torr#rulle	train oil/whale oil#spore#dry#cylinder
Correct decompounding of the original compound word		Real sense
transportör#rulle		transporter#roll/wheel = conveyor idle

The Swedish word for 'black grouse' ('orre') loses its final vowel when not being the right-most part of the compound (see section 2.5.2). The bold **s**'s in alternative 4 and 5 are considered interfix morphemes by SWETWOL.

The FDG-parser decompounding module prevented decompounding of '*transportorrullar' since the FDG parser analyzes '*transportorrullar' as a spelling error with the marker *.

My method for unscrambling the OCR errors was to use a sub set of the unanalyzed words in SWETWOL to create a list of words to be manually corrected. I used the outcome of the manual correction as input to a program performing automatic correction.

3.5 The decompounding modulation

I carried out the study with three different test settings, one without decompounding and two with different decompounding solutions. There were two main runs for each setting – a first run without a stop list and a second run with a stop list. I used a genuine excluding stop list based on the 157 most frequent lemmas in the collection (the stop list removed approximately 46 to 48 different terms per document). In all three settings, lemmatization was performed by a Functional Dependency Grammar (FDG) parser. The reason I chose the FDG parser is that I wanted to use the same morphological analyzer (SWETWOL) for all three settings. The three test settings are:

- I considered the first test setting as the baseline and it will henceforth be referred to as the **Baseline setting**. In the Baseline setting, I used no decompounding module.

- In the second test setting (henceforth **FDG setting**) I used the decompounding module integrated in the parser, which means that no interfix morphemes were removed and that the correct lemmas for each part of the decompounded compounds were not identified. The FDG setting is the most exhaustive decompounding setting.

- In the third test setting (henceforth **Volk&Andersson setting**), I used two different algorithms to find the most plausible suggestions for splitting a compound. The first algorithm is developed by Volk (Volk 1999) and the second one, which I used as a complement to Volk's algorithm, is a heuristic algorithm developed by the author of this thesis (Andersson 2003).

SWETWOL generated more than one decompound segment for almost 100,000 different compounds. These compounds had to be analyzed by Volk's algorithm, since only one segment was to be used in the present study. Out of the 100,000 compounds, Volk's algorithm managed to choose one preferred segmentation for 90,000, and the heuristic algorithm managed to resolve half of the remaining unanalyzed compounds. This result was gained when I had optimized the heuristic algorithm so that the entire collection of successfully resolved decompounded segments was used as a prefer list, if the context of the claim analyzed did not suggest one segmentation only. If the two algorithms (Volk algorithm and the heuristic algorithm) could not select one decompound segment for a compound, the compound in question was treated as a single index term.

In the Volk&Andersson setting I also implemented a special module (henceforth the *find-lemma module*) for finding correct Swedish lemmas and remove the plausible interfix morphemes within the compounds. The find-lemma module is a complement to the SWETWOL analysis of the compound parts. In the find-lemma module I created an algorithm

which used the phonological rules for compounding backwards, so that it removed the interfix morpheme and tried to find the correct lemma by using a lexicon containing 300,000 lemma forms analyzed by SWETWOL.

To run the program in order to create the Volk&Andersson setting took 55 hours. This is a very time consuming task compared to running the program which created the FDG setting and the Baseline setting. It took 58 minutes to create the FDG setting and only four and a half minutes to create the Baseline setting.

Figure 19 shows the distribution of tokens, lemmas and decompounded compounds for the three test settings.

Figure 19: number of tokens, lemmas etc for each test setting

	Baseline	FDG	Volk&Andersson
Number of tokens	10,981,589	10,981,589	10,981,589
Number of lemmas	9,680,895	11,091,082	10,704,118
Number of different lemmas	301,667	157,661	137,996
Number of decompounded compounds	-	1,207,097	1,003,776
Number of different decompounded compounds	-	171,046	150,000

The reduction of the different lemmas between the Baseline setting and the other decompounding settings is explained by the performance of the decompounding module – decompounding will generate an increased number of lemmas, but a reduced number of different lemmas, since a compound often designate a very specific concept while the compound parts do not. A compound word like 'kompositionsutjämningsutrymme' ('composition equalizing space') is a very specific low-frequency term while each of its parts will occur many times in many different documents in many different domains. As a consequence of decompounding, the text will be stripped from a particular set of specific terms, but will at the same time expose content bearing "concepts" hiding within compounds to the bag-of-word retrieval method as shown in figure 20 (see section 3.6).

3.6 Retrieval model implementation

I implemented the Vector Space Model in Perl. There exist already implementations in Perl of the VSM, and my first intention was to use one of these implementations. But since I was not certain how much memory resources and CPU power that was needed, I chose to implement my own VSM module, an unsophisticated and a time-consuming model using a redundant index. First I computed the pair of vectors (search topic vector, claim vector) to obtain the scalar product and then divided them by their norms as it is explained by Widdows (2004, p. 158ff.). The term weighting method used as coordinations was the combination of TF-IDF.

Each search topic set (containing 100 claims selected as search topics) took approximately 45 minutes to run. Since I used three different test settings and three different search topic sets (see section 3.7 below), the entire task took approximately nine hours. When I used the stop list, containing 157 words, these nine hours were reduced to one and a half hour for each different setting.

From the VSM module output, the top 100 highest similarity values for each search topic in the search topic set was written to special files to be used in the evaluation process. The cut-off 100 was selected arbitrary since average value of claims classified by the same main group or subgroup was 1.3 claims. Using an absolute similarity value as a threshold was not possible, since the output similarity values for the 30,117 calculations were very unevenly distributed from 0.05 to 0.9 with an average value of 0.2.

Figure 20 displays the similarity calculation with cosine as normalization factor, between the search topic 436822 and the retrieved and relevant claim 408121, with the indexing method Stoplist for each test setting. The search topic and the claim are earlier used as examples (see figure 14 and figure 15, in section 3.1.1).

Figure 20: example of similarity calculation

Baseline.Stoplist.Cosine			FDG.Stoplist.Cosine			Volk&Andersson.Stoplist.Cosine		
Index term	Search topic weight values	Clam's weight values	Index term	Search topic weight values	Clam's weight values	Index term	Search topic weight values	Clam's weight values
			berednings	8.92	2.97	**beredning**	6.82	2.27
			fjäder	1.01	2.02	**fjäder**	1.02	1.02
förhindra	1.33	1.33	förhindra	1.33	1.33	förhindra	1.33	1.33
			klump	3.02	6.03	**klump**	3	6
komma	1.35	2.71	komma	2.63	1.32	komma	2.71	1.35
						krav;patent	1.26	1.26
ledbar	1.91	1.91	**ledbar**	1.91	1.91	**ledbar**	1.91	1.91
lämpligen	4.79	3.2	lämpligen	3.2	4.79	lämpligen	3.2	4.79
			mark	5.46	3.64	**mark**	5.46	3.64
			maskin	1.17	3.51	**maskin**	1.19	3.56
patentkrav	4.5	2.5	patent	2.47	4.44	patent	2.47	4.44
			plant	7.6	2.53			
planta	16	8	**planta**	10.53	18.44	**planta**	18.31	20.92
plantering	9.3	6.2	**plantering**	6.13	9.19	**plantering**	40.59	40.59
			planterings	36.2	33.18			
rotklump	7.56	3.78	**rot**	2.09	4.17	**rot**	2.37	4.74
						röra	1.24	1.24
skogsplanta	3.52	3.52	*skogs*	2.75	2.75	**skog**	2.75	2.75
styra	2.28	3.42	**styr**	1.73	4.32	**styr**	1.74	4.35
			styra	3.4	2.27	**styra**	3.34	2.22
Cosine normalization factor	2797.8		2766.3			3299.7		
Similarity value for the Stoplist indexing method	0.097		0.581			0.655		
Corresponding similarity values for the All-term indexing method	0.102		0.579			0.651		

As the figure 20 shows there is almost twice as many common index terms in the FDG setting and Volk&Andersson setting as for the Baseline setting. This is the effect of the decompounding module in each of these two settings. The bolded index terms could be

considered relevant for the content of the invention. The index terms in italic for the computation using the FDG setting (FDG.Stoplist.Cosine) still have their interfix morphemes attached to the root morpheme (the term'*plant' has the empty interfix morpheme and the other two terms have 's' as interfix morpheme). As I have mentioned earlier (see section 3.5) the interfix morpheme is not removed in the FDG setting. Therefore, the terms 'plantering' ('plantation') and 'bereding' ('preparation'), get lower values in the FDG setting compared to the Volk&Andersson setting.

There are two index terms, in italics in the table above, for the computation with the Volk&Andersson setting that are errors which occurred in the indexing process. The index term '*krav;patent' should have been 'patentkrav' ('patent claim'), and the index term 'röra' (meaning both 'mess' noun, 'move' verb) should have been 'rör' ('tube'). The first index term is presumably a program error from my part and the second index term 'röra' is a clear case of the erroneously identified lemma in the find-lemma module.

Although the document length for the search topic, in the Baseline setting, is 147 lemmas (the different terms length is 97 types) and for the claim the document length is 146 lemmas (the different terms length also 97 types), only 10 terms were common both for the search topic and for the claim. For both the FDG setting and Volk&Andersson setting the document lengths have increased. FDG setting generates 186 lemmas (96 types) for the claim and 167 lemmas (100 types) for the search topic. The corresponding values for Volk&Andersson setting are 179 lemmas (86 types) for the claim and 168 lemmas (98 types) for the search topic.

3.6.1 Modification of normalization factor and indexing method

I also modified the normalization factor in the VSM calculation by using the simple normalization factor which I presented in section 2.2.1, and also by dropping the normalization factor entirely, since there have been discussions about how the cosine normalization factor affects shorter documents.

Furthermore, Mase*, et al.* (2005) addresses the problem with the selection of index terms and stop list implementations in the patent genre (see section 2.4.3). To examine the question marks around the selection of index terms I implemented an alternative selection procedure of index terms by reusing the old theory by Luhn (see introduction section 2). In the selection process not only the most frequent words, like 'the' or 'of', but also words with very low frequency, were cut off (Schultz 1968). I implemented an indexing method that only uses 80 percent of the terms (the 80 percent were arbitrarily chosen) in each claim by removing the 10 percent most frequent words and the 10 percent least frequent words in the claims, (i.e. 20 percent of the terms in each claim were removed).

Mase also addresses the problem with finding a good term weighting method in the patent genre. The universal TF-IDF is perhaps not the most effective term weighting method for this genre. By decompounding and recreating lemmas, this study also addressed the term weighting method indirectly by increasing the TF and DF for certain words – the compounds. Compounds are usually nouns and could have a higher informative value than words in general, especially the context produced compounds, a sub class of productive compounds.

3.7 Selection of search topics and the evaluation

The IPC classification codes at the main group and sub group level were used to extract search topic sets and also as the assessors in the evaluation. Those claims that are assigned the same codes are considered to be relevant to a claim (which is used as a search topic) – they are the golden standard that the computations should try to retrieve.

3.7.1 Search topic sets

I selected 300 claims for the search topic sets. I used the IPC code information in order to obtain an even distribution within the collection, in such a way that I picked out

1. every 22nd of the 2,276 claims that were classified by one main or sub group code only,
2. every 22nd of the 2,200 claims that were classified by 5 different main or sub group codes,
3. every 4th of the 445 claims that were classified by 10 different main or sub group codes.

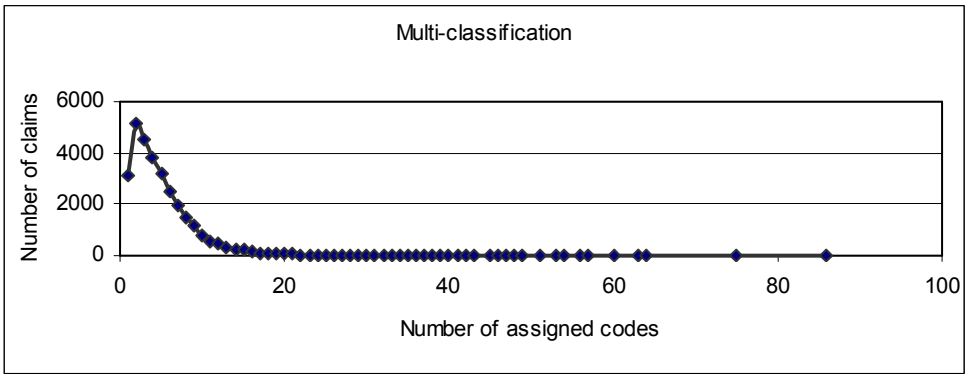
Henceforth, I will refer to the search topic sets as *UniqueIPC*, *5IPC* and *10IPC* respectively. Each search topic has been subjected to the same decompounding modulation with each test setting.

A claim will always be classified down to the IPC main group level in the collection. But not all claims are assigned the main group code of a sub group, since it is depending on the scope of the claim and that is why I chose to select the search topic set claims from both main and sub group levels. The number of IPC codes assigned to a patent reflects the coverage of the patent's scope. To select only those claims that have been assigned one IPC code, as search topic, would give a skew picture of the retrieval models performance since vocabulary will differ from those patent with a wider scope.

The IPC code and other patent classification system are created as search tool in order to help the patent engineers (and others) to find patent that are similar (see section 2.4.2). The aims with the IPC codes are to describe a patent content and the scope of the patent. If a patent is classified by three codes (red, blue and brown), the intersection of the patent captures the content of the patent. But the scope of the patent is still red, blue and brown. If we want to search for similar patents, we need to search in all codes (i.e. red, blue and brown), since a similar patent could well have only one or two codes in common.

The uneven distribution of the IPC codes in the collection made it more difficult to select candidates for search topic and to evaluate the output from the VSM module. In the collection almost 12,000 IPC main and sub group codes classify only one claim each, and more than 25,000 of the IPC codes at these levels (out of a total number of 33,106 main and sub group codes) classify no more than five claims each. Actually, when selecting the search topic set, I had to add the constraint that the search topic claims must share the main group code with at least one other claim, in order to avoid absence of assessors. On the other hand, one of the main group codes alone classifies no less 613 claims. The figure 21 illustrates the multi-classification (i.e. one patent document is classified at the same level by different codes) within the collection.

Figure 21: chart of multi-classification in the collection



The multi-classification, as I have already mentioned, made the selection of suitable search topics and evaluation of the result more difficult. As many as 5,092 claims are classified by two main group codes or sub group codes, 4 claims are classified by 60 codes and one claim is classified by 86 different codes at main group and sub group level. Roughly, the median multi-classification factor lands at 4 (the average value is lower due to the 12,000 codes only containing one claim), which means that typically, the claims are co-classified by four codes at the same level. Why will this make it more difficult to evaluate the output from the VSM module?

Only the IPC codes at the lowest level were used as assessors instead of human assessors in my study, and since one patent generally is classified by more than one IPC code this means that the relevant claim for a search topic could have more than one code in common with the search topic. For search topic set UniqueIPC this is not a problem since the search topic should only be assigned one code. But for 5IPC and 10IPC this is an existing problem, approximately 28 relevant claims for the 5IPC search topic have more than one code in common and for 10IPC the corresponding number is 32 relevant claims. The question is, how should this be reflected in the selected measurements recall, fallout, average precision and MAP? In the following sections I will describe how I dealt with multi-classification for each measurement.

3.7.2 Recall and average recall

I calculated recall traditionally for the retrieved documents per IPC code. Figure 22 shows a slightly simplified example.

Figure 22: evaluation of a search topic performance – recall and average recall

IPC codes	Retrieved and relevant claims	Golden standard of relevant claims	Retrieved claims (DN id:s)	Recall per IPC code	Average recall
G11B23/02	4	20	435661, 417142 416595, **407478**	20%	26%
G11B23/023	4	10	435661, 417142 416595, **407478**	40%	
B65D85/575	4	11	435661, **407478** 417142, 416595	36%	
G03B21/00	4	16	427878, 435661 416595, **407478**	25%	
G03B21/54	1	15	**407478**	7%	

The column for **retrieved claims** illustrates why the multi-classification phenomenon is important for how to carry out the evaluation. In the example search topic, one claim (marked in the figure in bold characters) was actually retrieved for all the classification codes. This would not have been reflected if I had calculated the recall for entire search topic without analysing the performance for each classification code.

Due to the extensive use of multi-classifications in the IPC, the recall values for the search topics in 5IPC and in 10IPC are computed differently than the recall value for the search topics in UniqueIPC. In order to reflect a search topic's performance in retrieving relevant claims for each of its codes, a recall value was first computed for each and every one of the search topic's codes, and from the recall values of the single codes an average recall value was computed for the search topic. Moreover, in order to visualize the multi-classification effect a "double quota factor" was introduced. The double quota factor reflects the multi-classification redundancy phenomenon, which inevitably occurs when using IPC codes as assessors. The double quota factor of the above example will be 12 (i.e. twelve times an observation of double noted). Appendix 1 contains more exhaustive documentation of the evaluation process for the search topic 436822 from UniqueIPC.

3.7.3 Fallout and average fallout

The fallout value measures how many non-relevant claims that were retrieved among the 100 highest similarity values. If a search topic gets the fallout value 1.0, this means that all the retrieved claims were of the type non-relevant. The fallout is based on the entire set of different relevant claims for a search topic, so no doubles are computed. If a search topic retrieves 32 relevant claims, the fallout value will be 0.68 (i.e.32/100). But, if there were five claims that would occur twice the fallout value would be 0.73 (i.e. (32-5)/100). Why use the fallout measurement?

For a user it is interesting to know both how many claims that were retrieved during a search session and how many of them were non-relevant claims, since it is a time-consuming task to swift for all documents. In view of the fact that only the 100 highest ranked claims, according to the similarity value, are retrieved for each search topic, the fallout value will indicate how many non-relevant claims the user has to go through.

3.7.4 Average precision and MAP

The traditional simple precision calculation is not applicable since the retrieved claims will always be 100, regardless of the number of relevant claims for each search topic.

I used mean average precision to evaluate the entire run for each search topic set. The mean average precision is calculated upon average precision which captures how the relevant claims are ranked in the retrieved set of claims and how many relevant claims that were retrieved (see section 2.3). The average precision is calculated iteratively in such a way that the performance (the number of relevant retrieved claims) is checked at every position in the ranking list where a new relevant claim is detected. Then, the number of relevant claims retrieved until this ranking position is divided by the ranking position value. This is repeated

until all the relevant claims are checked. The values are then added up and then divided by the golden standard set of relevant claims for a specific search topic.

Let us re-use our above example in figure 22 to illustrate the average precision calculation. Among the 100 retrieved claims, five different claims are relevant to this search topic (we disregard that some of these claims are represented in more than one classification code of the search topic). Figure 23 shows the ranking position of these five retrieved claims.

Figure 23: example of ranking list for retrieved and relevant claims

Ranking position	Retrieved and relevant claims (sharing at least one classification code with the search topic)
15	417142
63	416595
64	435661
85	427878
100	407478

A retrieved and relevant claim will be assigned the value 1. At the first ranking position at which a relevant claim is detected (the 15th is the first ranking position in the quoted example in figure 23), this value is divided by the ranking position value (i.e. 1/15). At each new such ranking position, 1 will be added to the number of relevant claims found until the preceding successful ranking position and the updated number of retrieved relevant claims will be divided by the ranking position value (i.e. 1/15, 2/63, 3/64, 4/85 and 5/100). These quotients will then be added up and divided by the number of golden standard relevant claims for the specific search topic.

In section 4 Result, I use an additional, average precision measurement – the interpolated average precision – to visualize special characteristics of a search topic when different modulations are used such as decompounding and indexing methods. In interpolated average precision calculation the ranking positions are fixed to be at every 10th position in the ranking list (i.e. average precision calculated for the ten highest similarity values, for the 20 highest similarity values, for the 30 highest similarity values and so forth, see figure 24).

Figure 24: example of Interpolated average precision calculation

Level	Calculation
10 retrieved	=7/10
20 retrieved	=12/20
30 retrieved	=18/30
40 retrieved	=18/40
50 retrieved	=18/50
60 retrieved	=18/60
70 retrieved	=18/70
80 retrieved	=18/80
90 retrieved	=18/90
100 retrieved	=18/100
SUM	3.87214285714286
Interpolated Ap	3.87214285714286/18 *ca* 0.215

4. Result

My presentation of the result is divided into three main parts – one for each search topic set (the search topics being classified at the IPC main group or sub group level either by one code (UniqueIPC), by five codes (5IPC) or by ten codes (10IPC)) – see section 3.7.1).

The main variable that I have modulated in the study is the decompounding variable, the Baseline setting being the test setting without decompounding. Each search topic set has been subjected to modulation with each decompounding setting. As I mention in section 3.6.1 I have also modulated the indexing methods and the normalization factors.

1. The retrieval model in the study is the Vector Space Model.

2. I used three different indexing methods. In the first indexing method, all terms are accepted as index terms. The second one is the Luhn indexing method – only 80% of the terms are accepted as index terms and in the third indexing method the stop list was used. Henceforth, I will refer to these indexing methods as *All-term indexing*, *Luhn indexing* and *Stoplist indexing* respectively. Since the Luhn indexing method did not perform as well as All-term and Stoplist, the computations with the Luhn indexing method will only be presented in the general table.

3. The normalization of the similarity values (i.e. the killing of the document length factor) was also carried out in different ways for the VSM – cosine normalization, simple normalization (i.e. only the claim's length is used in the calculation) and finally the complete elimination of the normalization calculation. Henceforth I will refer to the normalization approaches as *Cosine normalization*, *Simple normalization* and *No normalization* respectively.

The computation was carried out nine times for each different test setting. In the following, when referring to a specific computation, a dot notation is used. For example Baseline.All-term.Cosine refers to a computation where Baseline setting, All-term indexing and Cosine normalization have been used.

The evaluation was performed on main and sub group level (see section 2.4.2). For each individual search topic, average precision, fallout and recall was computed. The values which are presented in the general tables are average values for recall and fallout. The MAP (mean average precision) value is by its own an average value for evaluating an entire search topic set's average precision values. The tables will also display the median value since, as the diagrams will illustrate, the result is disparate. Some search topics generate very good results while some do not retrieve any relevant documents at all. The result will mainly be presented as statistical data in charts and tables. The main purpose of the charts and the tables is to illustrate the characteristics of a search topic set or of a specific search topic.

4.1 Search topic set UniqueIPC

The 100 search topics in UniqueIPC are classified by only one main group or sub group code. However, 12 of the search topics turned out to be classified by the same code as another of the search topics, i.e. the 100 search topics are classified by 94 different assessor codes. This type of co-occurrence was impossible to avoid, due to the sparseness in the collection of claims classified by a code with the UniqueIPC characteristics (see section 3.7). All in all, there are 100 main group or sub group codes used as assessors during the evaluation of UniqueIPC. Figure 25 shows the IPC section distribution for UniqueIPC, (for class distribution se appendix 5). There is no search topic classified by an IPC code in section D.

Figure 25: section distribution for search topic set UniqueIPC

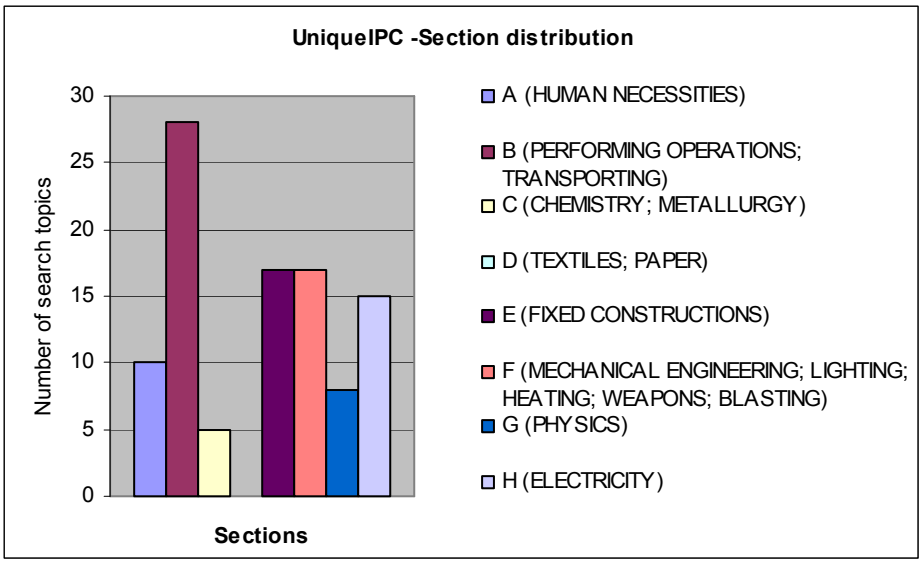


Figure 26 shows the average document length and the average number of different terms for the search topics, and the average document length and the average number of different terms for the golden standard relevant claims in UniqueIPC.

Figure 26: average length distribution for UniqueIPC and their golden standard relevant claims

	Baseline		FDG		Volk&Andersson	
	Document length	Number of different terms	Document length	Number of different terms	Document length	Number of different terms
Search topic	311	119	350	126	350	121
Golden standard relevant claims	312	112	358	119	358	113

As already discussed earlier in this thesis the average values give skew pictures of the content of the material. Some average values differ distinctly from the median values. For example, the median value for the golden standard relevant claims per search topic is 9 (while the corresponding average value is 15). This implies there are a few search topics with many relevant claims (there is one search topic that has 85 relevant claims) and these few search topics affect the average value. Another good illustration to the uneven distribution is the FDG setting values for the number of different terms in the golden standard relevant claims – the average value is 119, while the median value is as high as 284.

4.1.1 UniqueIPC – mean average precision

In figure 27, all 27 computations are presented.

Figure 27: MAP general table for UniqueIPC

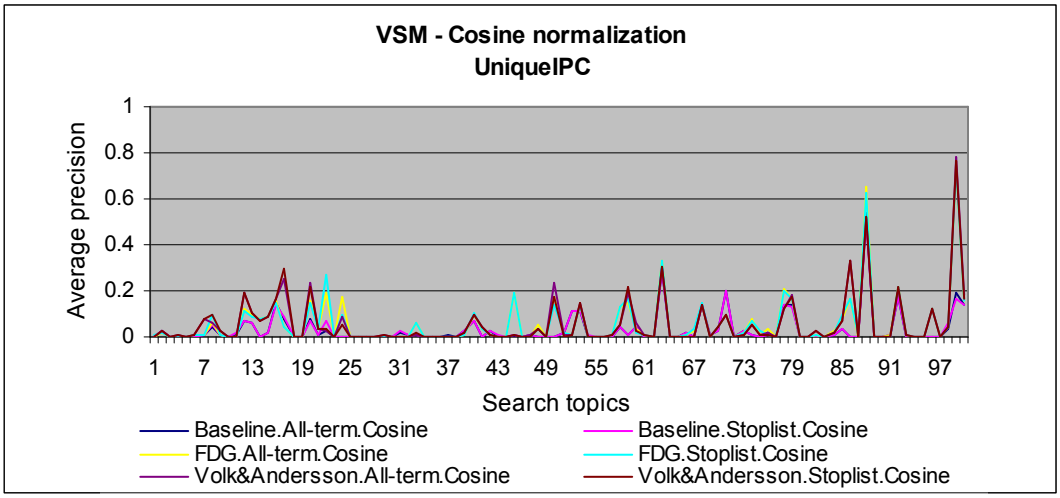
Mean average precision							
UniqueIPC							
Indexing method	Normalization	Baseline		FDG		Volk&Andersson	
		MAP	Median	MAP	Median	MAP	Median
All-term	Cosine	0.0349	0.0007	0.0583	0.0088	0.0576	0.0081
	No normalization	0.0257	0.0001	0.0312	0.0031	0.0321	0.0033
	Simple normalization	0.0349	0.0007	0.0539	0.0072	0.048	0.0069
Luhn	Cosine	0.0169	0	0.02	0	0.0158	0
	No normalization	0.0035	0	0.0016	0	0.0011	0
	Simple normalization	0.0161	0	0.0094	0	0.0047	0
Stoplist	Cosine	0.0328	0.0021	0.0576	0.0077	0.0569	0.0077
	No normalization	0.0289	0.0012	0.0352	0.0034	0.0346	0.0036
	Simple normalization	0.0337	0.0014	0.0548	0.0066	0.0467	0.0065

The figure indicates that when a decompounding module is used, a higher mean average precision value for the UniqueIPC set is generated. The best indexing method for the VSM seems to be All-term. None of the computations with the indexing method Luhn performs better than the computations with the other indexing methods, why is that? A plausible explanation is that the terms with low frequencies are important for the context and terms with a medium frequency are actually generally used in all type of patent documents.

These conclusions hold equally for the 5IPC search topic set and for the 10IPC search topic set. Computations with Simple normalization or No normalization generate overall lower average precision values. This conclusion holds for the 5IPC search topic set as well, and also for the 10IPC search topic set when it comes to the average precision and fallout values.

In figure 28 all computations with Cosine normalization and either indexing method All-term or Stoplist, are presented for the three test settings.

Figure 28: chart of AP for UniqueIPC



The figure 28 shows that some search topics did very well and others not so well. In figure 29 the search topics that generate the five highest average precision values and their recall values are presented.

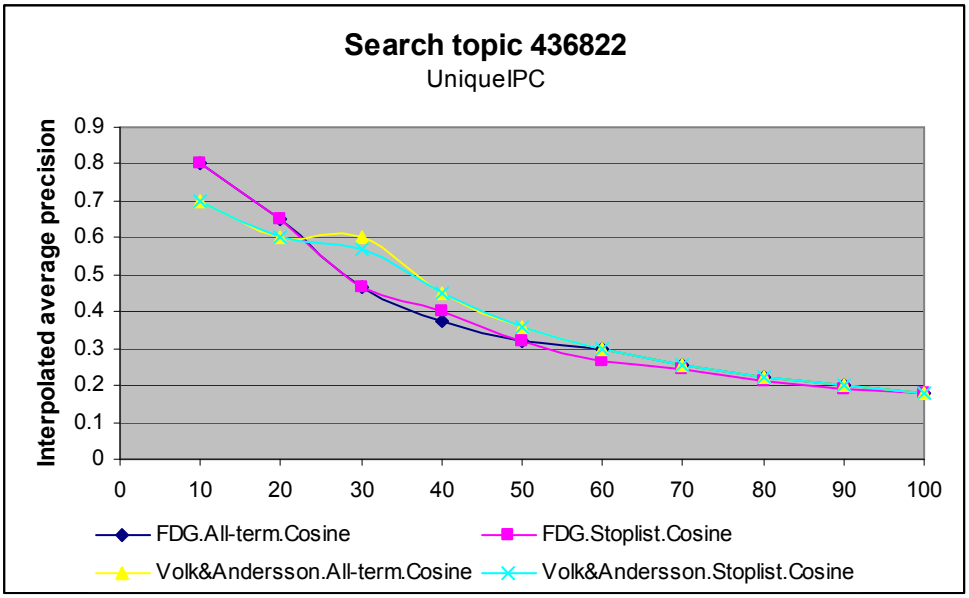
Figure 29: table of AP for UniqueIPC

The five highest average precision values for UniqueIPC													
Search topic	Golden standard relevant claims	Baseline				FDG				Volk&Andersson			
		All-term		Stoplist		All-term		Stoplist		All-term		Stoplist	
		AP	Recall	AP	Recall	AP	Recall	AP	Recall	AP	Recall	AP	Recall
436822	18	0.196	61%	0.169	50%	0.752	100%	0.743	100%	0.781	100%	0.765	100%
433688	6	0.542	100%	0.515	100%	0.648	100%	0.627	100%	0.513	100%	0.518	100%
423288	7	0.286	29%	0.286	29%	0.333	57%	0.334	57%	0.304	43%	0.304	43%
433512	3	0	0	0	0	0.167	33%	0.167	33%	0.333	33%	0.333	33%
409621	7	0.077	43%	0.090	43%	0.045	43%	0.043	43%	0.251	43%	0.295	43%

The computation Volk&Andersson.All-term.Cosine has the best average precision value – 0.781 for search topic 436822. For this search topic there are three more computations that generate average precision values over 0.7. Also the corresponding recall values are good – the recall is 100% which means that all 18 golden relevant claims are retrieved. When examining the fallout values for the computations this search topic has the fallout value 0.82 which is as good as it can get for this search topic.

Since all computations with FDG setting and Volk&Ansesson setting for the search topic 436822 generate the same recall value, it is interesting to visualize how the computations differ when it comes to how high up the retrieved and relevant claims are positioned in the ranking list. This is done by means of an interpolated average precision calculation (i.e. average precision calculated for the ten highest similarity values, for the 20 highest similarity values, for the 30 highest similarity values and so forth). The interpolated average precision calculation for search topic 436822 is shown in figure 30.

Figure 30: interpolated average precision for search topic 436822



From the figure 30 we can establish that FDG setting's computations found more relevant and retrieved claims among the 10 claims with the highest similarity values, than did the computations with Volk&Andersson setting (i.e. 8 golden relevant claims for the FDG setting's computations and 7 for the computations with Volk&Andersson setting).

Now, why is it that the Volk&Andersson setting anyway generates higher average precision than the FDG setting? This is explained by the fact that all 18 golden standard relevant claims are retrieved for the Volk&Andersson.All-term.Cosine computation already at the 29th ranking position, compared to a 59th position for the FDG.All-term.Cosine computation. When using the Stoplist indexing method the performance decreases (all relevant claims found at the 33rd position for the Volk&Andersson.Stoplist.Cosine computation and at the 94th position for the FDG.Stoplist.Cosine computation). However, as seen in figure 20 (see section 3.6) both computations with the FDG setting and the Volk&Andersson setting using the Stoplist indexing method have higher similarity values for the retrieved and relevant claim 408121 than for the corresponding computations using All-terms indexing method.

To summarize, the result so far vaguely indicates that the decompounding modules change the term weight factor for certain terms to the effect of increasing the similarity value. As a result the retrieved and relevant claims get higher positions in the ranking list. Furthermore, both for the general results and for the search topic 436822 it holds that the indexing method All-terms generates the best average precision value. This could indicate that terms in the stop list are necessary in the similarity calculation, since there are too few common terms if the stop list words are removed. Moreover, this is an illustration of the lexical sparseness in this type of document collection – there are not enough context bearing words even among claims classified by the same IPC code at main group and subgroup level. Also, what the decompounding does with the data indicates lexical or terms sparseness in the claims, since decompounding increases the number of different terms in the claims (see figure 26 in section 4.1).

4.1.2 UniqueIPC – fallout value

The fallout value measures how many non-relevant claims that were retrieved among the top 100 claims. For each computation an average fallout value is computed. The average fallout value is computed on the fallout value for each search topic. For each search topic the fallout value is calculated on the entire set of retrieved and relevant claims (no doubles are allowed).

If a search topic gets the fallout value 1.0. This means that all the retrieved claims were of the type non-relevant. The figure 31 shows the average fallout values and the median values for each computation.

Figure 31: fallout general table for UniqueIPC

Fallout per computations							
UniqueIPC							
Indexing method	Normalization	Baseline		FDG		Volk&Andersson	
		Average	Median	Average	Median	Average	Median
All-term	Cosine	0.98	0.99	0.97	0.99	0.97	0.99
	No normalization	0.99	0.99	0.98	0.99	0.98	0.99
	Simple normalization	0.98	0.99	0.98	0.99	0.97	0.99
Luhn	Cosine	0.99	1	0.99	1	0.99	1
	No normalization	0.99	1	0.99	1	0.99	1
	Simple normalization	0.99	1	0.99	1	0.99	1
Stoplist	Cosine	0.98	0.99	0.97	0.99	0.97	0.99
	No normalization	0.99	0.99	0.98	0.99	0.98	0.99
	Simple normalization	0.98	0.99	0.97	0.99	0.97	0.99

The figure shows that the average fallout values are very high, independently of type of modulation of the data. However, since the average of golden relevant claims for search topics in UniqueIPC is 9 claims, the fallout value would still be high even though all relevant claims were retrieved.

The five lowest fallout values ranging from 0.80 to 0.86 are hold by computations either with the FDG setting or Volk&Andersson setting. The computation that generated the overall lowest fallout value was Volk&Andersson.Stoplist.Cosine with 0.80 for search topic 420526. The search topic 420526 is classified in section F in the subclass covering cylinders, pressure vessels in general and sealing. For this search topic there are 32 golden relevant claims. Hence, the fallout value could have been 0.68 if all relevant claims were retrieved.

4.1.3 UniqueIPC – recall

For the search topic set UniqueIPC, the recall value is calculated traditionally, since every search topic is classified by only one IPC code. Figure 32 shows the recall values and median values for each computation.

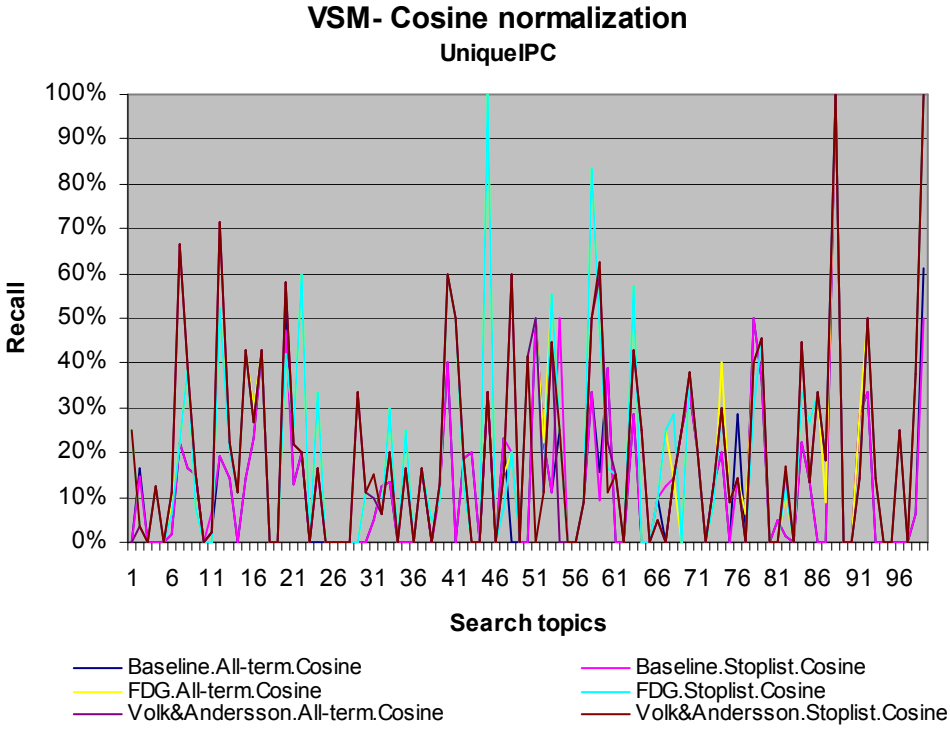
Figure 32: recall general table for UniqueIPC

Recall per computation							
UniqueIPC							
Indexing method	Normalization	Baseline		FDG		Volk&Andersson	
		Average	Median	Average	Median	Average	Median
All-term	Cosine	12.70%	3.50%	20.90%	12.50%	20.50%	14.30%
	No normalization	10.20%	0.70%	15.80%	11.80%	15.60%	11.10%
	Simple normalization	12.40%	1.90%	19.60%	13.70%	18.50%	14.30%
Luhn	Cosine	5.60%	0	7.30%	0	4.90%	0
	No normalization	3.20%	0	2.50%	0	1.70%	0
	Simple normalization	5.20%	0	5.30%	0	3.60%	0
Stoplist	Cosine	13.10%	7.90%	20.60%	12.20%	20.20%	14.30%
	No normalization	11.20%	4,0%	16.70%	11.10%	16,0%	10.60%
	Simple normalization	13.10%	5.50%	19.40%	12.90%	18.70%	13.10%

The result for the recall measurement displays a relation between the performances of the different computations which corresponds to the result for the mean average precision measurement.

In figure 33 all computations with Cosine normalization and either indexing method All-term or Stoplist, are presented for the three test settings. In table below the chart the search topics that receive the five highest recall values and their average precision values are presented.

Figure 33: chart and table of recall for UniqueIPC



Search topic	Golden standard relevant claims	Baseline				FDG				Volk&Andersson			
		All-term		Stop list		All-term		Stoplist		All-term		Stoplist	
		Recall	AP	Recall	AP	Recall	AP	Recall	AP	Recall	AP	Recall	AP
433688	6	100%	0.542	100%	0.515	100%	0.648	100%	0.627	100%	0.513	100%	0.518
436822	18	61%	0.196	50%	0.169	100%	0.752	100%	0.743	100%	0.781	100%	0.765
438074	5	40%	0.140	40%	0.140	80%	0.160	80%	0.160	100%	0.184	80%	0.169
417047	3	33%	0.004	33%	0.004	100%	0.195	100%	0.194	33%	0.009	33%	0.010
420515	6	33%	0.046	33%	0.046	83%	0.129	83%	0.129	50%	0.056	50%	0.056

The figure shows that recall for the search topics differ extensively from topic to topic, some search topics did very well and others did not retrieve any relevant claims among the top 100 retrieved claims. There is one search topic (433688) that generates top recall values independently of indexing method or the presence or absence of a decompounding module. However, the computations using the FDG setting generate higher average precision values.

The results for the three other search topics that get a recall value of 100% for one or more computations confirms what I have previously stated (and which also holds for the other search topic sets), namely that:

- a decompounding module is decisive,
    - the type of decompounding module may be important,
- the lexical data sparseness of the claims is decisive.

For instance, the results for search topic 417047 indicates that the decompounding module used in the FDG setting changes the weight for the terms (TF and DF, see section 2.2) in such a way that recall is improved for the search topic. Decisive for this search topic is also its relatively considerable length and its relatively large set of different terms, whereas the indexing method does not alter the recall value or the average precision value significantly.

To summarize UniqueIPC, the FDG decompounding setting will help retrieving all relevant claims owing to the more extended decompounding. At the same time, some of the results indicate that the FDG setting decompounds too much since the average precision value tends to be lower for FDG than for the Volk&Andersson setting, given they have the same recall value. The results speak vaguely in favor of the FDG setting, but it could have been the other way round had the comparable claims contained more common terms.

4.2 Search topic set 5IPC

Each of the100 search topics in 5IPC is classified by five different main group or sub group codes. However, 52 of the codes turned out to be common to two, three or four other search topics, i.e. the 100 search topics are classified by 468 different assessor codes. The multiple occurrences were impossible to avoid, due to the extensive use of multi-classification in the IPC-system. All in all, there are 500 main group or sub group codes used as assessors during the evaluation of 5IPC. Figure 34 shows the IPC section distribution for 5IPC, (for class distribution see appendix 5).

Figure 34: section distribution for search topic set 5IPC

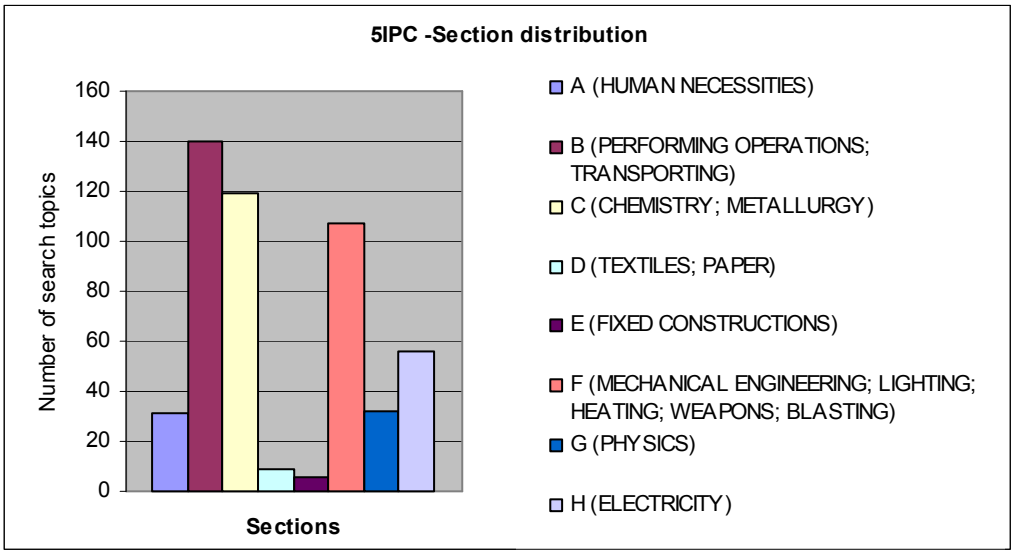


Figure 35 shows the average document length and the average number of different terms for the search topics, and the average document length and the average number of different terms for the golden standard relevant claims in 5IPC.

Figure 35: average length distribution for 5IPC and their golden standard relevant claims

	Baseline setting		FDG setting		Volk&Andersson setting	
	Document length	Number of different terms	Document length	Number of different terms	Document length	Number of different terms
Search topic	338	120	388	128	388	120
Golden standard relevant claims	313	109	362	119	109	104

The median search topic length differs from the average value, the median search topic being 66 to 77 terms shorter, while the average number of different terms for the search topics does not differ as much from the median number. For the golden standard relevant claims there is a difference between the average and the median document length. For the number of different terms in the golden standard relevant claims, there is a clear difference in the FDG setting, where the average value is 119 and the median is 279. The average value is decreased due to a few claims with very few different terms.

4.2.1 5IPC – mean average precision

In figure 36 all 27 computations are presented for search topic set 5IPC.

Figure 36: MAP general table for 5IPC

Mean average precision							
5IPC							
Indexing method	Normalization	Baseline		FDG		Volk&Andersson	
		MAP	Median	MAP	Median	MAP	Median
All-term	Cosine	0.0473	0.0171	0.0687	0.0332	0.0668	0.0262
	No normalization	0.0277	0.0098	0.0291	0.0104	0.0279	0.0097
	Simple normalization	0.0409	0.0145	0.0509	0.0231	0.0477	0.017
Luhn	Cosine	0.0238	0.0059	0.0282	0.0081	0.0209	0.0042
	No normalization	0.0098	0.0011	0.0064	0.0012	0.0043	0.0004
	Simple normalization	0.0178	0.0045	0.0139	0.0049	0.0095	0.001
Stoplist	Cosine	0.0503	0.0176	0.0681	0.0314	0.0649	0.0222
	No normalization	0.0343	0.0105	0.0313	0.0116	0.0293	0.0094
	Simple normalization	0.0457	0.0148	0.0512	0.0217	0.0466	0.0168

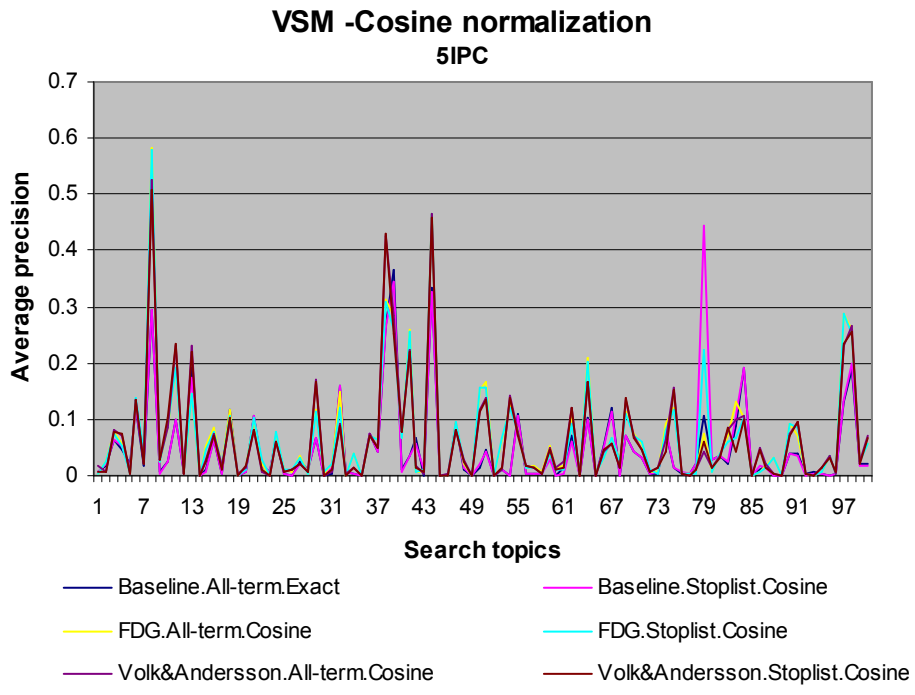
The figure indicates that when a decompounding module is used, a higher mean average precision value for the 5IPC set is generated. For instance, the MAP for the computation Baseline.All-term.Cosine increases with 41% when using the Volk&Andersson setting and with 45% when using the FDG setting.

As for UniqueIPC, the 5IPC results indicate that the decompounding modules change the term weight factor for certain terms, and this will affect the similarity value in a positive way. As a result the retrieved and relevant claims get higher positions in the ranking list and more relevant claims are also retrieved. The best indexing method for Baseline setting has change from All-term to Stoplist, whereas for the FDG setting and Volk&Andersson setting the best indexing method still is All-term, since the method generates slightly higher MAP values.

Another interesting observation is that the MAP value increases for the search topics in 5IPC compared with the search topics in UniqueIPC. The best computation in UniqueIPC generate a MAP value of 0.0583 while the MAP value for 5IPC has increase to 0.0687, although it is still the same computation performing best in both UniqueIPC and 5IPC – FDG.All-term.Cosine. Why do search topics in 5IPC perform better than search topics in UniqueIPC? There are two reasonable explanations; firstly that the search topics in 5IPC have more relevant claims than those in UniqueIPC, secondly that the relevant claims for the search topic in 5IPC generally share more than one IPC code with the search topic in question and this could be reflected by the vocabulary used.

In figure 37 all computations with Cosine normalization and either indexing method All-term or Stoplist, are presented for the three test settings. In the table below the chart the search topics that generate the five highest average precision values and their recall values are presented. Remember, this recall value for each search topic must be an average value, since each search topic is classified by five codes. Average recall is abbreviated as Arecall in the table.

Figure 37: chart and table of AP for 5IPC



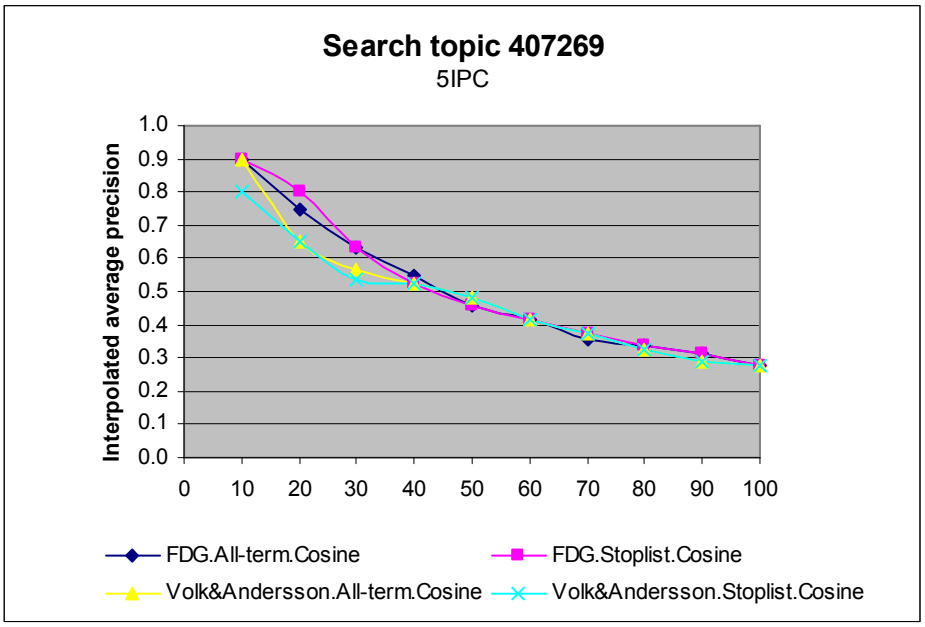
Search topic	All golden different relevant claims	Baseline				FDG				Volk&Andersson			
		All-term		Stoplist		All-term		Stoplist		All-term		Stoplist	
		AP	Arecall	AP	Arecall	AP	Arecall	AP	Arecall	AP	Arecall	AP	Arecall
407269	38	0.294	54%	0.295	54%	0.582	81%	0.580	81%	0.527	81%	0.507	77%
424338	57	0.334	69%	0.326	72%	0.466	85%	0.466	85%	0.465	85%	0.459	85%
419575	44	0.266	58%	0.264	58%	0.313	74%	0.309	74%	0.426	78%	0.432	82%
432599	180	0.105	21%	0.446	48%	0.075	15%	0.223	37%	0.043	20%	0.059	23%
420779	55	0.366	68%	0.344	72%	0.284	71%	0.260	71%	0.286	76%	0.252	73%

The figure shows that some search topics did very well and others not so well. High average precision values indicate that the retrieved and relevant claims have high positions in the ranking list.

The computation FDG.All-term.Cosine has the best average precision value – 0.582 for search topic 407269. For this search topic there are three more computations that generate average precision values over 0.5. Also the corresponding average recall (Arecall) values are good – the recall is 81%. The double quota value is 10.5. Practically this means that 28 different relevant claims were retrieved and when computing average recall 5 claims occurred more than once. Notice, that the best average precision value for a single search topic in 5IPC has decreased compared with the best value for a single search topic in UniqueIPC (i.e. 0.781 hold by Volk&Andersson.All-term.Cosine).

All computations with FDG and Volk&Andersson settings for the search topic 407269 retrieve the same number of different relevant claims. Figure 38 visualizes how the computations differ when it comes to how high up the retrieved and relevant claims are positioned in the ranking list. This is done with an interpolated average precision (by document level at every 10^{th}).

Figure 38: interpolated average precision for search topic 407269



From the figure we can see that the FDG setting computations rank the relevant and retrieved claims higher than the computation with Volk&Andersson setting.

4.2.2 5IPC – fallout value

The fallout value measures how many non-relevant claims that were retrieved amongst the top 100 claims. Figure 39 shows the average fallout values and the median values for the 5IPC for all 27 computations. Remember, the average fallout value is computed on the fallout value for each search topic. For each search topic the fallout value is calculated on the entire different set of retrieved and relevant claims, (no doubles are allowed).

Figure 39: fallout general table for 5IPC

Fallout per computations								
5IPC								
Retrieval model	Indexing method	Normalization	Baseline		FDG		Volk&Andersson	
			Average	Median	Average	Median	Average	Median
VSM	All-term	Cosine	0.93	0.95	0.91	0.94	0.91	0.94
		No normalization	0.94	0.96	0.94	0.95	0.94	0.96
		Simple normalization	0.93	0.95	0.92	0.94	0.92	0.94
	Luhn	Cosine	0.95	0.97	0.95	0.97	0.96	0.97
		No normalization	0.97	0.99	0.97	0.99	0.98	0.99
		Simple normalization	0.96	0.98	0.96	0.98	0.97	0.98
	Stoplist	Cosine	0.93	0.95	0.91	0.94	0.91	0.94
		No normalization	0.93	0.96	0.93	0.95	0.93	0.95
		Simple normalization	0.93	0.95	0.92	0.94	0.92	0.95

The figure shows that the average fallout values are very high, independently of type of modulation of the data, although the average of golden different relevant claim per search

topic is 88 claims. Yet, the fallout values for each computation for 5IPC have decreased compared with corresponding computation for UniqueIPC.

When examining the search topics that retrieved the lowest fallout values these values are obtain by computations either with the FDG setting or Volk&Andersson setting. However, the lowest fallout value was generated by the computation Baseline.Stoplist.Cosine with 0.46 for search topic 432599. The relatively low fallout value for this non-decompounding computation is probably explained by the fact that the claims of search topic 432599 happen to be chemistry claims with a high frequency of chemical compounds. Moreover, for this specific search topic there are 180 different golden relevant claims. Hence, the fallout value could have been much lower, even zero, had the computation performed better.

4.2.3 5IPC – recall value

The recall presented in the figure 40 shows an average value and median value for the entire search topic set for all 27 computations. Remember, this value is based on each search topics average recall value, since each search topic is classified by five codes. Recall is first calculated for each of the five codes of a search topic, and then an average value for the search topic is calculated from these five values (see section 3.7.2). Subsequently, an average value for a computation is calculated from the average values of each search topic.

Since each code for the search topics is computed individually, doubles (i.e. the search topic sharing one or more codes with the relevant claims in the target collection) are accepted. To visualize the multi-classification, an average double quota factor is also presented in the figure.

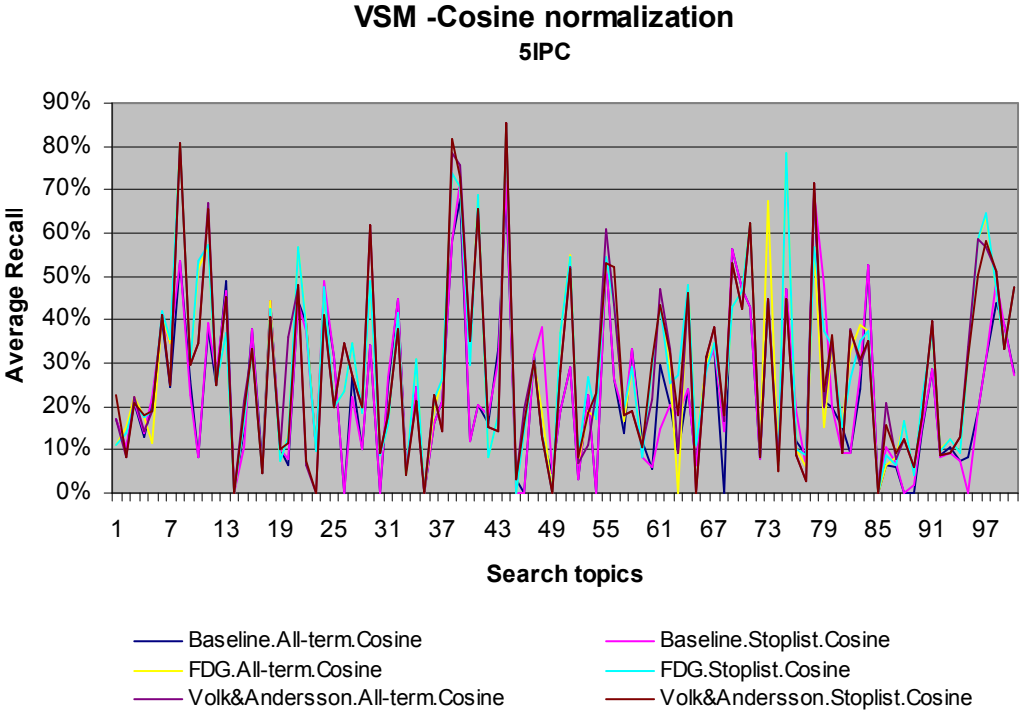
Figure 40: average recall general table for 5IPC

Average Recall for each computation										
5IPC										
Indexing method	Normalization	Baseline			FDG			Volk&Andersson		
		Average	Median	Double	Average	Median	Double	Average	Median	Double
All-term	Cosine	22.80%	20%	6	29.20%	27%	8	28.40%	23%	8
	No normalization	19%	14%	5	21.80%	17%	6	21.70%	17%	6
	Simple normalization	22.60%	19%	6	28.40%	26%	7	26.20%	23%	7
Luhn	Cosine	17.10%	12%	4	16.70%	12%	4	13.90%	11%	3
	No normalization	11.40%	8%	2	9.90%	7%	2	7.50%	4%	1
	Simple normalization	15%	10%	4	14.20%	12%	4	11.40%	9%	3
Stoplist	Cosine	23.60%	20%	6	29.40%	27%	8	28.40%	23%	8
	No normalization	21.50%	19%	6	22.50%	19%	6	22.60%	18%	6
	Simple normalization	23.60%	21%	7	28.30%	27%	7	26.40%	23%	7

The result for the recall measurement displays a relation between the performances of the different computations which corresponds to the result for the MAP measurement, i.e. the results indicate that a higher mean average recall value for the 5IPC set will be generated when a decompounding module is used. Even though the mean average recall is computed from each search topic average recall values all computations for 5IPC perform better compared to the corresponding computation for UniqueIPC .

In figure 41 all computations with Cosine normalization and either indexing method All-term or Stoplist, are presented for the three test settings. Below the figure, the search topics that receive the five highest average recall values and their average precision values are presented.

Figure 41: chart and table of Arecall for 5IPC



Search topic	All golden relevant claims	All golden different relevant claims	Baseline				FDG				Volk&Andersson			
			All-term		Stoplist		All-term		Stoplist		All-term		Stoplist	
			Arecall	AP	Arecall	AP	Arecall	AP	Arecall	AP	Arecall	AP	Arecall	AP
424338	140	57	69%	0.334	72%	0.326	85%	0.466	85%	0.466	85%	0.465	85%	0.459
419575	95	44	58%	0.266	58%	0.264	74%	0.313	74%	0.309	78%	0.426	82%	0.432
407269	100	38	54%	0.294	54%	0.295	81%	0.582	81%	0.580	81%	0.527	77%	0.507
431769	41	23	47%	0.014	47%	0.014	78%	0.120	78%	0.117	45%	0.155	45%	0.154
432384	21	16	69%	0.021	69%	0.021	57%	0.007	57%	0.005	71%	0.011	71%	0.010

The figure shows that average recall for the search topics differ extensively from topic to topic. When comparing the different computations for one specific search topic the average recall value tend to be higher for the computations using a decompounding module. However, there are two interesting search topics (431769 and 432384) that do not correspond to the general indication. For both of these search topics, one of the decompounding modules performs clearly better than the other two settings.

For search topic 407269, which was subject for discussion in section 4.2.1, the Volk&Andersson.Stoplist.Cosine computation average recall value is 77%, compared to 81% for the other decompounding module computations, but it still retrieves the same number of different relevant claims (28) as the other computations. This is explained by the multi-classification (see section 3.7.1) occurring among the golden relevant claims for this search topic.

In figure 42 all recall values for the search topic 407269 (i.e. all five codes and both the FDG setting and the Volk&Andersson setting) are presented.

Figure 42: each IPC codes recall values for search topic 407269

	FDG setting		Volk&Andersson setting	
IPC code	All-term	Stoplist	All-term	Stoplist
F16D65/097	92%	92%	92%	86%
F16D65/092	80%	80%	80%	76%
F16D55/224	83%	83%	83%	79%
F16D65/02	67%	67%	67%	62%
F16D55/22	83%	83%	83%	80%
Arecall	81%	81%	81%	77%

To summarize 5IPC, the FDG decompounding setting will help retrieving more relevant claims owing to the more extended decompounding. At the same time, some of the results indicate that the FDG setting decompounds too much since the average precision value tends to be lower for FDG than for the Volk&Andersson setting, given they have the same fallout value (the recall value is not comparable since it accepts doubles). Another observation is that the best performing search topic generally is classified by codes belonging to the same section, even down to subclass level, as shown in figure 42.

Although the 5IPC tend to indicate same results as UniqueIPC there is a slight difference. While 5IPC generates overall best results for the entire search topic set for all three measurements the single highest recall value or average precision value is hold by computations with the UniqueIPC set.

4.3 Search topic set – 10IPC

Each of the 100 search topics in 10IPC is classified by ten different main group or sub group codes. The number of codes that appeared in more than one search topic was 100 (i.e. the 100 search topics are classified by 900 different assessor codes). This multiple occurrence was impossible to avoid, due to the extensive use of multi-classification in the IPC-system. All in all, there are 1000 main group or sub group codes used as assessors during the evaluation of 10IPC. Figure 43 shows the IPC section distribution for 10IPC, (for class distribution see appendix 5).

Figure 43: section distribution for search topic set 10IPC

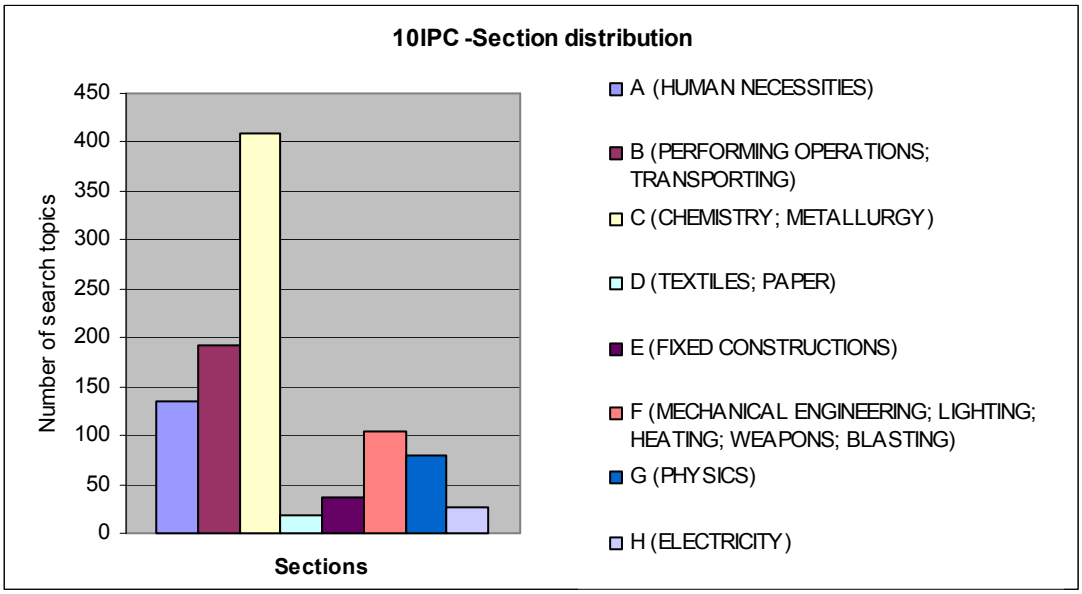


Figure 44 shows the average document length and the average number of different terms for the search topics, and the average document length and the average number of different terms for the golden standard relevant claims in the search topic set 10IPC.

Figure 44: average length distribution for 10IPC and their golden standard relevant claims

	Baseline setting		FDG setting		Volk&Andersson setting	
	Document length	Number of different terms	Document length	Number of different terms	Document length	Number of different terms
Search topic	324	115	374	123	374	114
Golden standard relevant claim	289	101	336	110	336	99

Still the median search topic length differs from the average value. The median values for all settings decrease with as much as 100 terms. However, the set of different terms for the search topics does not differ as much – the medians are almost the same as the average values. For the golden relevant claims there is a difference between the average values and median values but not for the set of different terms, except for the golden relevant claims with the setting FDG, where the median is 243 and the average value is only 110, which is due to few claims with very few different terms.

4.3.1 10IPC – mean average precision

In figure 45 all 27 computations are presented.

Figure 45: MAP general table for 10IPC

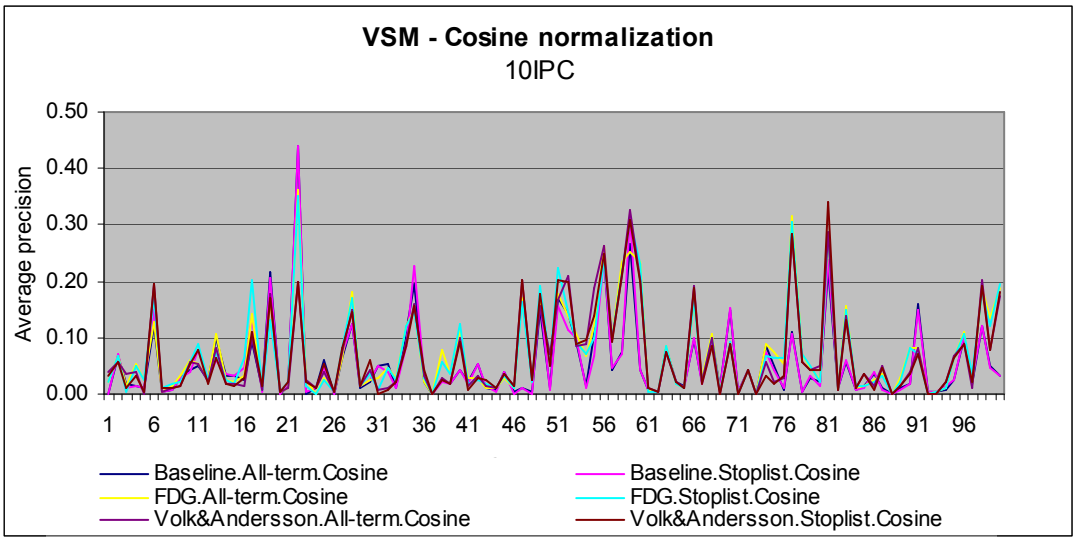
Mean average precision							
10IPC							
Indexing method	Normalization	Baseline		FDG		Volk&Andersson	
		MAP	Median	MAP	Median	MAP	Median
All-term	Cosine	0.0552	0.0242	0.0732	0.0342	0.0688	0.0373
	No normalization	0.0397	0.0187	0.0432	0.0194	0.0406	0.0161
	Simple normalization	0.0493	0.0255	0.0621	0.0287	0.0590	0.0256
Luhn	Cosine	0.0397	0.0187	0.0401	0.0216	0.0337	0.0132
	No normalization	0.0213	0.0061	0.0186	0.0047	0.0145	0.0015
	Simple normalization	0.0374	0.0197	0.0280	0.0112	0.0225	0.0055
Stoplist	Cosine	0.0552	0.0275	0.0732	0.0394	0.0681	0.0343
	No normalization	0.0432	0.0228	0.0470	0.0220	0.0436	0.0170
	Simple normalization	0.0509	0.0250	0.0645	0.0301	0.0590	0.0260

The figure indicates that when a decompounding module is used, a higher mean average precision value for the 10IPC set is generated. For instance, the mean average precision for the computation Baseline.All-term.Cosine increase with 25% when using the Volk&Andersson setting and with 33% when using the FDG setting.

As for other two search topic sets, the trend for 10IPC still is decompounding modules change the term weight factor for certain terms, and this will affect the similarity value in a positive way. The best indexing method for Volk&Andersson is All-term, whereas for the FDG setting and Baseline setting both indexing methods generate the same MAP values. Again the best MAP value increases for 10IPC compared with other two search topic sets.

In figure 46 all computations with Cosine normalization and either indexing method All-term or Stoplist, are presented for the three test settings.

Figure: 46 chart of AP for 10IPC



Even though, results for 10IPC are similar to 5IPC and UniqueIPC, that the decompounding modules will affect the average precision values positively. The figure 46 shows that the Baseline setting's computations generate higher value for some search topic than the decompounding settings.

In the figure 47 the search topics that generate the five highest average precision values and their average recall values are presented.

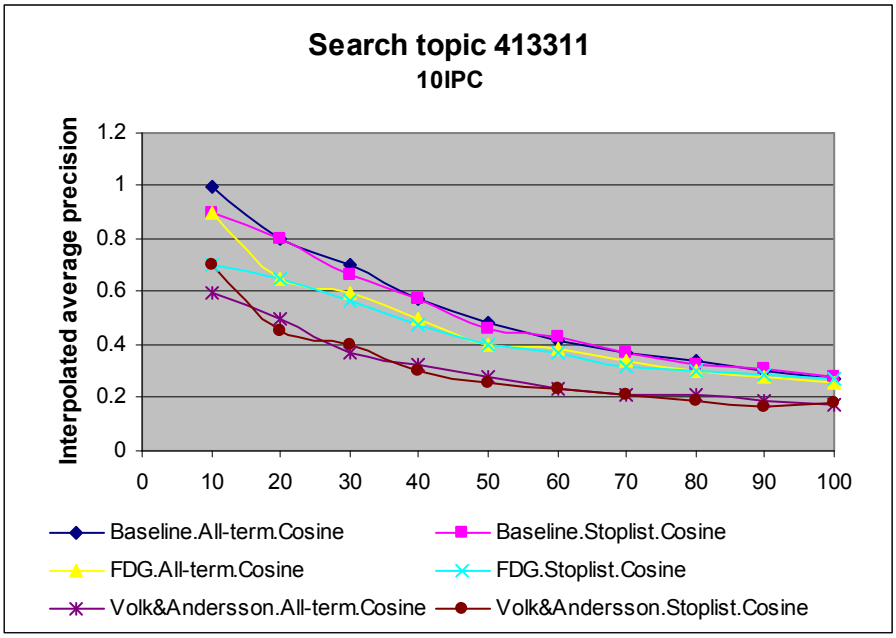
Figure: 47 table of AP for 10IPC

The five highest average precision for 10IPC													
Search topic	All golden different relevant claims	Baseline				FDG				Volk&Andersson			
		All-term		Stoplist		All-term		Stoplist		All-term		Stoplist	
		AP	Arecall	AP	Arecall	AP	Arecall	AP	Arecall	AP	Arecall	AP	Recall
413311	50	0.441	67%	0.439	67%	0.362	66%	0.349	67%	0.200	60%	0.198	59%
431208	537	0.219	18%	0.236	18%	0.281	17%	0.287	16%	0.286	17%	0.339	19%
425388	816	0.264	33%	0.314	38%	0.253	29%	0.310	37%	0.325	39%	0.310	38%
430075	139	0.111	22%	0.108	23%	0.316	48%	0.307	44%	0.284	45%	0.285	42%
424237	89	0.242	49%	0.240	48%	0.261	61%	0.237	58%	0.263	61%	0.249	59%

The best performing search topic 413311 retrieved the highest average precision value with a computation using the Baseline setting. All of the search topic IPC codes belong to section C (chemistry). This differ the 10IPC from the two other search topic sets, where either a computation with FDG setting or Volk&Andersson setting generated the single highest average precision value. Again, even if the best MAP value increases for 10IPC, the best single highest average precision value has decreased compared with the other two search topic sets.

Figure 48 shows an interpolated average precision calculation (by document level at every 10^{th}) for search topic 413311.

Figure 48: interpolated average precision for search topic 413311

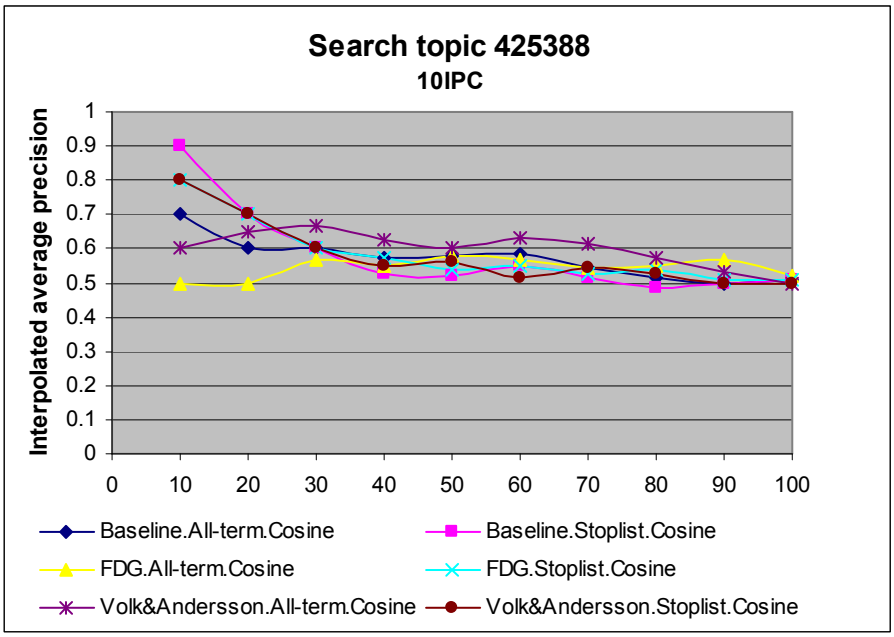


The figure 48 shows that a decompounding module will negatively affect the ranking position of the relevant and retrieved claims. Moreover, a more restricted decompounding module deteriorates the results much more. In other words – either you should not decompound anything or you should decompound everything.

From the computations with Volk&Andersson setting and FDG setting we can conclude that performance for this search topic decreases when a decompounding module is used, but why the exceptional deterioration with the Volk&Andersson setting? One reason could be that the lemmas re-created by the Volk&Andersson find-lemma module from the decompounded parts coincide with lemmas of different semantic features, and as a consequence the term weight will get biased.

For search topic 425388 on the other hand, all computations with all three settings (only Cosine normalization) retrieve almost the same number of different relevant claims. In figure 49 the interpolated average precision (by document level at every 10th) for this search topic is presented. Search topic 425388 is a good example of how decompounding combined with indexing method can influence the ranking position of relevant and retrieved claims.

Figure 49: interpolated average precision for search topic 425388



Among the 10 highest ranked claims Baseline.Stopplist.Cosine retrieves most relevant claims of all computations with 9 relevant claims. At the same time, the highest average precision value for the search topic 425388 is generated by Volk&Andersson.All-term.Cosine (0.325). The graph shows that Volk&Andersson.All-term.Cosine generates more newly-retrieved claims for every 10th document level.

Another aspect with search topic 425388 is that not all golden standard different relevant claims where possible to retrieve since the cut-off threshold for retrieved claims was too low (i.e. the golden standard set of relevant claims for this search topic is 816 and all in all also accepting doubles the set is as large as 1,073 claims).

4.3.2 10IPC – fallout value

The fallout value measures how many non-relevant claims that were retrieved amongst the top 100 claims. The average fallout values and the median values for the 10IPC are shown in figure 50.

Figure 50: fallout general table for 10IPC

Fallout per computations							
10IPC							
Indexing method	Normalization	Baseline		FDG		Volk&Andersson	
		Average	Median	Average	Median	Average	Median
All-term	Cosine	0.87	0.92	0.84	0.87	0.85	0.89
	No normalization	0.88	0.91	0.87	0.9	0.87	0.9
	Simple normalization	0.87	0.91	0.85	0.88	0.85	0.89
Luhn	Cosine	0.9	0.93	0.9	0.93	0.91	0.94
	No normalization	0.92	0.95	0.92	0.96	0.93	0.97
	Simple normalization	0.9	0.93	0.9	0.94	0.92	0.95
Stoplist	Cosine	0.87	0.91	0.84	0.87	0.84	0.89
	No normalization	0.87	0.91	0.86	0.89	0.87	0.9
	Simple normalization	0.87	0.89	0.84	0.87	0.85	0.89

The figure shows that the average fallout values are very high, independently of type of modulation of the data, although the average of golden relevant claim per search topic is 229 claims.

Also for 10IPC, the computations using a decompounding model generate generally lower fallout values. The five search topics that hold the lowest fallout values have a higher number of different golden standard relevant claims than the cut-off value (only the top 100 claims were to be retrieved). The number of different golden relevant claims for this search topics range from 416 to 816. Subsequently, the cut-off value used in this study is actually too low considering the quantity of golden relevant claims there are for each search topic in 10IPC.

The computation Volk&Andersson.Stoplist.Cosine generated the overall lowest fallout value with 0.41 for search topic 431208. The search topic is assigned 5 codes from both section C and A. The codes in section C belongs to the subclass covering heterocyclic compounds and the codes in section A all belongs to the class covering hygiene within medical or veterinary science. For this specific search topic there are 537 different golden relevant claims. Hence, the fallout value could have been much lower, even zero, had the computation performed better.

4.3.3 10IPC – recall value

The recall presented in the figure 51 shows the average recall values for the different computations performed on the search topic set. These values are based on each search topics average recall value. The median and the average double quota are also presented in the table for each computation.

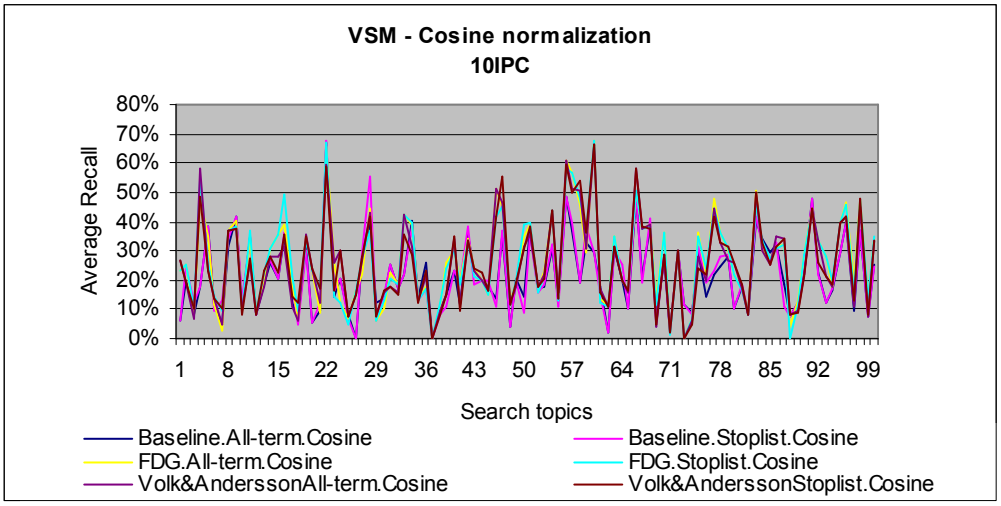
Figure 51: average recall general table for 10IPC

Average Recall per computation										
10IPC										
Indexing method	Normalization	Baseline			FDG			Volk&Andersson		
		Average	Median	Double	Average	Median	Double	Average	Median	Double
All-term	Cosine	21.50%	20%	15	26.80%	25%	18	25.70%	25%	16
	No normalization	18.70%	18%	13	19.30%	19%	13	19.50%	17%	13
	Simple normalization	21.90%	21%	14	24.50%	23%	16	24.10%	22%	16
Luhn	Cosine	17.40%	16%	13	18.60%	17%	13	16.10%	14%	11
	No normalization	12.60%	10%	9	12.80%	10%	9	9.70%	7%	7
	Simple normalization	16.80%	15%	13	17,00%	15%	12	13.50%	10%	10
Stoplist	Cosine	21.80%	20%	15	26.80%	25%	18	25.70%	24%	17
	No normalization	20.50%	19%	14	21.30%	20%	14	20.10%	18%	13
	Simple normalization	22.30%	21%	15	25.20%	24%	17	24.40%	23%	16

The results for the recall measurement show a relation between the performances of the different computations which corresponds to the result for the MAP measurement – i.e. the results indicate that a higher mean average recall value for the 10IPC set will be generated when a decompounding module is used. What separates the results for the 10IPC search topic set from the two other sets is that the Simple normalization seems to have a positive effect. For instance, the computation with the Baseline setting, both for indexing method All-term and Stoplist, obtain higher average recall value with Simple normalization.

In figure 52 all computations with Cosine normalization and either indexing method All-term or Stoplist, are presented for the three test settings.

Figure 52: chart of Arecall for 10IPC



The figure 52 shows that the average recall values for the search topics differ extensively from topic to topic. In figure 53 the search topics that receive the five highest average recall values and their average precision values are presented.

Figure 53: table of Arecall for 10IPC

The five highest average recall for 10IPC														
Search topic	All golden relevant claims	All different golden relevant claims	Baseline				FDG				Volk&Andersson			
			All-term		Stoplist		All-term		Stoplist		All-term		Stoplist	
			Arecall	AP	Arecall	AP	Arecall	AP	Arecall	AP	Arecall	AP	Arecall	AP
405247	163	129	18%	0.016	18%	0.015	48%	0.052	54%	0.050	58%	0.040	48%	0.033
413311	118	50	67%	0.441	67%	0.439	66%	0.362	67%	0.349	60%	0.200	59%	0.198
425414	174	87	29%	0.043	29%	0.043	68%	0.232	68%	0.231	66%	0.203	66%	0.199
424237	152	89	49%	0.242	48%	0.240	61%	0.261	58%	0.237	61%	0.263	59%	0.249
427590	91	61	49%	0.094	49%	0.100	51%	0.168	51%	0.167	58%	0.191	58%	0.188

When comparing the different computations for one specific search topic the average recall value tend to be higher for the computations using a decompounding module. For instance, the average recall values for search topic 425414 range from 29% to 68%. This specific search topic will get even higher values when using the Simple normalization (as high as 74% average recall with the computation Volk&Andersson.Stoplist.Simple).

As I have already mentioned, the double quota factor could well be the factor that changes the average recall value. The FDG setting computations tend to retrieve more relevant claims with more common codes than the computations with the Volk&Andersson setting. For instance, for the computation FDG.All-term.Cosine the average double quota factor is 18 and for the corresponding computations with the Volk&Andersson setting the average double quota factor is 16 (see figure 52). The loss in percentages correspond to a lower double quota factor, which indicates that the search topic that gets a higher average recall value only found more of the relevant claims that were already classified as relevant for one of the other codes for that search topic.

4.4 General analysis of the result

When comparing the three test settings (Baseline, FDG, Volk&Andersson) over the three search topic sets, we conclude that the FDG setting has the best mean average performance. However, only once it occurred that the FDG setting generated the best performance for a single search topic (in 5IPC for the measurement average precision).

It seems that very elaborate decompounding will give lower position in the retrieved list for the relevant claims – generally, the Volk&Andersson setting generates higher positions for the retrieved claims than does the FDG setting. On the other hand, a more restrict decompounding method will decrease the recall value – the FDG setting generally generates higher recall values than the Volk&Andersson setting.

The indexing method All-term and Stoplist generates significantly better result than the Luhn indexing method. As I have already mentioned, this could partly be explained by the distribution of the vocabulary used in patent claims, where the terms with low frequencies are important for the context and terms with medium frequency are more generally used claims.

At a first glance the results from the three search topic sets UniqueIPC, 5IPC and 10IPC tend to show the same results. That is a decompounding module will generate better values for MAP, fallout, and recall. However, there is a difference among the overall best results for the three search topic sets. For instance, overall best result for the MAP value for UniqueIPC is 0.0583, for 5IPC the MAP value have increased with 14 percent to 0.0687 and for 10IPC 26 percent to 0.0768. All these MAP values are obtained by the same computation FDG.All-term.Cosine.

The increase in MAP values for both 5IPC and 10IPC could well be explained by the corresponding increase of relevant claims. But this is not entirely true since the highest average precision value is obtained by a search topic belonging to the UniqueIPC with values as high as 0.781 for computation Volk&Andersson.All-term.Cosine and for computation FDG.All-term.Cosine the corresponding value is 0.752. The best average precision value for one search topic in 5IPC is 0.580 obtain by computation FDG.All-term.Cosine and for 10IPC the best average precision value is 0.436 obtain by Baseline.Stoplist.Cosine. The overall values increases in the order from UniqueIPC, 5IPC to the 10IPC and the reverse order is true for the single best values for one specific search topic. This contradiction is also observed for the other two measurements recall and fallout.

By studying the results, I have identified three main factors that could affect a search topic's ability to retrieve relevant claims – *document* (or *different terms*) *length, number of golden standard relevant claims* and *the IPC section classification*. One of the factors alone will not explain why certain search topics generated good average precision values.

4.4.1 Length

In figure 54 the average precision values for the search topics in UniqueIPC is contrasted with the average number of different terms of the retrieved and relevant claims for each search topic. Only the computation with indexing method All-term and Cosine normalization is presented for all three settings.

Figure 54: chart of length versus average precision for UniqueIPC

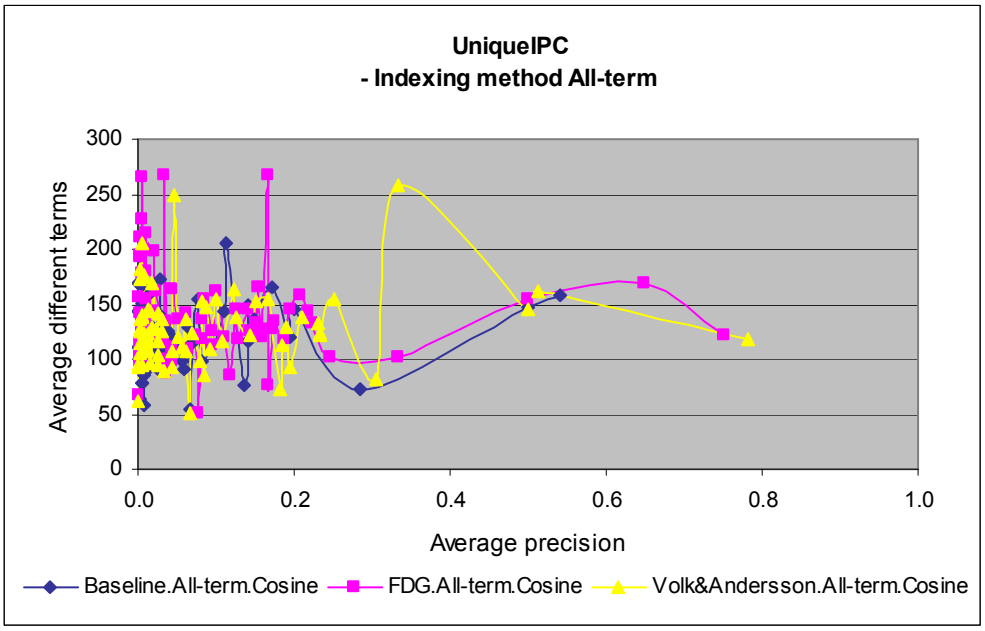


Figure 54 shows that there are five search topics retrieving a significantly higher average precision values than the other search topics. These five search topics contain the same amount of different terms as other search topics with considerable lower average precision values. However, when analyzing these five search topic there are actually four that generate good average precision for all three settings. The figure 55 shows the five search topics that generate the highest values for each setting's computation.

Figure 55: table of search topic and average precision values for UniqueIPC

The five search topics in UniqueIPC with the highest average precisions values					
Baseline.All-term.Cosine		FDG.All-term.Cosine		Volk&Andersson.All-term.Cosine	
Search Topic	Ap	Search Topic	Ap	Search Topic	Ap
433688	0.542	**436822**	0.752	**436822**	0.781
411110	0.500	**433688**	0.648	**433688**	0.513
423288	0.286	**411110**	0.500	**411110**	0.500
426334	0.200	**423288**	0.333	433512	0.333
436822	0.196	410847	0.246	**423288**	0.304

These search topics get higher values for the decompounding settings than for the Baseline setting. Also in 5IPC there is the same number of search topics doing well for all three settings. However, for 10IPC there are only three search topics that generate good average precision for all three settings' computations.

The FDG decompounding module increases both the document length and the number of different words more than the Volk&Andersson decompounding module does. The decrease in number of different terms for the Volk&Andersson setting could be the work of the module which removes the interfix morpheme and reconstructs the underlying lemma (i.e. the find-lemma module) used in the Volk&Andersson setting.

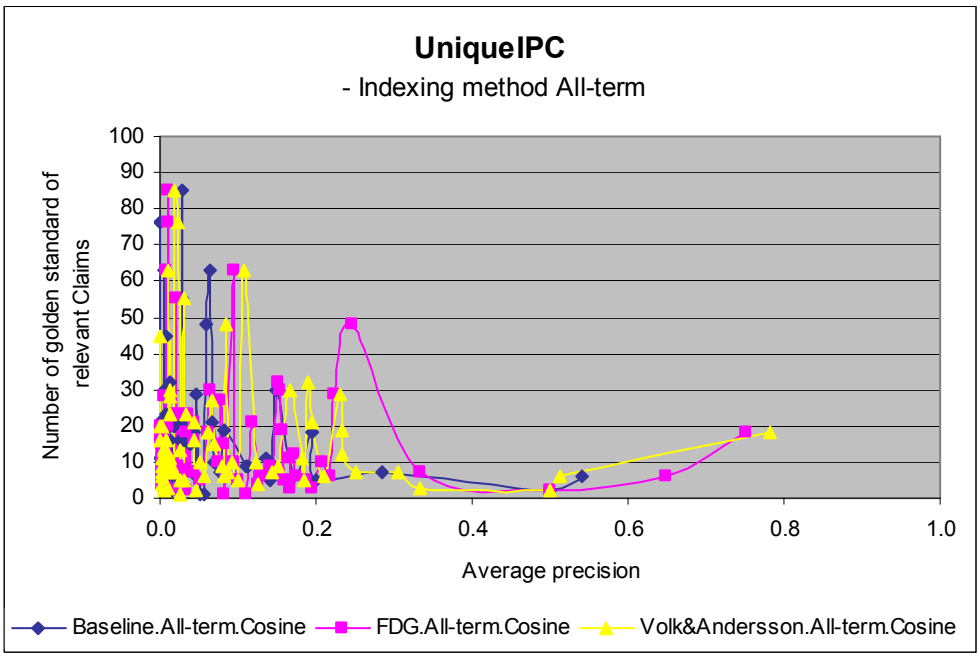
We can conclude that decompounding generates more common words, and at the same time extracts content bearing words for the bag-of-words method used in this study. As a result the overall best result is performed by a computation within the setting that allows most decompounding and thereby increases both the document length and the different term length – the FDG setting.

However, the length factor alone does not explain why certain search topics generate good average precision values, since there are many search topics with the same number of different terms or document length which perform poorer than the best performing search topics. Moreover, when analysing the best performing search topics for each search topic set, the same search topics perform well in all three test settings with some exceptions. In fact, this is also true when the indexing method Stoplist is used instead of All-term.

4.4.2 Number of golden standard relevant claims

In UniqueIPC the best performing search topics (i.e. search topics generating an average precision value equal to or greater than 0.5) have a set of golden standard relevant claims ranging from 1 to 18, as shown in figure 56.

Figure 56: chart of golden standard versus average precision for UniqueIPC



The figure shows that, for all three settings, there are eight search topics (actually only three different search topics) that generate average precision values equal to or greater than 0.5. When analysing the results for the other two search topic sets, 5IPC and 10IPC, the best performing search topics, in all three settings, tend to be the same search topics. The set of different golden standard relevant claims ranging from 38 to 180 claims for 5IPC and for 10IPC from 50 to 816 claims.

To conclude, the golden standard relevant claims factor alone does not explain why certain search topics generate good average precision values, since there are many search topics with

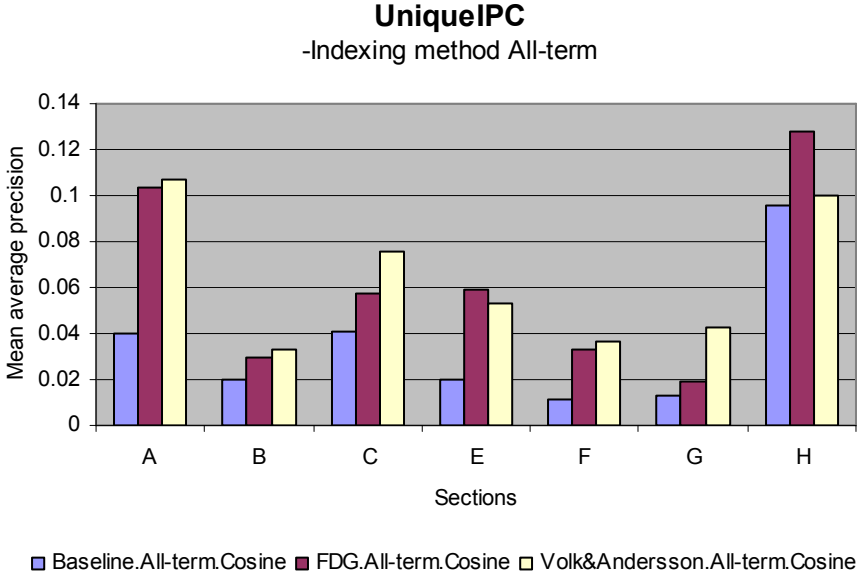
the same number of golden standard relevant claims that perform poorer than the best performing search topics. Moreover, when analysing the best performing search topics for each search topic set, the same search topics perform well in all three settings with some exceptions. In fact, this is also true when the indexing method Stoplist is used instead of All-term.

4.4.3 IPC section classification

The patent genre consists of several sublanguages (see section 2.4.1) and is extremely heterogeneous. Since patents describe different type of inventions the terminology, vocabulary, syntax will alter between patents. Different sections display different tendencies to use one or more sublanguages (as class C07, see section 2.4.5), and the ability of a search topic to retrieve relevant claims depends on the presence of sublanguages and the characteristics of the sublanguage or the sublanguages used in the patent scope.

Figure 57 shows the MAP values for the search topics in UniqueIPC in relation to the section which the search topics are classified by.

Figure 57: chart of mean average precision per section for UniqueIPC



Section
A (*HUMAN NECESSITIES*)
B (*PERFORMING OPERATIONS; TRANSPORTING*)
C (*CHEMISTRY; METALLURGY*)
E (*FIXED CONSTRUCTIONS*)
F (*MECHANICAL ENGINEERING; LIGHTING; HEATING; WEAPONS; BLASTING*)
G (*PHYSICS*)
H (*ELECTRICITY*)

The figure reveals the effect of decompounding, where the computations with the two decompounding settings generates generally higher values. The largest difference, between

the decompounding settings (FDG setting and Volk&Andersson setting) and the test setting without decompounding (Baseline setting), is shown in section A.

Between the two decompounding settings the Volk&Andersson setting's computation performed better in four section (A, B, C, G) and for the other two sections (H, E) the computation with FDG setting performed better. Even though corresponding computations with FDG setting generated an overall higher MAP values for the entire search topic set (see figure 27), the above figure shows that a more restrictive decompounding will generate better overall all sections.

The best MAP value for a section (H) is hold by computation FDG.All-term.Cosine. One would expect that this section (electricity) contains patents with a highly standardised sublanguage (such as measurement, electricity terms etc.) and a strong affiliation with legislative terminology. This means that the authors of the claims tend to use the same words for a specific concept. However this concept could or are hidden within compound, since Swedish is a compounding language, therefore by exhaustive decompounding more important features are exposed to the retrieval model.

On the other hand, also section C has a standardised sublanguage and as figure 57 shows the exhaustive decompounding method could be harmful towards the retrieval model. The result indicates that chemical compounds which are found in section C should be exposed for a more restrictive decompounding method for search topics in UniqueIPC. The tendency is the same for search topic in 5IPC classified in section C. However, the search topic in 10IPC classified in section C differ, since the results indicate that the best result will be obtained without decompounding and the second best result is generated when the exhaustive decompounding method is used.

The difference between using a more exhaustive decompounding method towards a more restrictive could also differ from search topic to search topic classified in the same section. For example, in section G (physics) there is one search topic, in UniqueIPC, that obtains twice as high average precision with the restrictive decompounding setting (Volk&Andersson) compared to the second best which is the Baseline setting. The FDG setting value is significant lower compared to the other two settings.

For search topics assigned more than one IPC code could be affected by the phenomenon "homogeneity". The homogeneity defined to which degree the IPC codes of a search topic belongs to the same IPC section, class, subclass and main group. The homogeneity parameter affects the performance of search topics within 5IPC and 10IPC. The generally tendency for both search topic sets are that: a higher average precision value will be obtain if all codes are classified in the same section for a search topic. For instance, the highest average precision value for one search topic, in 10IPC, is generated by a search topic with all its IPC codes belonging to section C.

Also the double quota factor seems to have an impact on the performance of the search topics, within 5IPC and 10IPC. The best performing search topics, both in 5IPC and 10IPC, have an average double quota factor of 3 (i.e. three claims are relevant for more than one of the IPC code for a search topic). To summarize, the IPC section classification factor reveals the complexity of the patent genre and the genre diversity in vocabulary among its sub domains.

5. Discussion and future research

It is necessary to investigate the patent genre from different aspects to get better results. Here we could make use of the work within NTCIR and the work within other on going projects.

Furthermore, it would be interesting to investigate if patent text style and structure differ more between one language and another than patent text generally differs from other genres. The finding of the cross-genre studies within NTCIR-7 evidently show that by selecting terms from both the research paper and patents generates better performance. In the section *Patent Issues* (5.3) I will more in detail reflect on earlier work done within the patent domain and the finding of my study.

Knowing more about how patent text differs between different languages could help in the IR-process when building a language independent retrieval system and automatic classifier. I strongly suggest that some adjustments regarding both specific language features and the text domain have to be done before indexing. For instance, we could learn from the work done by Gawronska and Erlendsson (see secton 3.3) in the bioinformatics field regarding chemical compounds and DNA-terms.

I fully agree with Fall *et al*. (see section 2.4.5) that the pre-processing should take up lesser time than it actually does. I have spent too much time coping with the pre-processing (i.e. pre-indexing) issues to check that the performance is reasonably good. Yet, my efforts have only made the raw data slightly more adaptable to the linguistic tools and the IR modules. In section *Morphological Analysis* (5.1) I will discuss the linguistic tools used in this study more exhaustively.

The absence of good discriminators and the terms sparseness in the documents explains the overall poor result in the study. Since the term weighting method and the retrieval model used make use of good discriminators to distinguish documents from each other, the method and model will not generate a good result when discriminators are absent. There are three main explanations to why the discriminators are absent:

- The OCR-process did not correctly identify words and therefore some of the significant words were lost – both increasing the terms sparseness and generating false similarity. For instance, the chemical compounds were not transformed at all or only to a small part. In section *OCR issues* (5.2) I will discuss issues connected to the OCR-errors.
- The text in the claims is only general, since the significance of the claim is described with an image – the important word being absent entirely. For instance, in patents consisting of mechanics or electronics inventions, images constitute almost all significance of the patent.
- Other studies in this field have indicated that claims are not good enough for either search topic or target set because of the vocabulary, style and shortness – claims being generally lexically sparse.

More studies have to be done on how to use the claim in the patent domain. My study indicates that claims do not contain enough material to use either as a search topic or as target set. The sparseness of terms in this type of collection is significant. The three different indexing methods used in this study (All-term) performs best for each search topic set and

measurement. Furthermore, when reducing the terms in each claim with the Luhn indexing method, i.e. only using those terms with a medium frequency, the performance decreases significantly for each computation. These findings, which correspond to earlier research done by Mase*, et al.* (see section 2.4.3), indicate that:

- low frequent terms are significant for a patent content,
- a general sparseness of important key terms in the claim sections.

The study done in this thesis shows that using decompounding modules for Swedish patent claims will increase the performance of the Vector Space Model (as shown in tables and charts section 4) –decompounding reduces the terms sparseness. To select compound and decompound terms from the claim section could well be something to implement in a present patent retrieval system for Swedish patent text. Also to use a query expansions technique could increase the performance of a patent retrieval system.

As the results show there are some good cases and the number of good cases increase when using a decompounding module. The next step is to find more aspects common to the good cases. For instance, does the vocabulary differ from the vocabulary in those claims that give a very poor result? It would also be interesting to use the claim similarity values as an additional ranking booster for the entire patent document. If the claims of the patent documents are very similar – is it or is it not a good indicator that the corresponding patents are similar?

5.1 Morphological analysis

My study shows that a decompounding module will increase both recall and average precision for Patent Retrieval systems for Swedish text. The decompounding module must be fully automatic – it is not feasible to handle the decompounding manually. In this study two different decompounding modules have been evaluated – the decompounding module of the FDG setting and the Volk algorithm in combination with the heuristic algorithm of the Volk&Andersson setting. Unfortunately, I have not been able to find information about how the FDG decompounding module actually works for Swedish text.

The Volk algorithm has a good resolving power i.e. manage to disambiguate 90,000 out of 100,000 (see section 3.5). But the heuristic algorithm in the study is only an embryo and it has not yet been fully evaluated and tested. A full scale testing of its resolving power would require a manually annotated collection with correct decompound segments assigned to each compound. Moreover, the collection should also contain information about if a compound should not be decompounded because of lexicalization or frequent use. However, this is also task depending – for machine translation one might want a more extensive decompounding, whereas in IR one might prefer the decompounding only of the context produced compounds, in order to increase precision. On the other hand, in Patent Retrieval it is very important to have a high performance in recall as van Dulkan proclaims, since it is important that no one else have already patented the invention (see section 2.1 and 2.4).

Two questions still remain unsolved:

- How to disambiguate a compound (how to choose the correct decompounding) taking the context into account?
- How to handle the lemmatization of the decompounded parts?

In my study the phonological rules on how Swedish compounds are allowed to be compounded were used backwards in order to find the closest lemma, but just like the heuristic decompounding algorithm this is only an embryo, which needs further testing.

The method used in Sjögbergh and Kann experiments, where the decompounding method is to use a pre-generated lists, and Dalianis suggestion, to split a compound according to four main rules (see section 2.5.1), would be interesting to try on patent documents. Also Karlgrens finding that the left element of the compound could well be useable in a query expansion in the patent domain as a more restrictive decompounding method.

5.2 OCR issues

My study distinctly shows that the OCR errors in the material will raise problems for the parser and for the retrieval model in the long run as earlier research have shown (see section 2.1.3). The OCR errors' effect on the performance of the parser was established on a small scale when 100 claims, selected on an arbitrary basis, were gone through.

The WIPO-alpha article authors mention that their patent documents contain few errors in the final collection (see section 2.4.5). This is not the case in the Swedish claim collection, and the most time consuming task in the pre-process has been the handling of problems originated in the OCR errors.

It is not possible to go through all material and manually correct all the OCR errors. Some general post process algorithms have to be developed and tested. My document collection is actually very well suited to this task since all or most of the documents are tagged with text font, which is important in OCR identification. Nylander's proof reading tool (see section 2.1.3) could be a good starting point for developing a more genre specific post processing correction. It is important to find a way to correct OCR errors without human intervention, and still getting good results and as few over-corrections as possible.

There is also a need to develop a parser which would accept a certain amount of contaminated data and still generate good results. In order to carry out a supervised comparison between different parsers, decompounding modules, tokenizers and other linguistic tools, a platform is needed. SVENSKA, mention in section 3.3, might serve as such a platform.

5.3 Patent retrieval issues

I would argue that there is a need for adapting parsers and other linguistic tools to the patent domain. For instance, linguistic tools have difficulties dealing with the analysis of tokens where the distinction lower case/upper case is fundamental (e.g. a chemical unit such as FeNo, or the term 'OCH-krets' quoted in my study, where the part preceding the hyphen will coincide with the Swedish conjunction 'och' ('and')). Other specific difficulties with patent texts are the abundance of long sentences with many commas and sentences which contains enumeration.

The linguistic normalization process should take care of stylistic phenomena such as bullet lists, massive use of punctuation, underlined text and also words written with space between the letters, as 'k ä n n e t e c k n a' ('c h a r a c t e r i z e'), which is very frequent in my collection – 'kännteckna' is actually the 11th most frequent lemma in the collection. A collection-based stop list would clearly remove this lemma, but in a patent context the word actually indicates that the next sentences or words will be the essence of the invention, and obviously 'kännteckna' should not be removed if one wants to explore significant indicators in the term weighting process.

At the NTCIR-6 workshop, Mase and Iwayama (see sections 2.4.3–2.4.4) concluded that claims are not sufficient for collecting patents with high similarity. My results show that this is also true for Swedish patent claims. Where my result at best obtains a MAP value of 0.07 and in NTCIR-3 to NTCIR-7 the best MAP values are over 0.2-0.4 for an entire search topic set. However, my study differs both in language and in task from NTCIR but what is more important is that the terms selected to be indexed are from different part of patent document in NTCIR experiments while in my study only the terms in the claim section are used. The results in my study evidently show that the claim section in a Swedish patent document does not contain enough content bearing words, although decompounding improves the retrieval process. My result confirms what earlier work done by Fall *et al*.(see section 2.4.5) and finding within the NTCIR-workshops (see section 2.4.4) already have established.

Furthermore, Mase and Iwayama concluded that both text analysis and retrieval algorithms should be modified to the technical field and the intention of the search topic, and this I also agree with. Some great researchers have with their work been able to capture the nature of written language (see section 2.1). A central assumption is that the authors tend to repeat important words which contribute to the essence of the topic of a text. However, this assumption and other assumptions of the nature of written language may not be applicable to the language used in patent claims, since the writers of patents tend to use different words or paraphrases to describe or point to the same essence of an idea.

Both my study and earlier work within the Patent Retrieval area indicate that it is difficult to capture the essence of an invention or an idea by using the words within the same patent. As Nanba, *et al*. exemplifies the hyperonym for the scholarly term "machine translation" is "natural language processing" but in patents the terms used are "automatic translation" or "language translation" (see section 2.4.4). Also the claim used as examples presented in section 3.1.1 show how the vocabulary can differ between two claims sharing the same IPC code. To develop efficient NLP-tools and retrieval systems for the patent domain we need to get a better understanding of the nature of patent language and its sublanguages. Moreover, as Sheremetyeva *et al*. (see section 2.4.1) point out, the language used in a patent is typically both a sublanguage of the domain of the invention and a legislative sublanguage.

The fact that even an average of 300 words, both as search topic and target collection, will not be able to capture the essence of a claim so that similar claims will be retrieved and have a high ranked position in the retrieved list, tells us something about the nature of the patent language – the essence of an idea (or a invention) is not entirely captured by the words used to describe the idea. The essence of the idea lies rather in how the words are combined together with a very few common key words such as words for a part of a chemical compound unit,

words for measurements, mechanical structure etc. Also, new words are highly frequent in the patent domain, as Fall et al (see section 2.4.5) addresses in the CLAIM-project.

Different languages have different ways to generate new words or to describe the same concept but with other words. In Swedish, the use of compounding for this purpose is very frequent, probably more frequent in technical texts, and as I show in my study, productive compounding is very frequent in patent texts. But the most interesting observation, which is in line with Karlgrens findings (see section 2.5), is that by reducing the compound into its parts the essence of the idea will also be revealed or captured by its parts, even if the entire word is never seen before. My study shows that using decompounding modules for Swedish patent claims will increase the performance of the Vector Space Model.

Even if the overall result is not as good as I would have whished, there are some search topics doing very well, and why is that? As shown in the result section 4.4 there could be several factors that influence a search topics ability to retrieve relevant claims, and it is likely that it is a combination of these that will influence the performance.

The data in my study are not statistically significant. The results only show indications of the performance of the retrieval model, the indexing methods and the normalizations methods used in the study. There are also several error sources in the process which could have contributed to the results:

- OCR errors
- The limitation of the NLP-tools for this special type of material
- Converting the material from EBCIDIC to ASSCII
- Program errors from my part
- Errors in the IPC assignment.

When it comes to the evaluation, some adjustments of the traditional measurements will perhaps be necessary, if we would like to use as assessors the IPC classification codes which are already assigned to patents. To use a certain patent classification system codes as assessors is not an entirely easy task, since multi-classification is the standard, and many similar patents have more than one code in common (see section 2.4.2). It would have been interesting to use ECLA codes as alternative assessors, in order to be able to compare the two assessors. However, this was not possible since not all the claims in the material had been assigned an ECLA code. In the future, a compromise could be to use IPC, F-term and ECLA as a complement to human assessors.

The struggle for finding the best way to detect similarity between patents and how to evaluate the performance is still in progress. However, since it could take up to two years to complete a Swedish application, an assisting tool which has a high performance of correctly identify similar patents would be very useful. My study is an embryo to deal with the similarity task and to evaluate the performance of the strategy chosen.

Finally, to do research within the Patent Retrieval area has revealed interesting problems and the work has deepened my experience as a programmer and as a linguist. I hope that my thesis will give inspiration to others to explore the patent genre for Swedish text.

6. Acknowledgements

During the work with my thesis there have been many people helping me and giving me support – my family, especially my father, and friends. Thank you for all the support and help.

I would also like to thank the Swedish Patent and Registration Office for the supply of the 30,217 claims used in this study, people at InfoData and Volvo Information Technology for extracting the material from the special cartridge format, people at Lingsoft for providing extra decompounding analysis for words, and people at the Swedish Institute of Computer Science for invaluable support.

A special thank you goes to Seija Hiltunen (the coordinator for students with disabilities at Stockholm University), and Jens Eeg-Olofsson. Another thank you goes to my cousin Lisbeth Andersson, working as a patent engineer, for giving me the idea to this thesis.

7. References

Aasa, J. (2004) *Unsupervised Resolution of PP Attachment Ambiguities in Swedish* Stockholm University, Department of Linguistics; D-Level Thesis in Computational Linguistics.

Ahlgren, P. (2004) *The Effects of Indexing Strategy-Query Term Combination on Retrieval Effectiveness in a Swedish Full Text* Doctor of Philosophy Thesis, University of Gothenburg University, Department of Library and Information Studies, Borås, Sweden.

Ahmad, K. and Al-Thubaity, A.-M. (2003) Can text analysis tell us something about technology progress?, In *Proceedings of the ACL-2003 workshop on Patent corpus processing* - Volume 20,(pp. 45-55), (Sapporo, Japan.)

Andersson, L. (2003) *Hantering av sammansatta ord vid indexering med två statistiska indexeringsmetoder* Stockholm University, Department of Linguistics; Bachelors Thesis in Computational linguistic.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999) *Modern Information Retrieval* New York, Addison-Wesley.

Bauer, L. (1994) *Introducing linguistic morphology* Bristol, Edinburgh Press.

Beitzel, S. M., Jensen, E. C. and Grossman, D. A. (2003) A Survey of Retrieval Strategies for OCR Text Collection, In *Proceedings 2003 Symposium on Document Image Understanding Technologies*, (Greenbelt, Maryland)

Bilting, U. and Skansholm, J. (1987) *Vägen till C* Lund, Studentlitteratur

Bruun, A. (1999) Development of the IPC as a search tool, *World Patent Information* - Volume 21(2), 97-100.

Buckley, C. and Voorhees, M. E. (2000) Evaluating evaluation measure stability, In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, (pp. 33-40), (Athens, Greece).

Chen, L., Tokuda, N. and Adachi, H. (2003) A patent document retrieval system addressing both semantic and syntactic properties, In *Proceedings of the ACL-2003 workshop on Patent corpus processing* - Volume 20, (pp. 1-6), (Sapporo, Japan)

Christiansen, T. and Torkington, N. (1998) *Perl cookbook Sebastopol* O'Reilly & Associates

Dalianis, H. (2005) Improving search engine retrieval using a compound splitter for Swedish, In *the 15th Nordic Conference on Computational linguistics,* (Joensuu, Findland)

Dodd, S., Library terms in English and Swedish, http://www.ub.uu.se/bibliotekstermer/?Swe=ä, [visited 2008-12-01]

Dura, E. (1998) *Parsing Words* Doctor of Philosophy Thesis, University of Gothenburg, Department of Swedish, Section of Computational Linguistics, Göteborg, Sweden.

Ejerhed, E., Källgren, G., Wennstedt, O. and Åström, M. (1992) *The linguistic annotation system of the Stockholm-Umeå corpus project* Stockholm University, Department of General Linguistics.

Ekeklint, S. (2001) *Tagga samman - ett verktyg gör semantisk analys av svenska sammansättningar* University of Gothenburg, Department of Swedish, Sections of Natural Language Processing and Lexicology; D-Level Thesis.

Ericksson, M. and Gambäck, B. (1997) SVENSK: A toolbox of Swedish language processing resources. In *Proceedings of the 2nd International Conference on Recent Advances in Natural Language Processing,* (Tzigov Chark, Bulgaria.)

Fall, C., Törcsvári, A., Benzineb, K. and Karetka, G. (2003), Automated categorization in the international patent classification, *ACM SIGIR Forum Publisher ACM Press* - Volume 37(1), 10-25.

Fall, C. J., Törcsvári, A., Fiévet, P. and Karetka, G. (2004) Automated categorization of German-language patent documents, *Expert Systems with Applications* - Volume 26, 269-277.

Fujii, A., (2007) Integrating Content and Citation Information for the NTCIR-6 Patent Retrieval Task, In *Proceedings of the 6th NTCIR Workshop Meeting*, (Tokyo, Japan.)

Fujii, A., Iwayama, M. and Kando, N., (2004), Overview of Patent Retrieval Task at NTCIR-4, In *Proceedings of 4th NTCIR Workshop Meeting*, (pp225-232), (Tokyo, Japan.)

Fujii, A., Iwayama, M. and Kando, N. (2005) Overview of Patent Retrieval Task at NTCIR-5 In *Proceedings of NTCIR-5th Workshop Meeting* (Tokyo, Japan.)

Fujii, A., Iwayama, M. and Kando, N. (2007) Introduction to the special issue on patent processing *Information Processing and Management* - Volume 43, 1149-1153.

Gawronska, B. and Erlendsson, B. (2005) Syntactic, Semantic and Referential patterns in Biomedical Text: towards in-depth text comprehension for the purpose of bioinformatics, In *Proceedings of the 2nd International Workshop on Natural Language Understanding and Cognitive Science NLUCS 2005* (Miami, USA)

Hansen, P. and Järvelin, K. (2000) The Information Seeking and Retrieval process at the Swedish Patent- and Registration Office Moving from Lab-based to real life work-task environment, In *Proceedings of the SIGIR 2000 workshop on patent retrieval*, (pp. 43-53), (Athens, Greece.)

Hansen, P. and Järvelin, K. (2005) Collaborative information retrieval in an information-intensive domain, *Information Processing and Management: an International Journal* - Volume 41(5), 1101-1119

Hedlund, T. A., Pirkola, A. and Järvelin, K. (2001), Aspects of Swedish morphology and semantics from the perspective of mono- and cross-language information retrieval *Information Processing and Management* - Volume 37(1), 147-161

IRF, Information Retrieval Facility Symposium 2007 and 2008, http://www.ir-facility.org/symposium [re-visited 2008-12-01],

Iwayama, M., Fujii, A., Kando, N. and Marukawa, Y. (2003) An Empirical Study on Retrieval Models for Different Document Genres: Patents and Newspaper Articles, In *Proceedings of the 26th annual international ACM SIGIR Conference on Research and Development in information Retrieval*, (pp. 251-258), (Toronto, Canada.)

Iwayama, M., Fujii, A., Kando, N. and Takano, A. (2003a) Overview of Patent Retrieval Task at NTCIR-3, In *Proceeding of the ACL-03 Workshop on Patent Corpus Processing*, (pp. 24-32), (Sapporo, Japan.)

Johnson, M., Handouts for class 2002, CG41 Morphology, the structure of words, http://www.cog.brown.edu/~mj/classes/cg41/handouts/wk02a.pdf, [visited 2003-10-15]

Jurafsky, D. and Martin, J. H. (2000) *Speech and Language Processing* New Jersey, Prentice-Hall Inc

Järborg, J. (1998) *Sammansättningssemantik*, Department of Swedish, Göteborg University,

Kando, N. (2000) What Shall We Evaluate? Preliminary Discussion for the NTCIR Patent IR Challenge (PIC) Based on the Brainstorming with the Specialized Intermediaries in Patent Searching and Patent Attorneys, In *Proceedings of the ACM SIGIR 2000 Workshop on Patent Retrieval in conjunction with the 23rd Annual International ACM SIGIR Conference on Research and Development in Information retrieval*, (Athens, Greece.)

Karlgren, J, Sahlgren, M. and Cöster, R.. (2004), Selective compound splitting of Swedish queries for Boolean combinations of truncation terms, In *the 4th Workshop of Cross language Evaluation Forum CLEF 2003*, (Trondheim, Norway.)

Karlgren, J. (2005) Occurrence of compound terms and their constituent elements in Swedish, In *the 15th Nordic Conference on Computational linguistics,* (Joensuu, Findland)

Karlsson, F. (1992), SWETWOL: A Comprehensive Morphological Analyser for Swedish, *Nordic Journal of Linguistic* - Volume 15, 1-45

Koskenniemi, K. (1983) *TWO-LEVEL MORPHOLOGY: A General Computation Model for Word-Form Recognition and Production*, Doctor of Philosophy Thesis, University of Helsinki, Department of General Linguistics, Helsinki, Finland

Krier, M. and Zaccà, F. (2002), Automatic categorisation applications at the European patent office, *World Patent Information* - Volume 24(3), 187-196

Lancaster, W. F. and Warner, A. J. (1993) *Information Retrieval Today Arlington*, VA, Information Resources Press

Larkey, L. S., (1999), A patent search and classification system, In *Proceedings of the 4th ACM conference on Digital libraries*, (pp 179-187), (Berkeley, California, United States.)

Lee, D. L., Chuang, H. and Seamons, K. (1997), Document ranking and the vector-space model, *IEEE Software [publisher IEEE Computer Society Press]* - Volume 14(2), 67-75

Lyon, M. (1999), Language related problems in the IPC and search systems using natural language, *World Patent Information* - Volume 21(2), 89-95

Malmgren, S.-G. (1994) *Svensk lexikologi – ord, ordbildning, ordböcker och orddatabaser* Lund, Studentlitteratur

Manning, C. D., Raghavan, P. and Schütze, H. (2008) *Introduction to Information Retrieval* Cambridge, MA, Cambridge University Press. (Online version) nlp.stanford.edu/IR-book/pdf/irbookprint.pdf [re-visited 2008-12-01]

Manning, C. D. and Schütze, H. (2002) *Foundations of statistical natural language processing* Cambridge, MA, Massachusetts Institute of Technology (MIT)

Mase, H. and Iwayama, M., (2007), NTCIR-6 Patent Retrieval Experiments at Hitachi, In *Proceedings of NTCIR-6 Workshop*, (Tokyo. Japan.)

Mase, H., Tadataka, M., Yuichi, O., Iwayama, M. and Tadaaki, O. (2005), Proposal of Two-Stage Patent Retrieval Method Considering the Claim Structure, *ACM Transactions on Asian Language Information Processing* - Volume 4 (2), 186-202

Mittendorf, E. and Schäuble, P., (1996), Measuring the effects of data corruption on information retrieval, In *Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval* (SDAIR)

Moens, M. (2000) *Automatic indexing and abstracting of document texts* Boston, Kluwer Academic Publishers

Nakatani, Y., Takada, K., Isoda, M., Okumura, M., Iwayama, M., Marukawa, Y. and Shinmori, A., (2002), NTT DTEC at Patent Retrieval Task, In *Working notes of the 3rd NTCIR Workshop Meeting. PartIII Patent Retrieval Task*, (pp. 25-31), (Tokyo. Japan),

Nanba, H. (2007) Query Expansion using an Automatically Constructed Thesaurus In *Proceedings of the 6th NTCIR Workshop*, (Tokyo. Japan.)

Nanba, H. , Fujii A., Iwayama, M., and Hashimoto, T. (2008) Owerview of the Patent MIning Task at the NTCIR-7 Workshop, In *Proceeding of the 7th NTCIR workshop,* (Tokyo, Japan)

Nylander, S. (2000) *Statistics and Graphotactical Rules in Finding* Uppsala University, Department of Linguistics; Master thesis

PaIR'08, 1st International CIKM Workshop on Patent Information Retrieval, http://www.cikm2008.org/workshops.php and http://www.ir-facility.org/events/pair08 [re-visited 2008-12-01],

Svenska Författningssamlingen (1967) *Patentlagen* § 82, SFS, 1967:837

Riad, T. (1997) *Svensk fonologikompendium* Stockholm University, Department of Scandinavian Languages, (3rd edition)

Rundell, M. and Fox, G. ed. (2002) *English Dictionary - for advanced learners*, Bloomsbury Publishing, Macmillan Publishers Limited

Salton, G. (1987), Historical Note: The past thirty years in information retrieval *Journal of the American Society for Information Science* - Volume 38(5), 375-380

Salton, G. and McGill, M. J. (1983) *Introduction to modern information retrieval* New York, McGraw-Hill Inc.

Salton, G., Wong, A. and Yang, C. S. (1975) A vector space model for automatic indexing *Communications of the ACM* -Volume 18(11), 613-620

Saracevic, T., Spink, A. and Wu, M., (1997), Users and Intermediaries in Information Retrieval: What Are They Talking About? User Modeling, In *the 6th International Conference, UM'97*, (Sardinia)

Schmid, H., (1994), Probabilistic Part-of-Speech Tagging Using Decision Trees, In *the International Conference on New Methods in Language Processing*, (Manchester, United Kingdom.)

Schultz, C. K. ed. (1968) *H. P. Luhn: pioneer of information science*, selected works, New York, Spartan Books

Schäuble, P. (1997) *Multimedia information retrieval: content-based information retrieval from large text and audio databases* Boston, Mass, Kluwer

Sheremetyeva, S. (2003), Natural language analysis of patent claims, In *Proceeding of the ACL-03 Workshop on Patent Corpus Processing*, (pp. 66-73), (Sapporo, Japan.)

Sheremetyeva, S., Nirenburg, S. and Nirenburg, I., (1996), Generating Patent Claims from Interactive Input, In *Proceeding of the 8th International Workshop on Natural Language Generation*, (pp 61-70), (Herstmonceux, Sussex)

Shinmori, A., Okumura, M., Marukawa, Y. and Iwayama, M., (2003), Patent Claim Processing for Readability, In *Proceeding of the ACL-03 Workshop on Patent Corpus Processing*, (pp 56-65), (Sapporo, Japan)

Sjöbergh, J. and Kann, V. (2004) Finding the Correct Interpretation of Swedish Compound a Statistical Approach, In *Proceeding of the 4th International Conference of Language Resources and Evaluation LREC-2004*, (pp. 899-902), (Lisbon, Portugal)

Sparck, J. K., Walker, S. and Robertson, S. E. (2000), A probabilistic model of information retrieval: development and comparative experiments: Part 1, Part 2, *Information Processing and Management* -Volume 36(6), 779-808, 809-840

Svenska Akademien, *Svenska Akademiens ordbok*, http://g3.spraakdata.gu.se/saob/ (entries: STÄNDIG; STÅND), [re-visited 2007-11-01]

Tapanainen, P. (1999) *Parsing in two frameworks: finite-state and functional dependency grammar* Doctor of Philosophy Thesis, University of Helsinki, Helsinki, Finland

Tapanainen, P. and Järvinen (1997) *A non-projective dependency parser*, ANLP'97, University of Helsinki, Department of General Linguistics

Teleman, U., Hellberg, S., Andersson, E. and Christensen, L. (1999) *Svenska Akademiens grammatik* [The grammar of the Swedish Academy], (4 volumes), Stockholm, Svenska Akademien, NorstedsOrdbok,

van Dulken, S. (1999), Free patent databases on the Internet: a critical view, *World Patent Information* -Volume 21(4); 253-257

van Roy, P. and Haridi, S. (2004) *Concepts, Techniques, and Models of Computer Programming* MIT Press

Wanner, L., Baeza-Yates, R., Brügmann, S., Codina, J., Diallo, B., Escorsa, E., Giereth, M., Kompatsiaris, Y., Papadopoulos, S., Pianta, E., Gemma, P., Puhlmann, I., Gautam, R., Rotard, M., Schoester, P., Serafini, L. and Zervaki, V. (2008) Towards content-oriented patent document processing *World Patent Information* -Volume 30(1), 21-30

Widdows, D. (2004) *Geometry and Meaning* United States, CSLI Publication

Vinciarelli, A. (2005), Noisy Text Categorization, *IEEE Transactions on pattern analysis and machine intelligence* -Volume 27(12), 1882-1895

WIPO (2004) International Patent Classification -Guide, 7th edition, http://www.wipo.int/classification/fulltext/new_ipc/guideeng.htm [visited 2004-04-15]

WIPO (2004a) Introductory Manual to the International Patent Classification (IPC), 7th edition, http://www.wipo.int/classification/en/manual/manual2.htm [visited 2004-04-15]

WIPO, IPCCAT - International Patent Classification version 7 Categorization Assistant, http://www.wipo.int/ipccat/ipc.html (select: Heading General help, Topi: 12. Training sets and Limitations), [visited 2007-11-05]

Wolfram, D. and Zhang, J. (2008), The Influence of Indexing Practices and Weighting Algorithms on Document Spaces, *Journal of the American Society for Information Science and Technology* -Volume 59(1), 3-11

Volk, M., (1999), Choosing the right lemma when analysing German nouns In *Multilinguale Corpora: Codierung, Strukturierung, Analyse 11. Jahrestagung der GLDV*, (Frankfurt, Germany)

Wong, S. K. and Raghavan, V. V. (1984) Vector space model of information retrieval: a reevaluation, In *Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval*, (Cambridge, England)

Wong, S. K., Ziarko, W., and Wong, P. C. (1985) Generalized vector spaces model in information retrieval. In *Proceedings of the 8th Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (pp 18-25), (Montreal, Quebec, Canada)

Voutilainen, A. (2001), Parsing Swedish, In *Proceedings 13th Nordic Conference on Computational Linguistics (Nodalida-01)*, (Uppsala, Sweden.) stp.ling.uu.se/nodalida01/pdf/voutilainen.pdf [re-visited 2008-12-01],

Zipf, G. K. (1949) *Human Behavior and the Principle of Least Effort* London, Hafner

8. Appendices

Appendix 1: Documentation of Search topic 436822 in UniqueIPC

Documentation of Search topic: 436822 in UniqueIPC

Publication number: SE436822 Publication date: 1985-01-28
Inventor: TORNSTROM I Applicant: KOPPARFORS AB (SE)

IPC Code	Number of claims in IPC Code	Golden standard relevant claims
A01C11/02	18	427611, 434903, 430743, 413828, 423019, 421985, 425624, 425042, 422871, 408121, 423172, 430557 406025, 404646, 430558, 428748, 435886, 411413

Abstract of SE436822

A planting and ground treatment device for a planting machine is composed of a rotational planting part (1) which through an in all directions elastically yielding unit (10) is in connection to only one ground treatment device (11-13), which has an angularly adjusted cutting edge (13). The cutting edge is preferably inclined in a downward direction from the arrangement's centre and removes ground vegetation from the planting area. At the planting device a steering and forming surface (2) is attached to pack earth around the plant so that a mound effect is achieved. The arrangement even includes a locking device (19) that prevents the plant from being pushed up during the planting operation

Documentation of similarity calculation

The top five highest similarity values for the Search topic (the bold is the highest similarity value)

DN	Similarity value	Computation (test setting, indexing method, normalization factor)
421985	0.60682096	Volk&Andersson.Stoplist.Cosine
421985	0.595822149	Volk&Andersson.All-term.Cosine
408121	**0.65498803**	Volk&Andersson.Stoplist.Cosine
408121	0.651621052	Volk&Andersson.All-term.Cosine
411413	0.618017567	Volk&Andersson.Stoplist.Cosine
411413	0.613449502	Volk&Andersson.All-term.Cosine
411413	0.593879595	FDG.Stoplist.Cosine
411413	0.589529947	FDG.All-term.Cosine
428748	0.597699813	Volk&Andersson.Stoplist.Cosine
430743	0.589490453	Volk&Andersson.Stoplist.Cosine

Documentation of relevant and retrieved claim 408121

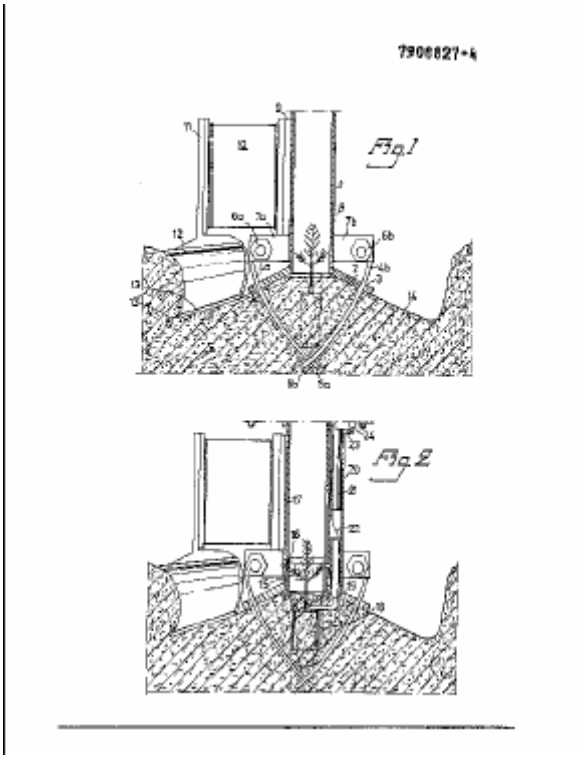
Publication number: SE408121
Publication date: 1979-05-21
Inventor: PETRE H Applicant: DOMENVERKET (SE)
IPC codes: A01C11/02

Entire text of retrieved and relevant claims: 408121

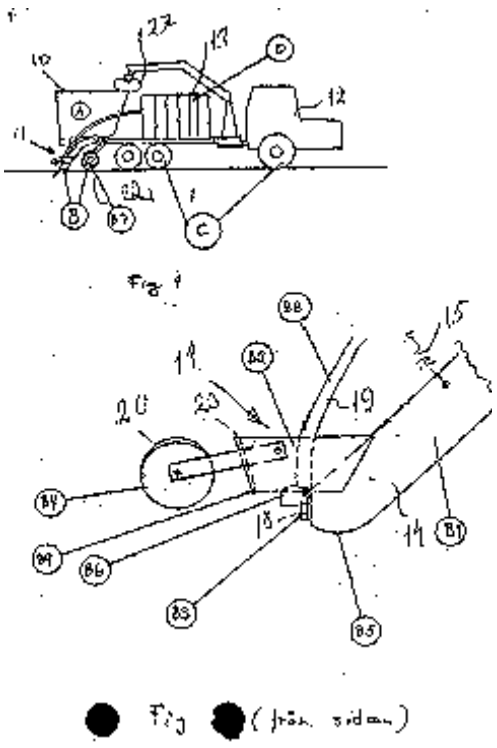
No English abstract was published for claim 408121. The translation of this claim was done by the author of this thesis. The purpose of the translation is to give the non-speaking Swedish reader an understanding of the claim used as an example.

1. A planting aggregate (or device) , especially the planting of forest plants, is characterized by at least one planting unit (11) which consist of a unit for supply of seedlings to a standby mode (18) for planting, and a unit (24,26) for the supply of a predetermined amount of soil past the standby mode, and also a direction of a suitable planting location guiding sleeve (21,16,17), that controls the soil supply, in such a way, that the soil will enclose the plant root ball, and the guiding sleeve consists of above the standby mode as an arranged soil split unit (21), which splits the deposition of soil into two falling parts, one of each side, of the plant, and two at each side of the standby mode located wings (16,17), which is arranged to steer the deposition of soil in the direction of the plant root ball. 2. The device (planting aggregate) under the patent claim 1, is characterized by the guiding sleeve (21, 16, 17), which is set up to control soil supply so that every opportunity for the deposition of soil completely or partially achieved will release the plant from the standby mode.3. The devices (planting aggregates) under the patent claim 1 or 2, are characterized by the planting unit (10), i.e. soil distribution unit (24, 26) and guiding sleeve (21, 16, 17), and they are combined as a unit in order to be a suitable base machine connect plug-in aggregate (or device), thereby basic machine is appropriate in adaptation of carrying a soil magazine (10) and a plant stack. 4. The devices (planting aggregates) in any of the claims 1-3, are characterized by that they are equipped with a number of planting units and related soil distribution units, guiding sleeves etc., thereby the peer distance between the planting device units will be easy to change.5. The devices (planting aggregates) in any of the claims 1-4, are characterized by each of the planting unit consist of a basic machine or a tripod preferable changeable pendulum arm (14), which lower end part may be arranged to be drag on the ground. 6. The devices (planting aggregates) in any of the claims 1-5, are characterized by, that the pendulum arm (14) is a spring release mechanism (15) directed towards a neutral position, the springs will preferable have progressively characteristic of an increased spring functionality. 7. The devices (planting aggregates) in any of the claims 1-6, are characterized by the advance forward motion direction of the planting units, which have adaptable units for withdrawal and push on functionalities (23, 20). 8. The devices (planting aggregates) in any of the claims 1-7, are characterized by, that the push on functional units consists of slanting rollers (20), which are located on both sides of the area, in which the planting unit moves during the forward motion phase. 9. The devices (planting aggregates) in any of the claims 1-8, are characterized by the advance forward motion direction, in front, of each planting unit, and these units have an appropriate type of temporarily excluded or avoidable obligation soil preparing unit (22). 10. The devices (planting aggregates) in any of the claims 1-9, are characterized by an interlocking units, which prevents soil supply or release of a plant in standby mode, if the planting unit is occupied in an inappropriate location.

Mosaic data belonging to the search topic 436822



Mosaic data belonging to the relevant claim 408121



Documentation of the evaluation process

For Computation: Volk&Andersson.All-term.Cosine

Unique set of relevant and retrieved:18
All in all relevant claims:18
Recall for class:A01C11/02
Recall value: 100%
Fallout cal: 0.82=(100-18-0)/(100-0)
Average similarity value for the retrieved and relevant claims is 0.399820589235131

Ranking list of retrieved and relevant claims

Retrieved and Relevant claims	Position
408121	1
411413	2
421985	3
428748	4
430743	5
435886	6
413828	9
422871	11
423019	12
430558	15
425624	16
430557	19
427611	21
423172	22
434903	23
406025	24
404646	27
425042	29

Interpolated average precision calculation

Precision at given document cut-off point every 10th until 100

Level	Calculation
10 retrieved	=7/10
20 retrieved	=12/20
30 retrieved	=18/30
40 retrieved	=18/40
50 retrieved	=18/50
60 retrieved	=18/60
70 retrieved	=18/70
80 retrieved	=18/80
90 retrieved	=18/90
100 retrieved	=18/100
SUM	3.87214285714286
Interpolated Ap	3.87214285714286/18

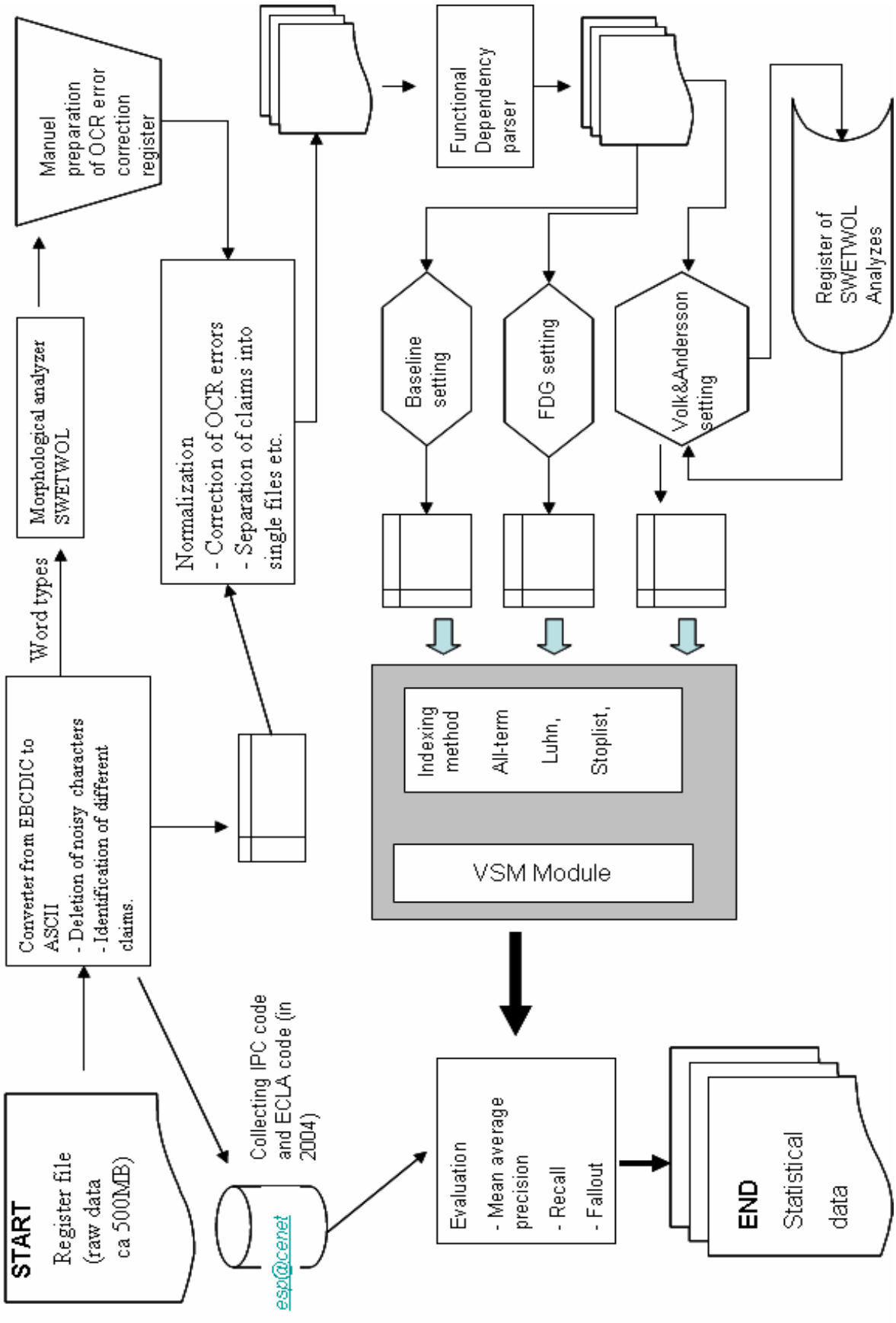
Interpolated average precision:0.215119047619048

Average precision calculation

1/1	6/6	11/16	16/24
2/2	7/9	12/19	17/27
3/3	8/11	13/21	18/29
4/4	9/12	14/22	SUM=14.065367239009
5/5	10/15	15/23	Ap=14.065367239009/18

Average precision value:
0.781409291056058

Appendix 2: Flow chart of the entire study



Appendix 3: A retrieval model used in NTCIR-3

$$(1 + \log(fq,t)) \times idft \times (1+\log(fd,t)/(1+\log(avefd)) \times (1/(avedlb+S\times(dlbd-avedlb)))$$

q query
d document
t term
N number of documents in the collection
idft inverse document frequency of term t
dlbx number of unique terms in x (query or document)
dlfx sum of term frequencies in x
avefx average of term frequencies in x
avedlb average of dlbx in the collection
S a constant 0.2

(Iwayama, 2003b, p. 255)

Appendix 4: Parsed sentence by the Functional Dependency Parser

1	Aggregat	aggregat		N NOM &NH
2	enligt	enligt	mod:>1	PREP &N<
3	patentkravet	patent#krav	pcomp:>2	N SG NOM &NH
4	1	1	mod:>3	<Card> NUM NOM &N<
5	,	,		
6	kännetecknat	kännetecknad		A SG NOM &NH AD SG &MV
7	därav	därav		ADV &AH
8	,	,		
9	att	att	pm:>13	CS &CS
10	styrorganen	styr#organ	subj:>13	N PL NOM &NH
11	(21,16,17	(21,16,17	mod:>10	<Card> NUM NOM &N<
12	)	)		
13	är	vara		V PRES &MV
14	inrättade	inrättad	sc:>13	A NOM &NH
15	att	att	v-ch:>16	INFMARK &AUX
16	styra	styra	mod:>14	V INF &MV
17	jordtillförseln	jord#tillförsel	obj:>16	N SG NOM &NH
18	så	så	advl:>16	ADV &AH
19	,	,		
20	att	att	pm:>30	CS &CS
21	den	den	subj:>30	PRON SG NOM &NH
22	för	för	advl:>30	PREP &AH
23	varje	varje	det:>24	DET SG NOM &>N
24	tillfälle	tillfälle	pcomp:>22	N NOM &>N &NH
25	nedfallande	nedfallande	attr:>26	NDE &>N
26	jordmängden	jord#mängd	mod:>24	N SG NOM &NH
27	helt	helt	advl:>30	ADV &AH
28	eller	eller	cc:>27	CC &CC
29	delvis	delvis	cc:>27	ADV &AH
30	åstadkommer	åstadkomma		V PRES &MV
31	plantans	planta	attr:>32	N SG GEN &>N
32	frigörande	fri#görande	obj:>30	NDE &NH
33	från	från	advl:>30	PREP &AH &N<
34	vänteläget	vänte#läge	pcomp:>33	N SG NOM &NH
35	.	.		Lim

Translation: The device (planting aggregate) under the patent claim 1, is characterized by the guiding sleeve (21, 16, 17), which is set up to control soil supply so that every opportunity for the deposition of soil completely or partially achieved will release the plant from the standby mode.

Appendix 5: Class distribution for each search topic set