FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Optimizing Text Classification for the Medical Domain: Identification of Papers on Off-Label Drug Use

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Johannes Dünser

Matrikelnummer 0404414

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dr. Andreas Rauber

Wien, 20.08.2012

(Unterschrift Verfasser) (Unterschrift Betreuung)

Erklärung zur Verfassung der Arbeit

Johannes Dünser
Mariahilfer Straße 170/16, 1150 Wien

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

Wien, 20.08.2012 _____

Johannes Dünser

Acknowledgements

First of all, I would like to extend my sincere gratitude to Bita Mesgarpour who provided the initial idea and the text collection on which this thesis is based. She also provided valuable feedback about the medical aspects of the work.

My gratitude also goes out to Prof. Andreas Rauber for letting me write this thesis under his supervision. His incredibly fast feedback and continued support led me to the right direction.

I also want to thank everybody else out there who inspired, supported, motivated or had any other positive influence on me during the creation of this thesis. You know who you are!

Abstract

Automated text classification is a well studied field and is successfully utilized for many different applications. General-purpose text classification systems can handle any kind of natural language text. However, with increasing specificity of the content the effectiveness of such a system deteriorates. A domain specific optimization is necessary to increase the performance further. This thesis is focused on optimizing a text classification system for the medical domain and, in particular, to detect articles originating from a biomedical literature database which discuss the topic 'off-label drug use'. The integration of the Unified Medical Language System (UMLS) as a rich source of biomedical background knowledge enables the application to reduce synonymous terms, resolve ambiguous concepts and expand the documents with hypernyms. Compared to the baseline classifier the improved system shows an increase in precision of 11.7% and an increase in recall of 2.5%. While these results are a significant improvement, there still is room for improvement. The high amount of concepts not suitable for document enrichment and the high interconnectedness in the ontology poses a serious problem for the expansion techniques. An analysis of the implemented stemming algorithm and stop word list suggests that a topic sensitive adaptation could prove beneficial. The thesis closes by outlining future work which will be necessary to solve the open issues and further improve the performance of biomedical text classification.

Kurzfassung

Systeme zur automatischen Textklassifikation sind weit verbreitet und werden für viele Applikationen erfolgreich eingesetzt. Besonders bei Texten mit sehr spezifischem Inhalt liefern diese Systeme jedoch oft nicht die optimale Leistung. Eine domänenspezifische Optimierung ist notwendig. Das Ziel der vorliegenden Arbeit ist die Optimierung eines Systems zur automatischen Textklassifikation am Beispiel des medizinischen Fachbereichs. Im Besonderen sollen Artikel welche sich mit der Thematik 'off-label drug use' (zu Deutsch 'zulassungsüberschreitende Anwendung') beschäftigen erkannt werden. Die Integration des Unified Medical Language System (UMLS) als medizinische Ontologie erlaubt die Reduktion von synonymen Begriffen, das Anreichern der Dokumente mit Oberbegriffen und das kontextsensitive Ersetzen von mehrdeutigen Konzepten. Die Auswertung des optimierten Systems im Vergleich zur Prototyp-Applikation zeigt eine Verbesserung der Genauigkeit um 11.7% und eine Verbesserung der Trefferquote um 2.5%. Obwohl die erzielten Resultate eine eindeutige Weiterentwicklung gegenüber dem Prototyp aufzeigen, besteht noch offenes Potential. Die inhaltliche Komplexität der verwendeten Ontologie verursacht Probleme bei der Anreicherung der Dokumente mit Oberbegriffen. Eine Analyse des verwendeten Stemming-Algorithmus und der Stopwort-Liste zeigt, dass eine domänenspezifische Anpassung sinnvoll wäre. Um eine weitere Verbesserung zu erreichen, müssen dieses und die anderen offenen Probleme im Rahmen von zukünftigen Forschungsarbeiten geklärt werden.

By striving to do the impossible, man has always achieved what is possible. Those who have cautiously done no more than they believed possible have never taken a single step forward. – Mikhail Alexandrovich Bakunin

Contents

CHAPTER 1

Introduction

1.1 Motivation and Background

The general motivation behind all information retrieval (IR) applications is simple. The sheer overwhelming amount of data available today in practically any area can barely be handled by human experts. Manual processing is not only time-consuming and often boring work for the highly educated domain experts but also very expensive. Aside from general-purpose text classification, it is also common to introduce domain specific background knowledge. Considering that the medical terminology is very complex and contains many synonyms and polysemes it seems obvious that domain specific optimization enables significant improvements over a generic system. The Unified Medical Language System (UMLS) Metathesaurus is a rich ontology composed of many different sources in the biomedical domain. By utilizing this vast amount of knowledge to enrich documents the noise arising from synonymous and ambiguous concepts can be minimized.

1.1.1 Text Classification

The automated classification of natural language text documents into a set of predefined classes is a subfield of information retrieval. A text classification (TC) system can be either single-label or multi-label. In single-label classification every document is assigned to exactly one class while a multi-label classification system assigns n class-labels to each document with $0 \leq n \leq |C|$. C is the set of all predefined classes relevant for the topic at hand. In binary classification each document must be assigned either to the class c_i or its complement $\bar{c}_i$. Typical applications for TC are text routing, text filtering, word sense disambiguation (WSD) or contextual advertising. A text routing application could for instance automatically decide in which subsections (e.g. *'politics'*, *'science'*, *'sports'*, ...) incoming news from a newswire are to be

placed. Contextual advertising chooses advertisements which relate to the context the user is currently reading. WSD is the task of finding the intended meaning for a polyseme. For instance can a *'cell'* be a small confined space or a functional unit of living organisms depending on the context in which the word is used. WSD can be seen as a special form of text classification [Fab02]. It is obvious that TC can be used to achieve a lot of different goals. This thesis focuses on binary text filtering which is the assignment of one of the two class labels *'relevant'* or *'irrelevant'* to each document. Such a system can be used to filter a stream of incoming documents and only select instances which might relevant for a particular topic. The TC system implemented for this thesis focuses on the detection of off-label drug use in biomedical papers.

1.1.2 Unlicensed and Off-label Drug Use

The U.S. National Library of Medicine defines unlicensed or off-label drug use as follows:

> *The practice of prescribing or using a drug outside the scope of the drug's official approved label as designated by a regulatory agency concerning the treatment of a particular disease or condition.* [1]

Off-label drug use arises through many pathways with the most common being prescriptions for unapproved clinical indications or unapproved age groups [Sta08]. In most countries (including Austria) the prescription of off-label pharmaceuticals is not prohibited by law. While the practice is very common, the specific application is often not supported by strong evidence. A report from 2003 showed that of 160 common drugs, 21% of all prescriptions can be classified as off-label [Sta08]. Some drugs (e.g. antipsychotics) were actually more often prescribed off-label than within the boundaries defined by the regulatory agency. The benefit-risk balance of these drug applications is either negative or unknown. There are many studies on off-label drug use but they are difficult to find because of variations in phrasing and focus of the studies [MMH12]. Manual search strategies are time-consuming and suffer from low precision. An automated text classification system which detects articles on off-label drug use in literature databases could be beneficial to a regulatory agency.

1.2 Problem Statement

Prior to this thesis, a study on off-label drug use in the literature databases Medline and Embase was conducted [MMH12]. Combined, these two databases contain over 24 million records of international biomedical literature. The study employed a complex search pattern to retrieve approximately 4419 Medline and 6240 Embase records containing specific keywords relevant to the topic of off-label drug use. A problem with this retrieval approach is that it is focused on

[1] `http://www.ncbi.nlm.nih.gov/mesh/68056687`

the presence or absence of popular keywords. Articles which do not use any of the predefined keywords in the search query can never be found. The main issues however are the rather low precision of this retrieval approach and the time consuming manual classification of the over 7000 unique articles. Each exported record contains the title of the article, assigned keywords, substance names, and in most cases, the abstract of the paper. For some instances the full article had to be retrieved manually to categorize the document since either no abstract was available or it was not informative enough to make a clear decision.

Considering the costs involved with the manual classification of over 7000 text documents by a human domain expert, a fully automated classification system seems most desirable. Such a classification system can be trained by manually labeled samples and can therefore be used for various classification tasks as long as appropriate training data is available. This thesis will investigate if an automated text classification system can be used to efficiently detect articles which deal with off-label drug use in medical literature databases. Other questions which will be examined in this thesis include: Which state of the art methods can solve the classification problem best, and what is their performance? Can the performance of the system be improved by adding domain specific background knowledge? How can the recall or precision of the system be increased, and how big is the trade-off between these two measures? How time-dependent is the optimized classification system?

All techniques are evaluated on the Medline and Embase test collections which were created during the preliminary study. Aside several common feature selection functions a new method, the class discrimination ratio (CDR), is presented and evaluated against other state of the art functions.

1.3 Outline

This thesis is organized as follows.

Chapter 2 presents an overview of state of the art techniques in the IR and TC field. It also recapitulates several related works which focus on the medical domain.

Chapter 3 starts with a detailed analysis of the test collection and afterwards focuses on the experimental design of the prototype application and the choice of optimal parameters.

Chapter 4 shows results for cross validation with different combinations of techniques and feature vector size. The best performing classifiers are trained on the Medline and Embase dataset and evaluated against the previously unseen validation set.

Chapter 5 compares the results to those achieved in related work and discusses open issues and future research.

CHAPTER 2

Background and Related Work

This chapter will provide an overview of state of the art machine learning (ML) techniques with special focus on text classification (TC). Some typical problems and possible solutions in document representation will be presented. Also a short introduction of the UMLS Metathesaurus will highlight its components and value for document enrichment. A detailed examination of commonly used performance measures and an overview about related work on medical IR systems is presented at the end of this chapter.

2.1 Terminology

The research area of machine learning, information retrieval and text classification is a broad one and so is its terminology. The primary research topic of this thesis is text classification which is also often referred to as *text categorization* and sometimes *topic spotting* [Fab02].

Since the system under investigation is trained with pre-labeled data the term learning automatically means supervised learning. The attributes used for training the classifier shall be referred to as features where all features combined form the feature vector. The term phrase is not necessarily used to describe a lexical phrase but should be considered an ordered group of at least two words. The concept *document expansion* or *document enrichment* is used in a similar way as the commonly used *query expansion*. Documents can be augmented with either phrases extracted from the collection or concepts from a knowledge source like the UMLS Metathesaurus.

2.2 Document Representation

As with any ML approach, the first task of text classification is to find a suitable representation of the data at hand. Texts are written to be human readable and differ very much in choice of words, sentence-/document-length, amount of distinctive words, complexity of sentence structures and so on. A set of optimal features extract as much information about the content of a document as possible while omitting the obvious redundancies. The choice of features has a direct impact on the performance of the resulting classifier.

2.2.1 Vector Space Model

After the choice of a set of k features that will represent all documents $d_i \in D$ we have a feature vector $\vec{v_i} = \langle w_{i1}, w_{i2}, ..., w_{ik} \rangle$ for each document. If a feature is not present in a document the respective value in the vector is zero. All feature vectors can be assembled in a matrix M_{ik} which is a compact representation of the corpus.

Bag of Words

The most obvious features of a text document are the individual words. We can assume that the appearance rate of a word in a document, the term frequency (TF), is indicative of its importance for the text. The document frequency (DF) is the number of documents in the corpus that contain the word. The bag of words (BOW) is the group of words that is chosen by some selection algorithm to represent all documents of a corpus. Typically a document collection contains a large quantity of distinct words of which only a small subset of words are chosen as features to represent the documents. Aside from performance gains, the reduction in feature space has also shown to improve generalization accuracy and decreases the risk of *overfitting* [Joa98].

Phrases as Features

While the BOW approach performs reasonably well and is widely adopted it only captures a certain aspect of the documents and completely misses all semantic information that is expressed by the order of words. An example would be the phrase *'offlabel prescription'* which indicates a special concept that is different from that of the individual words and which is lost in a simple BOW approach. This fact suggests that TC could be improved by using important and topic relevant phrases as indexing features. There are several different ways to define such phrases from the corpus, namely *syntactic phrases*, *semantic phrases* and *statistical phrases*.

Syntactic phrases are predefined syntactic structures of terms (e.g. noun phrases, verb phrases, ...). First a part of speech (POS) tagger is used to identify the lexical category of each word. A manually defined grammar is then used to identify maximal length phrases in the documents

[MCAC97]. Such a system would for instance identify the phrase *'offlabel prescription'* from the sentence *'Offlabel prescription is frequent in the setting of adult surgical intensive care unit.'* as a maximal noun-phrase since there are two nouns followed by a verb.

A semantic phrase is an actual concept name from a lexicon or a thesaurus. The difference to the other two types of phrases is that there is a limited amount of predefined features. New concepts introduced in the documents cannot be identified as a phrase and are therefore lost in the indexing process. Because of this fact, it is vital that the knowledge source covers the target domain extensively.

Statistical phrases are groups of words that appear often enough together in the corpus to suspect that the group represents a concept distinct from the meaning of the individual words. Another interpretation sometimes used in literature is the *n-gram* which represents a statistical phrase with *n* word stems and all its permutations [CMF01, MG98].

Results from literature show an unclear picture about the performance gain from adding phrases as features. Often an increase in performance in some cases and a decrease in others is reported [CMF01, MG98, MCAC97]. The results also indicate that a reduction of single word features in favor of phrases hurts performance. One negative impact of phrases on the effectiveness of an IR system is the often near synonymous characteristic of phrases and their individual words [Dol92].

2.2.2 Synonyms, Polysemes and Hyponyms

Two distinct words are considered synonyms if they have identical or almost identical meanings. An example from the medical domain would be *'carcinoma'* which is the medical term for the most common form of *'cancer'*. Polysemes on the other hand have the exact same spelling but different senses. A typical example for a polyseme is the word *'cell'* which can have many different meanings depending on the context in which it is used. Words belonging to these two categories can add quite a lot of noise to the data and they are omnipresent, especially in the medical domain [BCC04]. Because synonyms and polysemes have a rather significant impact on the performance of any IR system, techniques to compensate are an active research area. A common method to handle problems arising from synonymous terms is query expansion which can improve domain specific IR significantly [BCC04]. Its goal is to introduce synonymous terms to a query with the aid of a thesaurus. The desired effect is to also find documents that contain only the word *'carcinoma'* if the user entered the search term *'cancer'*.

The task of finding the appropriate meaning for a polyseme is called word sense disambiguation (WSD) and is especially important for natural language processing (NLP) tasks. WSD takes the vicinity of the word into account to find the intended meaning of the word in its respective context with the aid of background knowledge like a thesaurus. One example from literature is to use the conceptual density of concepts in WordNet to disambiguate between polysemes [AR96, HSS03]. Another is to use tfidf similarities between the document and the Wikipedia

articles of the concepts in question to disambiguate between the possible senses [WHZC09]. For instance can the term *'jaguar'* refer to a car or an animal depending on the context in which it is used in the text. In Section 3.5.3 a strategy for WSD based on conceptual density will be presented using the UMLS Metathesaurus as background knowledge.

A word is called a hyponym of another word (its hypernym) if it represents a more specialized concept. In computer science these relationships are often called *isa*, e.g. *'sparrow isa bird'* and are used in ontologies to build hierarchical structures of concepts. By utilizing this information an IR system could generalize the word *'sparrow'* to the concept *'bird'* which expresses additional information that was previously only available on an implicit basis. Experiments performed with document clustering showed an improvement in effectiveness when adding generalized concepts up to a depth of 5 [HSS03]. A TC system which uses Wikipedia as a source of background knowledge achieved the best results with adding only the direct hyponyms [WHZC09].

2.3 Feature Space Reduction

Since the amount of distinct terms and concepts in the corpus is normally very large, some form of reduction process is necessary. Dimensionality reduction decreases the computational complexity of the classification task on one hand, and improves its performance on the other. Aside from other positive effects on the statistical quality of document representation an appropriate reduction also decreases the risk of overfitting the classifier to the training data.

2.3.1 Stemming and Reduction with a Predefined Stop Word List

Stemming is the process of reducing a word to its morphological root or *word stem*. Stemming is used successfully in many state of the art IR applications [WHZC09, AMWZ09, HSS03, CMF01, MCAC97, YP97] but there are also indications that it can hurt performance in some cases [Fab02]. The general assumption is that two words which have the same word stem also represent the same concept and can be exchanged for the document indexing. For example will *'used'* and *'using'* both be stemmed to *'use'*. This will not only reduce the dimensionality of the feature vector but is also expected reduce the stochastic dependency between the morphological variants which in turn improves the document representation. Of course the opposite is also possible, two words have the same morphological root but represent different concepts. The two words *'complication'* and *'complicated'* are both stemmed to the same word stem *'complic'* although they have quite a different meaning.

Another relatively popular approach to decrease the size of the feature vector is to compile a list of known stop words by hand. These are words which contain no information about any specific topic and just serve a functional purpose. Many stoplists have been created and successfully used to improve the performance of IR systems in the past [HSS03, RS02, YP97,

Sal71]. Stop words can also be identified by their DF in a test collection which, however, has the disadvantage of possibly also removing popular terms that actually contain information about the class distribution.

Empirical evaluation of stemming and the use of a predefined stop word list was carried out for this thesis and is presented in Section 4.1.1. In addition several examples from the test collection are given in which these two techniques can actually decrease the performance of an IR system.

2.3.2 Feature Selection

A very basic method of dimensionality reduction (DR) is feature selection which is selecting a subset of n features from the original feature vector $|\vec{v}| = k$ where $n < k$. In most ML tasks some form of manual feature selection is initially performed when searching for a suitable representation for the data. Further reduction through the selection of the most important features improves performance and reduces problems arising from overfitting to the training data. Finding suitable reduction functions for IR and TC tasks is covered in many articles. A selection of popular functions is presented in Section 3.6.3 and an evaluation of the performance in Section 4.1.2.

2.3.3 Feature Extraction

Feature extraction is another approach to DR that attempts to create n new features from an original k features $n < k$. Its motivation is to compress the feature space and keep as much information of the original data as possible. The original vector $|\vec{v_o}| = k$ is transformed into the new $|\vec{v_n}| = n$ by some transformation function. One of the most popular techniques for feature extraction in TC is latent semantic indexing (LSI) which is explained in Section 2.4.3 in more depth. Another method is term clustering which attempts to form clusters of words which have a high co-occurrence and are therefore assumed to be either near-synonymous or semantically related. These clusters are then used as indexing features instead of the single words. It has been shown that clusters of words and phrases, generated with a reciprocal nearest neighbor (RNN) algorithm, perform not as good as pure word based indexing [Dol92].

2.3.4 Overfitting

In ML a learning system or classifier is trained with pre-labeled test data. The intended goal is of course that the system will be able to generalize from the training data and classify previously unseen data in a similar manner. Overfitting happens if the classifier learns the special characteristics of the training data but fails to learn the constitutive characteristics of the classes. Overfitting happens if the classifier is tuned for high effectiveness on the training data. Such

a classifier can classify the training data quite well but fails to generalize on unseen data. The problem can be prevented to some degree by using k-fold cross validation [HCL03].

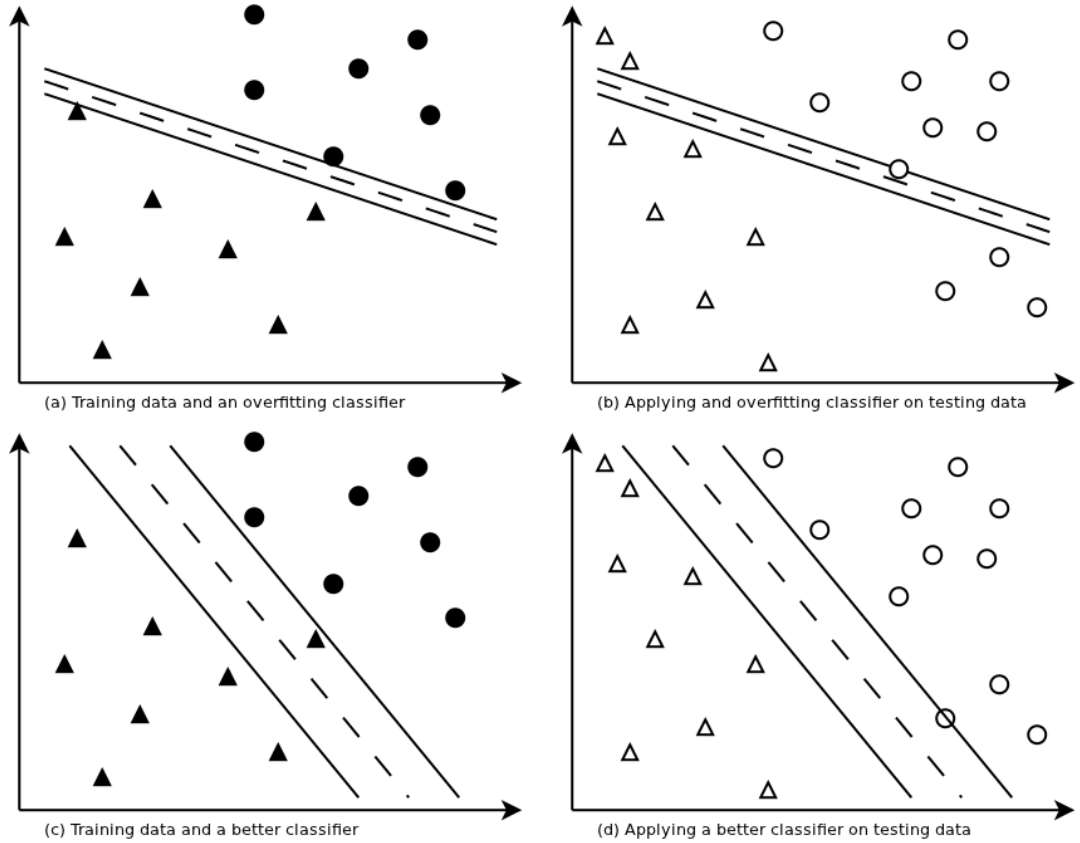

Figure 2.1: Overfitting and a better classifier (● and ▲: training data, ○ and △: testing data), [HCL03].

Figure 2.1 shows an example of an overfitting classifier (a, b) and a better classifier (c, d). The training of the overfitting classifier (a) misses the fact that a better separation of the classes is possible. When presented with unseen data (b) it performs bad. A more generalizing classifier (c) learns the underlying characteristics of the classes and performs better when presented with unseen data.

2.4 Feature Weighting

Since a classifier does not accept free text of arbitrary length, all documents have to be mapped to a uniform feature vector. After the choice of appropriate features to represent all documents in the corpus, suitable values for the features are calculated with a term-weighting function. The

expression 'term' is used here synonymous with 'feature'. However, it should be noted, that phrases and UMLS concepts are also used as features in this thesis. The result of this process is a compact representation consisting of k values for all of the i documents. This index matrix can directly be used as input for a classifier or by a classifier-building algorithm. The experiments performed for this thesis focus on three popular weighting functions: tfidf, language modeling and latent semantic indexing.

2.4.1 TFIDF

Term frequency times inverse document frequency (tfidf) is a standard term-weighting function used by many IR systems. The most basic form is $tf(t_k)/df(t_k)$ however many variants exist. The intuition behind this formula is that a term's importance for a document increases with the number of occurrences in the document $tf(tk_k)$ and decreases with the number of occurrences in the corpus $idf(t_k)$. Additionally a normalization to the document length can be carried out to compensate for the length of the text. Previous work shows that length normalization is not necessary for short documents and otherwise only if the deviation in length is large [SB88].

2.4.2 Language Modeling

Statistical language modeling is a technique used in many NLP applications. For every document in the collection a probabilistic language model is estimated individually. The model is then used to predict the probability of a term in the associated document. In contrast to other statistical models like the 2-Poisson model or the n-Poisson model, language modeling (LM) has the advantage of making no unwarranted prior assumptions about the parameters of the data. Models are non-parametric and estimated for each document individually. By generating a model for each document, LM relaxes the assumption that the collection is generated by a predefined set of classes (language models). Indexing by language modeling has been reported to perform significantly better than tfidf weighting [PC98].

For every document d a language model M_d is created which estimates the probability of a feature t_k for the document d, $\hat{p}(t_k|M_d)$. The maximum likelihood estimate for a term t_k is $\hat{p}_{ml}(t_k|M_d)$ (Formula 2.1) which is the mean probability for receiving t_k if we select a random term from document d with length dl_d.

$$\hat{p}_{ml}(t_k|M_d) = \frac{tf(t_k, d)}{dl_d} \tag{2.1}$$

One problem with the simple estimator is that it is created with only very limited data. This issue can be solved by calculating an estimate from all documents d from the corpus that contain the

term t_k. $\hat{p}_{avg}$ (Formula 2.2) is the mean probability to pick t_k from a document containing it.

$$\hat{p}_{avg}(t_k) = \frac{\sum_{d_{(t_k \in d)}} p_{ml}(t_k|M_d)}{df_{t_k}} \tag{2.2}$$

While the previously modeled estimator is more robust, it assumes that every document that contains t_k is drawn from the same language model. There is of course the risk that this assumption is wrong which is taken into account by $\hat{R}_{t_k,d}$ (Formula 2.3). $\hat{R}_{t_k,d}$ decreases the impact of the estimate from the whole corpus $\hat{p}_{avg}$ if the term t_k occurs less frequent in the document d as in other documents containing it. $\bar{f}_{t_k}$ is the mean term frequency of term t_k in documents where it occurs.

$$\hat{R}_{t_k,d} = \left(\frac{1}{1+\bar{f}_{t_k}}\right) \cdot \left(\frac{\bar{f}_{t_k}}{1+\bar{f}_{t_k}}\right)^{tf_{t_k,d}} \tag{2.3}$$

Putting the parts together gives the equation depicted in Formula 2.4. $\hat{p}(t_k|M_d)$ uses an estimate drawn from the single document and one drawn from the whole collection. The previously described risk function $\hat{R}_{t_k,d}$ weights the two estimates depending on the risk that the document d is draw from a different language model than the average document containing t_k. If a term t_k does not occur in document d it is nevertheless not impossible. The assumption is made that term t_k occurs in the document with the same probability as it occurs in the collection. cf_{t_k} is the term count for t_k and cs for all terms in the corpus.

$$\hat{p}(t_k|M_d) = \begin{cases} p_{ml}(t_k,d)^{1-\hat{R}_{t_k,d}} \cdot p_{avg}(t_k)^{\hat{R}_{t_k,d}} & \text{if } tf_{t_k,d} > 0, \\ \frac{cf_{t_k}}{cs} & \text{else.} \end{cases} \tag{2.4}$$

2.4.3 Latent Semantic Indexing

While most traditional approaches to text indexing use some form of feature selection, LSI attempts to create a new set of features to represent all documents of the corpus. The idea behind LSI is based on the general assumption that there is some higher-order structure behind the association between terms, documents and the classes. These structures form implicit concepts which can be used as indexing features. The advantages of this approach is the reduction of noise resulting from the choice of different words for the same underlying concept and the efficient compression of the feature vector to a small fraction of the original size. LSI was first introduced to IR in 1990 [DDF$^+$90]. The proposed approach uses singular value decomposition (SVD) to break the term-document matrix down into linearly independent factors. The matrix M_{ik}

contains the term-frequencies for every term t_k in every document d_i of the corpus D. Formula 2.5 shows the breakdown of the matrix M into the matrices of the left and right singular vectors U and V. S is the diagonal matrix of singular values.

$$M_{ik} = U_{ii} \cdot S_{ik} \cdot V_{kk}^T \tag{2.5}$$

These three resulting matrices contain the document-document, document-term and term-term similarities. In practice, most factors are very small which is why reduced forms of SVD are commonly used to build the index. In experiments, collections with 5000-7000 indexing terms were reduced by LSI to 20-100 factors [DDF$^+$90]. Despite the high information compression the results were superior to a term-only approach.

2.5 Empirical Evaluation

Because of the complexity of state of the art information retrieval systems and the high dimensionality of the input data, the evaluation of the performance of such a system can only be done by empirical experiments. Meaningful and standardized performance measures are very important to enable researchers to accurately measure optimizations of their system on the one hand and compare it to related work on the other.

2.5.1 Evaluation Measures

There are several measures to describe the performance of an IR system depending on the type of classification. To evaluate a binary classifier one can create a two-way contingency table, as shown in Table 2.1, which contains the four values true positive (TP), false positive (FP), false negative (FN) and true negative (TN) [Yan99, Fab02].

Common performance measures which can be calculated from the values of the contingency table are *accuracy* (Formula 2.6), *precision* (Formula 2.7) and *recall* (Formula 2.8).

Table 2.1: Two-way Contingency Table.

	Relevant documents	Irrelevant documents
Classified as relevant	TP (Correct result	FP (Unexpected Result)
Classified as irrelevant	FN (Missing result)	TN (Correct absence of result)

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.6}$$

$$precision = \frac{TP}{TP + FP} \tag{2.7}$$

$$recall = \frac{TP}{TP + FN} \tag{2.8}$$

The problem with these performance measures is that they cannot be used by themselves to optimize or evaluate a classifier. Usually the documents are not distributed evenly over the classes. The Medline test set (described in more detail in Section 3.1) for example contains 2168 relevant and 3179 irrelevant documents. A classifier that rejects all documents would have an accuracy of 59.45%. This could be much worse, however, in a real world example with only a handful of relevant documents per 1000 documents. In general there is a tradeoff between recall and precision with the break-even point (BEP) at $precision = recall$. If a classifier would accept all documents as being relevant it would have a perfect recall of 100%, but at the cost of a terrible precision of 40.55%. Which performance measure, recall or precision, is more important depends on the task. However, in practice a reasonable balance of both measures is desirable. The F_1 measure (Formula 2.9) is an example of a single numbered performance measure that takes both, recall and precision, into account. If $precision = recall$ then $F_1 = precision = recall = BEP$, otherwise F_1 is always lower than the other performance measures except the BEP.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.9}$$

F_β (Formula 2.10) on the other hand allows for a different weighting of recall and precision. For $0 \leq \beta < 1$ precision is considered more important. For $\beta = 1$, F_β coincides with F_1. For $\beta > 1$, recall gets preference.

$$F_\beta = \frac{(\beta^2 + 1) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{2.10}$$

According to Yang, F_1 is a good choice for evaluation and parameter optimization as it resembles an even balance between precision and recall an avoids problems arising from unevenly distributed classes [Yan99].

2.5.2 ROC Analysis

A commonly used visualization method of the performance of a classifier is the receiver operating characteristic (ROC) curve. Originating from signal detection theory, ROC graphs are used to depict the tradeoff between the true positive rate and the false positive rate. The graph is two-dimensional with the true positive rate plotted on the y-axis and the false positive rate plotted on the x-axis. Figure 2.2 shows a simple ROC graph with five discrete classifiers [Faw06].

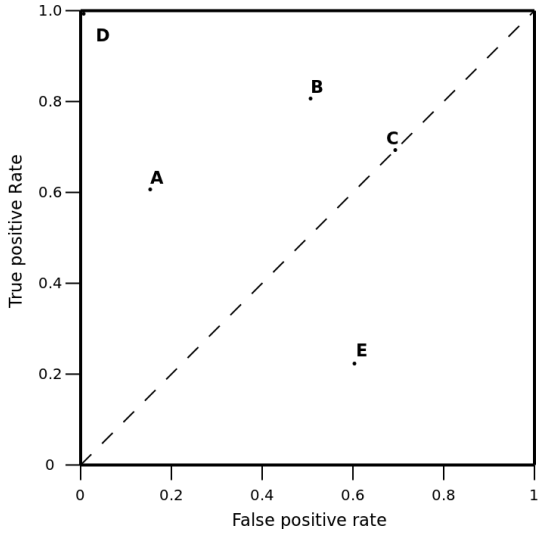

Figure 2.2: Basic ROC graph with five discrete classifiers [Faw06].

Classifiers on the diagonal $x = y$ (C) achieve the same results like a classifier that is randomly guessing class memberships. They can be considered to have no information about the classes. Any classifier below the diagonal line (E) performs worse than random guessing. Those classifiers actually posses information but apply it incorrectly. The ideal classifier lies in the upper left corner (D) of the ROC space and has a perfect true positive rate of 1.0 and a false positive rate of 0. In general, classifiers which are located on the lower left of the ROC space (A) are considered *conservative* since they have a lower true positive rate but also expose fewer false positives. Classifiers on the upper right side of the graph (B), on the other hand, are thought of as *liberal* because they identify more positive samples at the cost of a higher false positive rate. Conservative classifiers need strong evidence and tend to have a better precision, liberal classifiers have a higher recall at the expense of more false positives.

The classifiers depicted in Figure 2.2 are all discrete classifiers which output a class label for every data instance provided. Some classifiers return a probability for the class memberships instead. If the probability for a class is above a certain threshold, the classifier reports a positive class membership for the instance in question. By varying the threshold between 0% and 100%, a tradeoff between type-1 (FP) and type-2 (FN) can be achieved. This also allows us to draw a

curve through ROC space for a single classifier.

A convenient property of ROC curves is that they are comparable for different class distributions and error costs [Faw06]. While the operating point may change, the graph stays the same. A popular performance measure which can be calculated from the ROC graph is the area under the ROC curve (AUC). A classifier will typically have an AUC between 0.5 (random classifier) and 1.0 (perfect classifier). The score is a single performance value which also takes the tradeoff between precision and recall into account. A classifier with a higher AUC must not always perform better in any operating point, however, it is a very effective single value performance measure. The AUC is also the probability by which a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. Section 4.7 depicts the ROC curves of two implemented classifiers and compares their performance according to their points in ROC space.

2.5.3 Establishing a baseline

Comparing the work of the scientific community in the area of IR is often hard because of the use of many different test collections and weak baselines. In a comparative study performed, the authors discovered that in the period between 1998 and 2008 no measurable improvement can be observed [AMWZ09]. The 106 surveyed publications use a total of 83 different test collections making it very hard to compare the results. Also many publications use non-competitive baselines which are below the average of the previously achieved results, yet many of these claim statistical significance.

One aspect to consider is the additivity of single improvements. An experiment carried out used 6 independent options in the Indri system [AMWZ09]. The base system with no options turned on is the baseline on which all combinations of options are evaluated. The results of the experiment suggests that improvements are additive on average but not all combinations improve performance over the baseline. The additivity of techniques has to be confirmed for individual cases.

2.5.4 Tuning Noise

A big problem in ML is the way in which new features are crafted. Not unlike other scientific areas, progress is achieved with the process of experimentation and validation. Beginning from a baseline the researcher starts crafting features and validating them on the given dataset. In practice this is often an iterative process of trial and error which tends to fit the features and parameters of the model to the evaluation measure and the data collection. Also the risk of elevating quasi-random features which produce small gains on the evaluated dataset increases.

Blanco and Zaragoza showed in an impressive way how the introduction of a random perturbation to the document-score of an IR system can produce seemingly statistically significant

improvements over the baseline [BZ11]. The problem is the way statistical experiments are usually carried out. The random perturbation increases performance in some runs and decreases it in others. In the experiments, the impact of the random parameter λ is varied and for each value 200 runs are performed. From each, only the score of the winning λ is used for the *Wilcoxon signed-rank* test. In statistics this is called a *multiple comparisons* setting and depicts the problem that the more variations are considered the more likely a statistical significance is found. The solution is to decrease the p-value for these kind of tests. In practice one has to be very careful not to overfit parameters to the specific test conditions. Also if the results have to be compared to a baseline with a statistic test, 1-sided statistical tests should be used as they seem to be more robust [BZ11].

2.6 Unified Medical Language System

The UMLS is a collection of biomedical vocabularies developed and maintained by the US National Library of Medicine. It incorporates many different languages and provides mapping structures to allow translation between terminology systems. The UMLS 2011AB[1] release contains more than 2.6 million concepts and 8.6 million unique concept names from 161 source vocabularies. The UMLS includes tools to customize the Metathesaurus and limit the data to certain source vocabularies and languages. It is also possible to generate customized database load scripts which enable direct access to the ontology. All vocabularies are available at no charge for research purposes, but registration is required. The core components of the UMLS are the *Metathesaurus*, the *Semantic Network* and the *SPECIALIST Lexicon* [Bod04].

2.6.1 Metathesaurus

The Metathesaurus is a collection of interrelated biomedical concepts represented by a set of synonymous names which in the UMLS terminology are called *atoms*. Concepts are linked to other concepts by different relations depending on the source vocabulary. The most common relationship is *'isa'* which resembles a hierarchical connection or *Hyponymy*. An example of such a relationship is *'Heart attack' isa 'Structural disorder of heart'*. Other examples for relationships are *'same_as'* and *'part_of'*. There are also special concepts like *'Duplicate concept'* which indicate a special characteristic of the linked concepts. The MetaMap program [Aro01] is distributed as part of UMLS. It identifies concepts in free text and returns a ranked list of Metathesaurus concepts. MetaMap takes advantage of the SPECIALIST Lexicon to generate variants of phrases and to identify them in free text.

[1] UMLS 2011AB release: `http://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/release/notes.html`

2.6.2 Semantic Network

The Semantic Network defines semantic types and assigns one or more to each Metathesaurus concept depending on their role in the source vocabulary [LHM93]. The semantic types are, like concepts, hierarchically organized and categorize the individual concepts into broad subject groups. The type *'Plant'* for instance has an *isa*-relationship to the semantic type *'Organism'*. The 2011AB release includes 133 different semantic types and 54 relationships[2].

2.6.3 SPECIALIST Lexicon

The third core component of the UMLS is a biomedical lexicon which was created for the SPECIALIST Natural Language Processing System. It is intended as an English lexicon with focus on the biomedical domain. The lexicon contains entries with different spelling and part of speech variants. Each entry can be single or multi word (phrases) and contains a base form. The 2011AB version of the SPECIALIST Lexicon contains more than 1.5 million entries.

2.7 Medical Information Retrieval

Since there appear to be none or only marginal improvements in IR in recent years, as indicated by a rather extensive survey of IR research [AMWZ09], many researchers integrate background knowledge to enrich document representation. There are many domain independent knowledge sources available for free on the internet. WordNet is frequently used because it is a relatively easy to use lexical database of English words, grouped into sets of synonymous concepts and linked by semantic and lexical relations [HSS03, AR96]. Another obvious source of background information is Wikipedia which contains an enormous amount of data including hyperlinks between concepts and hierarchical categories. Wikipedia has been used successfully to improve TC on several test collections inducing OHSUMED [WHZC09].

While background knowledge in general seems to improve results, domain specific knowledge seems a better choice since it is a better reflection of the used terminology. The Unified Medical Language System (UMLS) is a collection of many medical knowledge sources [LHM93, Bod04]. The MetaMap program [Aro01] uses this medical Metathesaurus to map free text to concepts defined by UMLS. It was originally implemented to improve retrieval of domain specific bibliographic material but has since been used in several data mining efforts. A retrieval system [Aro01], enhanced with concepts identified by MetaMap, has been evaluated against the statistical IR systems SMART and INQUERY. The system showed an improvement in performance through the use of query expansion. Another system [HPDD00] which implemented query expansion with concepts from the UMLS Metathesaurus on documents from the OHSUMED test collection, showed mixed results. The query expansion actually degraded

[2]UMLS Semantic Network: `http://www.ncbi.nlm.nih.gov/books/NBK9679/`

retrieval performance overall. However, approximately a third of the queries showed improvement.

Textpresso is a retrieval and extraction system for biological literature which employs the Gene Ontology database [MKSS04]. The effectiveness of the system in automatic identification of journal articles was evaluated on full article texts and abstracts which were categorized into six predefined classes by a human expert. In a keyword search, 94.7% of the expected results were found when searching in full texts. However, the same search in abstracts had a recall of only 44.6%. The big difference in the two results can be explained by the fact that specific keywords were more likely to appear in full text than in the abstract. The system achieved slightly better precision when only using the abstracts. Especially single keyword searches often returned large numbers of irrelevant articles. The authors concluded that a limitation of Textpresso lies in the flaws of the used ontology and that the use of other data sources like UMLS and SNOMED will help to increase the specificity of the system [MKSS04].

Domain-specific synonym expansion is considered a big challenge for IR systems. For this reason, the MultiText group of the TREC 2004 Genomics track focused on developing a biomedical retrieval system which uses several domain-specific knowledge sources to improve retrieval performance [BCC04]. Besides the generation of lexical variants for specialized biomedical terms, the system focused on synonym expansion since the use of different names and symbols was considered one of the biggest problems for the specific task. Three knowledge sources were integrated to deal with acronyms, synonyms and symbols for genes and proteins. AcroMed contains acronyms of medical terms automatically generated from Medline abstracts. The euGenes and LocusLink databases were used to provide mappings between gene symbols and their full names. The generation of lexical variants as well as the expansion of acronyms improved the system according to the experiments performed. However, adding synonyms from the gene databases had a negative effect on precision while increasing the recall slightly. The suspected reason for this is that the amount of added alias symbols is far too high, which increases the risk of query drift.

Since ontologies are expensive to produce, work has been conducted to automatically derive concept hierarchies from text using clustering techniques [BCH06]. The created ontology is then compared to the MeSH Tree Structures ontology in a text classification experiment. Only noun-phrase concepts and single terms were used for indexing. The system which used the automatically generated ontology performed nearly as good as the one using the MeSH ontology. Both systems outperformed the simple BOW approach.

SNOMEDCT has also been used in an experiment to measure semantic similarity and relatedness between medical concepts [PPPC07]. Techniques are explored which have previously only been used with domain independent knowledge sources like WordNet. The techniques include several path based measures which use characteristics of the connection of the two concepts in the ontology. Another measure of similarity is the *information content* which counts the frequency of a concept and its subsumed concepts in the corpus. The frequent co-occurrence of two concepts in the corpus is also indicative of semantic relatedness. This measure is called

context vector measure and defines the semantic similarity between two SNOMEDCT concepts. It is calculated for all concepts whose frequency in the collection exceed a predefined threshold.

2.8 Summary

IR is a broad field of research with countless techniques which are often difficult to compare since researchers tend to use different test collections and evaluation measures. Synonymous and polysemous terms and concepts have a fairly big negative impact on the performance of TC systems, especially in the complex biomedical domain. Several systems were introduced which solve these issues by enriching the documents with background knowledge. The most extensive knowledge source for the biomedical domain is the UMLS which is described in more detail in Section 3.5. The Metathesaurus and Semantic Network will be used to improve the classification performance of the system evaluated in this thesis.

Another issue previously discussed is overfitting a classifier to the training set or even to quasi-random features. This can, in part, be prevented by the use of 10-fold cross validation for parameter optimization. F_1 is a suitable single-value performance measure and will be used to compare the performance of different classifiers.

The choice of appropriate features is an important first step of the indexing process. Common features for TC are single words (BOW), statistical phrases and semantic phrases (concepts from a lexicon). To reduce the size of the feature vector and decrease the overfitting effect, a subset of features has to be selected according to a feature selection function. The most frequently used feature weight functions from literature will examined in the next chapter. In addition, a new selection function, the class discrimination ratio (CDR), will be presented and evaluated against state of the art functions in Section 4.1.2.

The three introduced indexing weights tfidf, LM and LSI will be implemented, and evaluated in Section 4.2.

CHAPTER 3

Generic and Domain-Specific Optimization of TC

The focus of this chapter is to investigate state of the art text classification techniques and to assess the potential of domain specific optimization. The binary classification task is to detect documents which deal with the topic *'off-label drug use'*, as previously defined in Section 1.1.2. In a first step the datasets of the two sources, Medline and Embase, shall be examined in great detail to outline the boundaries for the classification task. A detailed description of the application design and the various modules will follow. Also a novel feature selection function, namely CDR, will be presented which shows better results than other functions used in related work. This chapter will close with the empirical examination of several parameters of the classifier.

3.1 Dataset

The training and validation data which forms the basis for the experiments performed for this thesis are exported records from the medical literature databases Medline[1] and Embase[2]. Embase claims to contain all, approximately 19 million, records from Medline as well as over 5 million additional records. The exported data contains the title and, in most cases, the abstract of an article published in a medical journal since 1966 (Medline) and 1947 (Embase), as well as other additional data like medical subject headings (MeSH) tags.

In a first attempt to retrieve the articles of interest, a domain expert performed a manual search in both databases [MMH12]. The queries employ a boolean 'OR' to connect several sub-queries

[1] http://www.ncbi.nlm.nih.gov/pubmed/
[2] http://www.embase.com/

and a boolean 'NOT' to exclude irrelevant articles. The search strategy also includes a reduction of certain words to their stem and an 'adjacent'-operator that allows random words between two search terms. To improve the search strategy, combinations of 73 individual queries were tested. Table 3.1 depicts several sub-queries from the study which clearly show the complexity of the topic. Query 1 has the highest individual precision but its recall on the combined dataset is quite low. More complex queries try to capture individual cases of off-label use (query 3). The search strategy also contains queries to exclude articles (query 5). The search strategy with the best precision for the combined dataset had an overall recall of 49% at a precision of 84%. Depending on the combination of the queries either the recall or the precision of the strategy can be maximized. The study also showed that the recall of the queries is much higher if only the MEDLINE dataset is used [MMH12].

Table 3.1: Excerpt of several search queries from the original study on off-label drug use [MMH12].

	Query	# Relevant	# Irrelevant	Recall (Full Set)	Precision
1	off label*.af.	1663	307	40.9	84.4
2	(non evidence base* us*).af.	2150	1390	52.86	60.73
3	((no* licen?ed for adj3 use*) not now licen?ed).af.	1929	1058	47.43	64.58
4	((inappropriate us* and indication) not (antibiotic* or antimicrobial)).af.	2065	1233	50.77	62.61
5	not (stent* or veterinar*).af.	1992 [3]	380	48.98	83.98

The retrieved dataset contains 4347 Medline and 6238 Embase documents. The dataset also contains several broken or duplicate records which are removed in the parsing process. The retrieved records were manually labeled by the domain expert. In this time consuming process, 2168 Medline and 3869 Embase records were labeled as being 'relevant' for the topic, i.e., 'off-label drug use'. The precision of the initial search is, as to be expected, not very good with 66.6% for the search in Medline and 72.5% for the search in Embase. It is difficult to make an estimate about the recall of the initial search as, according to the domain expert, many topic-relevant documents don't use any explicit words that identify them as being *'off-label'*.

Since all records were retrieved with a complex query there is a strong bias towards certain words. Because of this bias it seems reasonable to expand the corpus by 1000 random Medline records to improve detection of true negatives (irrelevant documents). The random records were selected from Medline by generating random IDs and checked by the domain expert to ensure that they are not topic relevant. For the rest of this thesis the randomly selected records are considered to be part of the Medline dataset.

[3] This sub-query is used to exclude irrelevant records from the search and is only evaluated together with other sub-queries.

3.1.1 Ovid-XML File Format

The exported records from Medline and Embase are stored in the Ovid-XML-Output[4] file format. The only difference between the exported data from Medline and Embase is the name of several of the tags. Four fields of the XML records were chosen as text representation of the article: *'Title'*, *'MeSH Subject Headings'*, *'Abstract'* and *'Name of Substance'*. Every record also contains a unique identifier which allows detection of duplicates and tracking of the documents throughout the processing pipeline. 1851 records of the merged dataset contain no abstract. However, all other fields were present in all instances.

3.1.2 Training Set

All initial tests, including the parameter selection, were performed on the Medline dataset. An evaluation on the whole dataset (Medline and Embase combined) is performed for the most promising set of techniques in Section 4.5. The combined dataset is also used to train the classifiers for the final validation against previously unseen data (Section 4.8).

The Embase and Medline datasets overlap in many documents. When merging the two corpora, duplicate documents are identified by their ID and title. The titles are converted to lower-case and stripped of any non [a-z] characters before comparison. For all duplicates the longer version is kept since in several cases the length of the abstracts differ between the two corpora. After this merge operation 8118 documents (4236 relevant and 3882 irrelevant) remain in the combined dataset including the random Medline documents.

Figure 3.1 shows the distribution of distinct terms per document in the merged corpus. Of the 8118 records in the corpus, 1525 contain below 50 distinct terms before any reduction process and can therefore be considered to be badly represented. After stemming, stop word removal and reduction to 5000 terms (selection with CDR-score, see Section 3.6.4) 238 records are represented by less than 10 terms.

As there is a change in terminology in all domains over time, it is also important to consider the date of the records in the dataset. Only 13 relevant and 248 irrelevant of the retrieved records were published before 1980. 80% of all records have a publication date later than 1998 which makes the dataset fairly up to date. Figure 3.2 indicates that in recent years, much more articles which cover the topic of off-label drug use have been published. Approximately 86.6% of all relevant samples have been published in the 10-year period between 2002 and 2011. Since all samples in the validation set are from the year 2011 only a minimal shift in terminology is to be expected between the training and the validation data.

[4] `ftp://ftp.ovid.com/support/Ovid/software/ovidxmloutput.dtd`

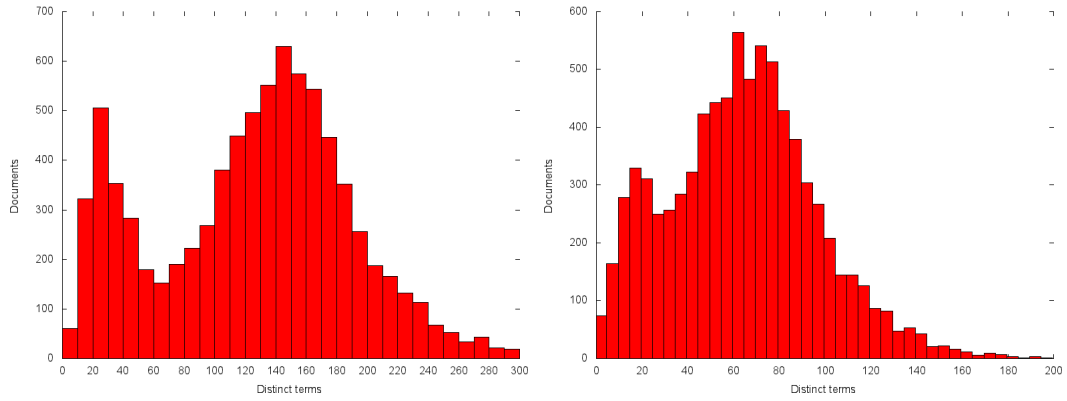

Figure 3.1: Distribution of Medline+Embase documents according to the document-length (number of terms), without (left) and with (right) reduction to 5000 terms.

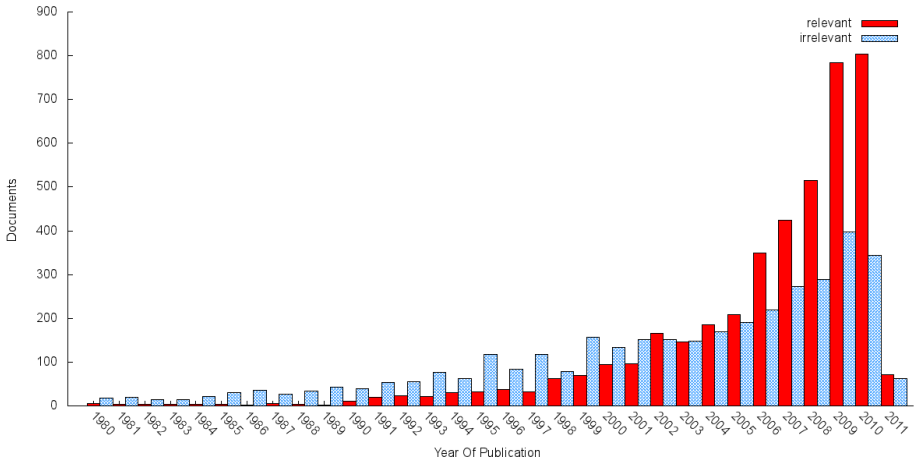

Figure 3.2: Medline+Embase records over time since 1980.

3.1.3 Validation Set

The dataset used to validate the results from the cross validation consists of 500 records exported from Medline of which 79 are labeled as relevant. The validation samples are recently published articles from 2011 and are not found within the training set. The histogram in Figure 3.3 shows the distribution of distinct terms per document after reduction to the terms selected from the training set. Out of the 5000 indexing terms selected from the training set, 1708 were present in the validation set. By average, a document in the validation set was represented by 53 terms, 19 were represented by less than 10 terms.

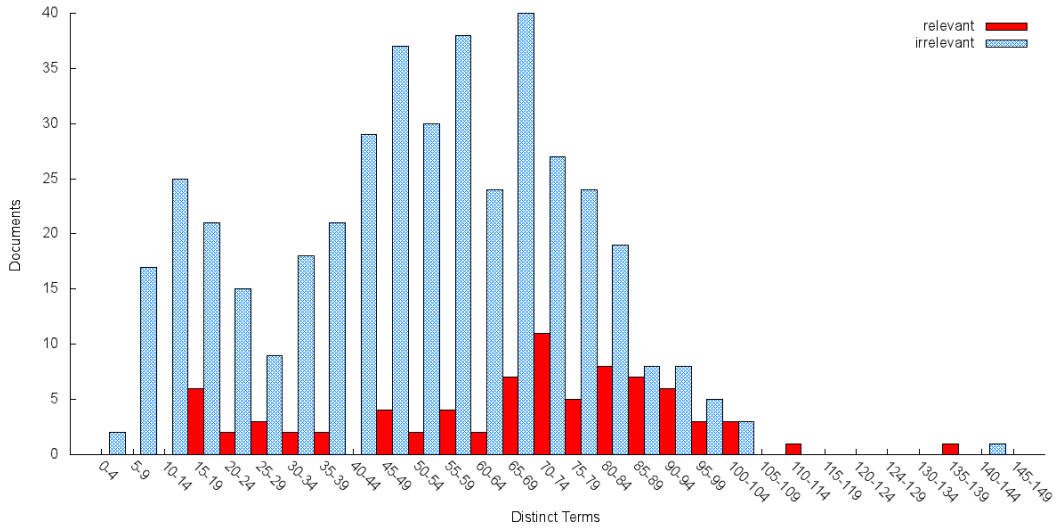

Figure 3.3: Distribution of documents in the validation set according to the document-length (number of terms), after reduction to the indexing terms selected from the training set.

3.1.4 Time Dependency

Since the data collection consists of articles from a rather long time span (65 years), it must be assumed that there are significant changes in terminology as well as a general shift of focus in respect to the topic of interest. Figure 3.4 depicts the occurrence rate of eight terms and two phrases over time relative to the amount of documents available for the specific year. The selected features are within the best 1%, according to their CDR-scores (see Section 3.6.4), and can therefore be considered to be important for the collection. The phrase 'stent thrombosi' appears in many recent irrelevant documents, published after 2005, because drug-eluting stents are sometimes used off-label and have been approved by regulatory agencies around that time period. These documents were selected in the initial search because they contain the phrase 'off label'. However, stents are medical devices and not drugs which makes these articles irrelevant for the research study.

An important aspect of a feature is the time of introduction to the collection. For instance is the term 'blood' present in old documents while 'fluoxetin' (better known by its trade name 'Prozac') only appears in documents which date to 1998 and later. In fact, many recently approved drugs are relevant to off-label use (e.g. 'duloxetine', 'ranibizumab') which clearly shows the importance of having sufficient up-to-date training samples.

The importance of a term for a specific class can change over time. For instance are the terms 'blood' and 'acid' quite indicative for a document to be irrelevant to the topic of off-label use before 1990. Documents which contain these terms and date after 2000, however, are more likely

to be relevant to the topic. The term 'rat' is an example for a feature which almost exclusively occurs in irrelevant documents over the whole timespan of the collection. This is due to the fact that drug use with animals is not considered relevant for the topic.

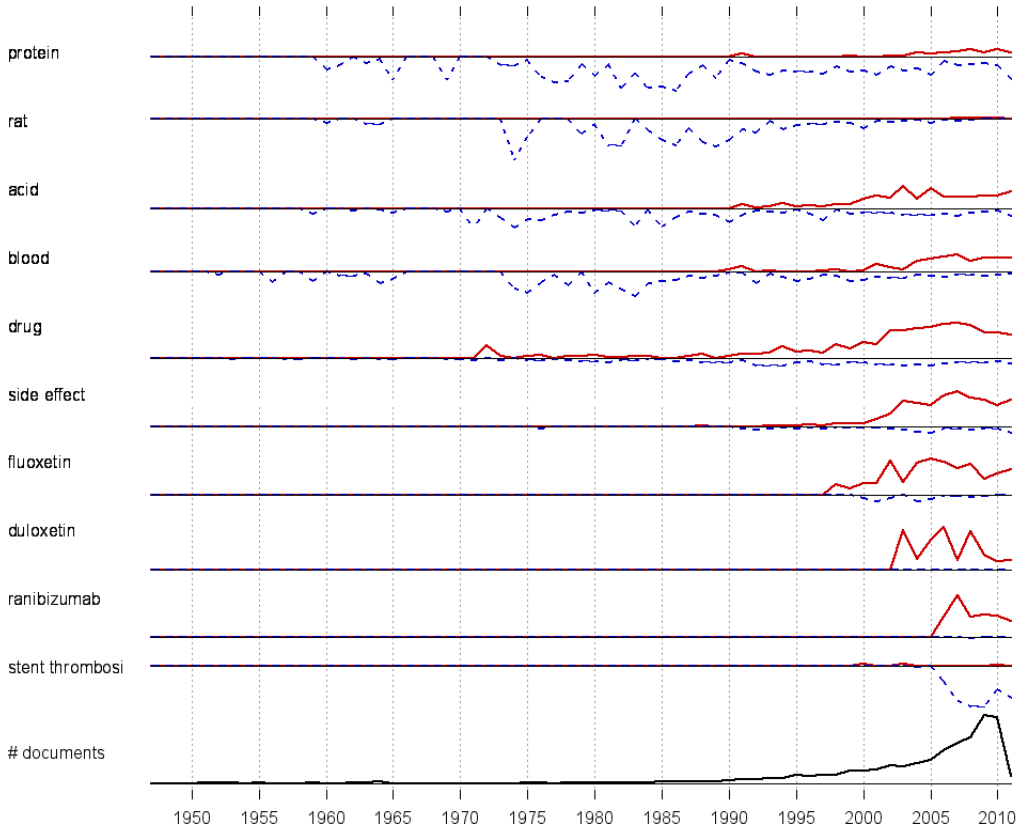

Figure 3.4: Occurrence rate of terms and phrases in relevant documents (red, solid) and irrelevant documents (blue, dashed) over time, relative to the number of documents available for the specific year.

3.2 System Design

The application described in this section was developed to perform experiments with various techniques and should be considered a research prototype. Its main focus was on being flexible and modular to enable the quick integration of new features and tests. Figure 3.5 shows the implemented modules of the application and the data flow between them.

In the task at hand, all input documents are encapsulated in the Ovid-XML file format. The input

format allows easy distinction of different parts of the documents like the title and the associated MeSH tags. Other input formats, like free text or HTML files, can easily be implemented on top of the existing parser.

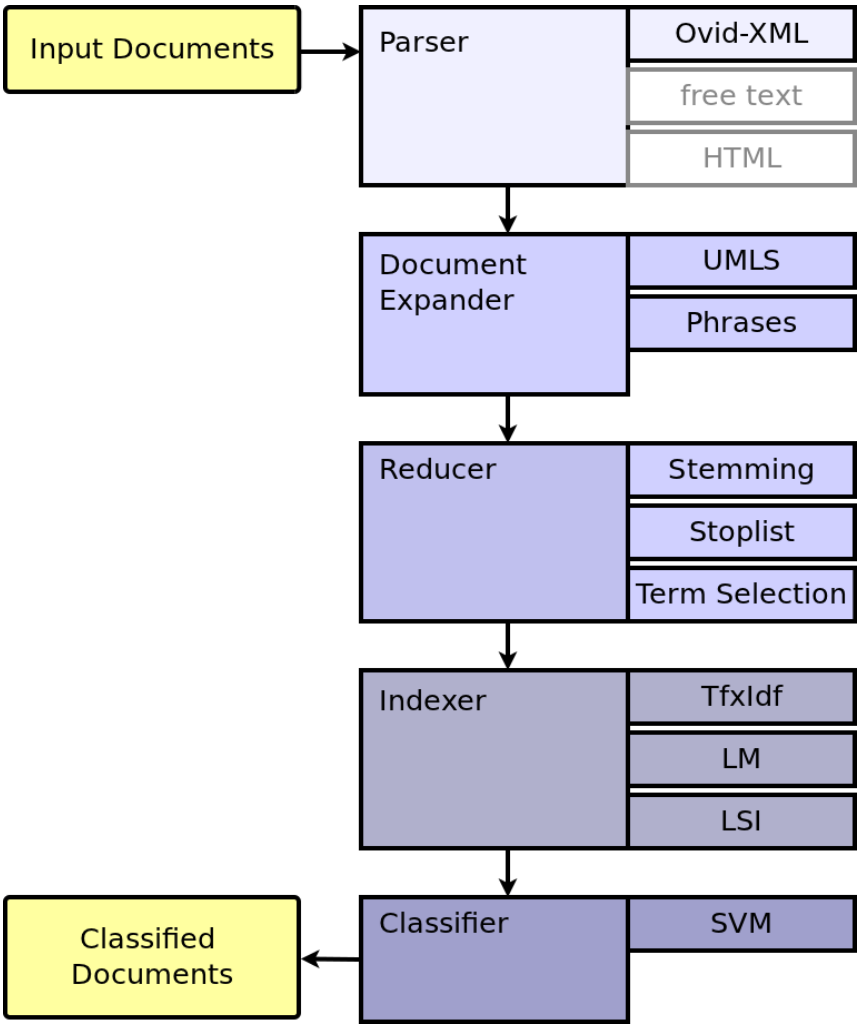

Figure 3.5: Document processing steps.

In the next processing step the documents are optionally augmented with background knowledge. For this thesis the UMLS Metathesaurus has been chosen as domain specific knowledge source. Another method to improve the document representation is to generate a list of statistically relevant phrases for the whole corpus and then add these to the respective documents.

Reduction of the feature vector is achieved through the stemming of all terms and afterward removing terms by a predefined stop list. After this process, a selection of the most representative features reduces the vector to an arbitrary size.

Three popular indexing mechanisms have been implemented in the application. Tfidf is one of the most common techniques. LM has been used in a wide variety of NLP applications and is the indexing method that produced the best results in the experiments carried out for this thesis, as shown in Section 4.2. LSI compresses the original feature space into linearly independent factors and has been shown to perform well for retrieval tasks [DDF$^+$90]. For this thesis an SVM-based classifier was chosen since literature suggests it is best suited for IR and TC tasks [Joa98, YL99].

3.3 Document Parsing

All input documents are in the Ovid-XML file format which allows the parser to distinguish between different elements of the original data-record. The four elements which were chosen for indexing (see Section 3.1.1) are stripped of any nested XML elements. In the next processing step the text is split into single words. All non alphabetic characters except the hyphen are considered to be a word boundary and are removed. If two words are connected by a hyphen they are merged to a single word, the hyphen is then also removed. All remaining words are folded to lower case. The internal representation of the document is now an ordered list of words. Additionally to the text representation a unique ID and the year of the publication is stored for every document. A separate file contains the identifiers of the relevant records previously manually assigned by the domain expert. All documents are assigned one of the two class labels *'relevant'* or *'irrelevant'*.

3.4 Statistical Phrases

The application optionally supports the use of statistical phrases as additional indexing features. A statistical phrase is a sequence of n words which appears at least k times in the corpus. The maximal length of the sequence is set to $n = 5$ words with similar values reported in literature [MG98]. The minimal occurrence of the sequence in the corpus k is necessary to reduce the number of possible phrases. k has been set to 5 which seems to remove almost all random sequences of words from the list of detected phrases. In contrast to some other implementations permutations of the sequence are not considered to be the same phrase. The generation of phrases is performed after stemming and stopword removal but before any term selection. All phrases are assigned a score depending on the occurrence rate and the distribution in the two classes. The score is calculated with the same formula as the CDR-score for the term selection, see Section 3.6.4. For any two sequences, $A = \{a_1, a_2, ..., a_n\}, B = \{b_1, b_2, ..., b_m\}, m > n$ with $a_1 = b_1, a_2 = b_2, ..., a_n = b_n$, only the one with a higher score is kept. This process removes approximately 80% of the identified phrases with more than two words, in favor of the shorter sequence. All identified phrases are then added to the documents according to their occurrence. Since the same feature selection function is used for single words and phrases, the process is indicative how informative phrases are in comparison to single terms. Table 3.2

shows the result before and after the selection of the 5000 highest scoring terms and phrases of the merged Medline+Embase dataset, graded by the CDR-score. Since the score ranks features according to their appearance rate in the collection and how class-discriminating they are, it can be considered a general performance measure for the features. Although the average score is lower for phrases, almost as many 2-word-phrases are kept for indexing as single words. 60.84% of all kept features are multi-word phrases which indicates their value for document representation.

Table 3.2: Phrases and single words selected for indexing from Medline+Embase dataset, CDR reduction to 5000 features.

	before CDR selection	after CDR selection	% kept	avg. CDR of kept
single word	35676	1858	5.2%	2131
2-word-phrase	29955	1810	6%	1556
3-word-phrase	5310	487	9.2%	1581
4-word-phrase	3149	250	7.9%	1369
5-word-phrase	3862	495	12.8%	1252

3.5 Document Expansion with UMLS

The UMLS Metathesaurus can be accessed online with a dedicated API[5] or directly from a local database. From a performance standpoint the online access seems to be a serious bottleneck, especially if many evaluation runs have to be carried out. To avoid this problem and to enable offline classification the application uses a local MySQL database. The tool MetamorphoSys[6] which is shipped with all UMLS releases allows the generation of a customized subset of the UMLS Knowledge Sources and MySql load scripts.

3.5.1 Concepts and Relationships

The core component of the UMLS Metathesaurus are the concepts which are stored in the database table MRCONSO. An excerpt of the concept *'Myocardial infarction'* is shown in Table 3.3. Every concept is identified by the unique identifier CUI which is guaranteed not to change over time and is not reused if the concept is removed from the Metathesaurus. Every UMLS concept consists of a set of atoms which contain a designation of the concept. Atoms have a term type (TTY) which indicates the purpose for the concept. Atoms of the type *'SY'* are *designated synonym*'s and *'IS'* are *obsolete synonym*'s. *'FN'* indicates that the atom contains a *Fully Specified Name* which means that the source lexicon considers it the most common designation

[5] `http://www.nlm.nih.gov/api/`

[6] `http://www.nlm.nih.gov/pubs/factsheets/umlsmetamorph.html`

Table 3.3: Atoms assigned to the concept *'Myocardial infarction'* (table MRCONSO).

CUI	LAT	SCUI	SAB	TTY	STR
C0027051	ENG	22298006	SNOMEDCT	PT	*Myocardial infarction*
C0027051	ENG	22298006	SNOMEDCT	IS	*Myocardial infarction, NOS*
C0027051	ENG	22298006	SNOMEDCT	SY	*Heart attack*
C0027051	ENG	194796000	SNOMEDCT	IS	*Attack - heart*
C0027051	ENG	22298006	SNOMEDCT	IS	*Heart attack, NOS*
C0027051	ENG	22298006	SNOMEDCT	SY	*Infarction of heart*
C0027051	ENG	22298006	SNOMEDCT	IS	*Infarction of heart, NOS*
C0027051	ENG	22298006	SNOMEDCT	SY	*Myocardial infarct*
C0027051	ENG	22298006	SNOMEDCT	SY	*Cardiac infarction*
C0027051	ENG	22298006	SNOMEDCT	IS	*Cardiac infarction, NOS*
C0027051	ENG	22298006	SNOMEDCT	SY	*MI - Myocardial infarction*
C0027051	ENG	22298006	SNOMEDCT	FN	*Myocardial infarction (disorder)*

used by clinicians to name the concept. The *Preferred Term*, described with the abbreviation *'PT'*, on the other hand, is the primary designation in UMLS. The addition of *'NOS'* (not otherwise specified) to some atoms originates from the source lexicon SNOMEDCT and means that they are based on a classification concept or an administrative definition [7] and are, in most cases, of limited value. However, they are still considered active by UMLS and are therefore treated like other atoms. Which atom is chosen as preferred term depends on the precedence ranking chosen by the user during the creation of a customized subset of the Metathesaurus. There are many other term types, which depend on the source lexicon [8]. For this application the type of an atom can be ignored, all atoms of a UMLS concept are considered synonymous designations for it. The identifier SCUI can be used to identify the concept in the knowledge source from where is originates, which is SNOMEDCT in this example. One atom of the UMLS concept also links to a different concept in SNOMEDCT than the other atoms. This is due to the mapping of 161 knowledge sources to one concept space in UMLS. Since the only knowledge source used for this prototype was SNOMEDCT, all concepts that overlap are merged. For instance would the concept C0027051 be merged with C0010072 *'Coronary artery thrombosis'*, C0155626 *'Acute myocardial infarction'* and C0340324 *'Silent myocardial infarction'*. Every atom also has a language tag assigned. This makes it easy to identify concepts in different languages which normally is a big problem for IR applications since the number of synonymous terms increases dramatically for every additional language. For this thesis, however, we focus only on the English language.

The Semantic Network introduces many different semantic types for the concepts in addition to

[7] `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1174894/`

[8] `http://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/release/abbreviations.html`

Table 3.4: Excerpt from the table MRREL, showing selected relationships of the concept *'Myocardial infarction'*.

CUI2	RELA	CUI1	Name of the related concept (from MRREL)
C0027051	has_clinical_course	C0750729	Courses
C0027051	occurs_before	C0152107	Postmyocardial infarction syndrome
C0027051	isa	C0878544	Cardiomyopathy
C0027051	isa	C1264235	Injury of anatomical site
C0027051	isa	C1274012	Ambiguous concept

over 170 different semantic relationships between them. The most common relationship is *'isa'* which indicates a generalization between the two concepts involved. Table 3.4 shows several typical relations from the concept *'Myocardial infarction'*. The concept C1274012 (*'Ambiguous concept'*) is one of several functional concepts which is related to a rather big subset of the whole Metathesaurus. These functional concepts cannot be used for document enrichment and have to be removed because they form a 2-step-connection between many not semantically related concepts. All removed concepts can be found in the appendix in Table C.2. The only other relationship besides *'isa'* used for the prototype application is *'same_as'* which points to a synonymous concept. Two concepts connected by this relation are treated like the overlapping SNOMEDCT concepts. Together they form a group of synonymous concepts. While all atoms are used to identify the concepts, only the one with the smallest CUI is used to represent the concept-group.

3.5.2 Identifying Concepts in Free Text

Identifying UMLS concepts in free text is the first step in enriching the documents with background knowledge. This task has previously been examined by Aronson which resulted in the MetaMap application [Aro01]. MetaMap takes free text as input data and outputs an ordered list of possible candidates for every concept identified. It also performs a candidate evaluation and assigns a score between 0 (no match) and 1000 (perfect match) to each mapping. The concept identification process uses a POS tagger to identify phrases, e.g. noun phrases, in the free text. Then the SPECIALIST Lexicon and an additional synonym database are used to generate variants as well as acronyms and abbreviations. Because of the complexity of the identification algorithm, MetaMap is rather slow in comparison to a simple string matching algorithm. Despite the sophisticated approach MetaMap misses a lot of concepts. One example is the incorrectly identified phrase *'for hypertension. patients'* where a sentence boundary is ignored and only the concept 'C0030705:Patient' is identified with a score of 861. The more important concept 'C0020538:Hypertension', however, is not detected. MetaMap also frequently identifies incorrect concepts. For instance is the concept 'C0010366:Genetic crossing over' identified in the

following sentence: *'Patients were randomized to split tablets or whole tablets for weeks, then crossed over to the other group for weeks.'*. These incorrectly identified concepts potentially add a lot of noise to the documents which poses a big problem for document expansion with MetaMap. Also considering the computational costs involved with MetaMap, a simple string matching approach which only identifies conceptual phrases that have a corresponding atom in UMLS seems desirable.

The simple algorithm implemented for this thesis works very well with the UMLS Metathesaurus since it already contains many lexical variants for each concept. A 1:1 string matching also decreases the risk of identifying incorrect concepts. The implemented algorithm always tries to match as many words as possible to a concept. Beginning with $n = 8$ words, the list of concepts is searched for any matching atom. If no concept is found the length n is decreased until it reaches zero. Also only atoms with at least 5 characters are considered for the matching since the UMLS Metathesaurus contains many abbreviations falsely matched to short words. The merged Medline+Embase dataset contains 2 154 544 words out of which the matching algorithm identified 635 511 UMLS concepts with an average length of 1.34 words. 13 853 distinct concepts were identified in the corpus. The algorithm also implements a resolution of ambiguous concepts as described in Section 3.5.3.

Table 3.5 shows a comparison of the simple identification approach with the MetaMap program. The simple pattern matching algorithm implemented for this thesis actually identified more concepts in the subset of 100 randomly picked documents than MetaMap. This can easily be explained by the fact that MetaMap often maps a whole sentence to only one concept even if several concepts are present. A manual survey of the multi-word concepts identified by the simple algorithm showed that of the 8896 concepts only 29 were incorrectly identified because of random word adjacencies. Of the 7009 concepts identified by MetaMap only 2996 were a 'perfect match' (score 1000). The average score which MetaMap assigned to an identified concept was 894. It must be assumed that many of the concepts that did not receive a perfect score were actually identified incorrectly and would potentially introduce additional noise to the documents if they were use for document expansion. Also the computational complexity of MetaMap is fairly high. It took MetaMap 203 times longer to identify the concepts than the simple string matching algorithm.

Table 3.5: Simple concept identification vs. MetaMap (100 randomly selected documents).

	Meta Map	Simple approach
Concepts identified	7009	8896
Distinct concepts	1837	1897
Required time for computation	6 min 46 sec	2 sec

3.5.3 Word Sense Disambiguation

Three possible ways of handling ambiguous terms are explored in this thesis conceptually based on [HSS03]. The probably most obvious solution is to ignore polysemous concepts and just choose the first concept, e.g. the one with the lowest identifier. Another simple approach is to add all polysemous concepts in the hope of achieving an effect similar to synonym expansion. The most sophisticated approach implemented is a context based WSD which utilizes background knowledge to distinguish between concepts depending on the vicinity of the concept in the text. Out of 635 511 concepts identified in the Medline+Embase dataset 12.2% are polysemous.

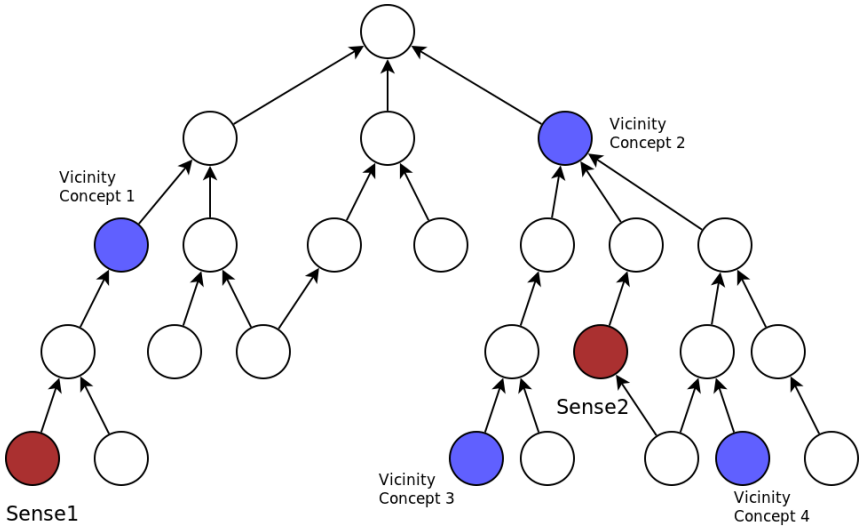

Figure 3.6: Example of WSD with an ontology (two possible senses and four vicinity concepts).

The context based WSD algorithm processes each document twice. In the first pass, all concepts are identified in the free text but only unambiguous concepts are considered for document expansion. The seconds pass performs the WSD on the ambiguous concepts using n previously identified concepts in the vicinity of the concept in question to distinguish between the senses. If possible, the vicinity consists of the closest $\frac{n}{2}$ concepts on either side of the ambiguous concept. Otherwise, the n nearest concepts are used. These vicinity concepts are then used to find the most probable sense for the ambiguous concept as depicted in Figure 3.6. First, the distance between the two senses and all vicinity concepts in the ontology is calculated. The distance between two concepts is defined as the number of edges on the shortest path between two vertices. The direction of the edges is ignored for the calculation of the distance. Edges between two concepts exist if there is either an *'isa'* or an *'inverse_isa'* relationship between the two concepts in UMLS. The sense with the lowest average distance to all vicinity concepts is chosen as the most probable sense in this context. In the example shown in Figure 3.6, the concept 'Sense2' is chosen by the algorithm since its average distance to the vicinity concepts is lower (3.75 for 'Sense2', 5.75 for 'Sense1'). While this example only employs four vicinity concepts, the eval-

uation of the WSD algorithm was performed using $n = 6$ concepts to represent the context of an ambiguous concept.

Because of the vast amount of concepts and relations in the ontology, the calculation of the distance between two concepts (breadth-first search) is a computationally very expensive process. To improve the performance of the application, the distance between the concepts which were identified in the dataset was precalculated and stored on disk.

3.5.4 Concept Generalization

Yet another form of document enrichment is the introduction of hypernyms to the documents. The UMLS Metathesaurus, together with the Semantic Network, is a rich source of hierarchical *'isa'* relationships allowing concept generalization of arbitrary depth. Prior research reported improved performance for document enrichment with hypernyms, up to a depth of 5 [HSS03, WHZC09]. For this application the effectiveness of concept generalization with depth of up to 2 is investigated. Since the Semantic Network is no strict hierarchy, a concept can have several less specialized concepts. The concept *'Aspirin'*, for instance, has 6 direct hypernyms and 12 for a depth of 2 (see Figure 3.7). The Metathesaurus contains several functional and navigational concepts like the depicted *'Duplicate Concept'*. While these concepts provide no additional information, they also connect semantically unrelated concepts which is problematic for concept generalization.

Even after removing the most interconnected functional and navigational concepts the amount of hypernyms with a depth of 2 and beyond is problematic. Table 3.6 shows the number of concepts added to the collection for every identified concept depending on the hypernym-depth.

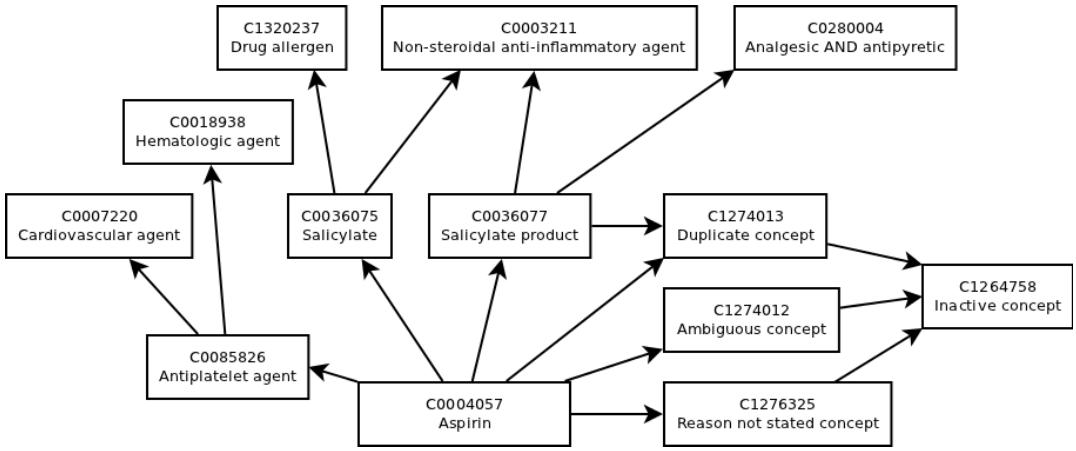

Figure 3.7: Hypernyms of the concept *'Aspirin'* from the UMLS Metathesaurus up to a depth of 2.

The highly connected functional concepts listed in C.2 were removed from the ontology prior to the document enrichment. For depths of 2 and more there is a high variance in the amount of concepts added per identified concept. This is due to the varying concept density in the UMLS Metathesaurus. There are also concepts which refer to a large amount of generalized concepts, e.g. *'Ataxia telangiectasia'* which has 30 hypernyms at depth 2. Adding too many near synonymous concepts increases the risk of introducing new data to a document which was not originally present and reduces the weight of the original concept. It is also reasonable to assume that the highly varying amount of concepts added for depth above 2 introduces an unwarranted prioritization of concepts in dense areas of the ontology above those in sparse regions.

Table 3.6: Hypernym concepts added to the merged Medline+Embase collection for hypernym-depths between 0 and 5.

Depth	# Concepts added	Avg. concepts [9]	Avg. deviation [10]	Max. concepts [11]
0	635 511	1	0	1
1	1 662 441	2.66	0.79	13
2	3 044 283	4.79	1.99	31
3	4 402 433	6.93	3.16	54
4	5 602 445	8.82	4.29	72
5	6 536 673	10.29	5.29	76

3.6 Dimensionality Reduction

Reduction of the feature set is an important step before indexing. Three different reduction techniques are described in this section. Stemming and stop word removal are common techniques and are used in most state of the art systems. Following is a survey of feature selection functions found in literature and a newly developed function, the class discrimination ratio (CDR).

3.6.1 Snowball Stemmer

Stemming is performed with the Java implementation of the popular snowball stemmer[12] written by Martin Porter. Snowball defines itself as *'Small string processing language designed for creating stemming algorithms for use in Information Retrieval'*. While only the English stemmer is used for this application many other languages are supported.

[9] Average concepts added per identified concept.
[10] Average absolute deviation of concepts added per identified concept.
[11] Maximal number of concepts added for an identified concept.
[12] `http://snowball.tartarus.org/`

3.6.2 SMART Stoplist

Experiments performed with several different stop word lists showed that removing terms from a predefined list can improve document classification. The stop word list used in the prototype application origins from the SMART system [Sal71]. The list is available online[13] and is also included in the appendix C.1. Apostrophes and the following words have been removed from the stop word list because of their importance for the dataset: *off*, *use*, *used*, *uses* and *using*. The final list consists of 566 words and single letters.

3.6.3 Feature Selection

Feature selection is an important step to reduce overfitting and improve processing performance. Another positive effect is the reduction of noisy features which can greatly improve classification. The previously defined features are single words, also called BOW, phrases and UMLS concepts which all are reduced by the same selection function. Even after stemming and stop word removal there are over 50 000 features left for documents expanded with statistical phrases. Without any reduction, the resulting index would consist of over $4 \cdot 10^8$ double values and require 3.2GB of memory. The selection of appropriate features to represent the entire corpus, and of course all yet unknown documents, is based on information about the distribution of the features in the training corpus. Another relevant aspect to consider is the distribution of the features in the classes. If one feature is almost exclusively present in one class it has a high *class discriminating* value. Following is a survey of various selection functions which have been implemented and tested for this thesis.

All selection functions return a $score_k$ for each distinct feature t_k in the corpus. The features are then ordered according to their score which indicates the expected indexing quality. An upper and lower-threshold determines the features selected for indexing. Except for DF-thresholding, all selection functions choose the top-n features from the ranked feature list. See Section 4.1.2 for a performance evaluation and comparison of the selection functions. The experimental results comply with those reported in literature [Fab02]. The only exception is *odds ratio* which performed much worse in the empirical evaluation performed for this thesis.

Document Frequency Thresholding

DF-thresholding was originally implemented in the first version of the prototype application. The document frequency $df(t_k)$ is the number of documents in the corpus which contain a term t_k. All features than have a $score_k$ below the minimal threshold $minDf$ or above the maximal threshold $maxDf$ are removed from the index. The principle behind the algorithm is that all terms which are present in almost all documents are assumed to be stop words. Terms which

[13] `http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop`

occur only in a small minority of the training corpus, on the other hand, are not influential for the global representation and are often noisy [YP97].

$$score_k = df(t_k) \tag{3.1}$$

DIA Association Factor

The DIA association factor orders the features according to their maximal occurrence probability in any of the classes. $p(t_k, c_i)$ is the probability that a document of class c_i contains the term t_k. The probabilities for all features are approximated from their occurrence rate in the training corpus [CMF01, Fab02].

$$score_k = \max_{i \in \{1,2\}} (p(t_k, c_i)) \tag{3.2}$$

Term Frequency Thresholding

Term frequency $tf(t_k)$ is the number of occurrences of the term or feature in the corpus. While the selection of features just according to their appearance rate was not expected to perform very well, it delivers results comparable to those achieved with selection by DIA-score (see Section 4.1.2).

$$score_k = tf(t_k) \tag{3.3}$$

Information Gain

The information gain (IG) measures the amount of information gained by knowing about the presence or absence of a term t_k in a document. The probability that a term t_k does not occur in a document is expressed as $p(\bar{t}_k)$. $p(c_i)$ is the probability for a document to be of class c_i. Although this thesis is focused on binary classification a generalized formula is provided [YP97, CMF01, Fab02].

$$score_k = \sum_{c \in \{c1,c2\}} \sum_{t \in \{t_k, \bar{t_k}\}} p(t,c) \cdot \log \frac{p(t,c)}{p(t) \cdot p(c)} \tag{3.4}$$

Mutual Information

Mutual Information is a concept developed in information theory that measures the dependence of two random variables. In this case the dependence between the probability of occurrence for a term t_k and for a class c_i is calculated. A weakness of the MI-score is that it is dominated by marginal probabilities. This results in problems if the corpus contains terms with widely differing frequencies [YP97, RS02, Fab02].

$$score_k = \max_{i \in \{1,2\}} \log \frac{p(t_k, c_i)}{p(t_k) \cdot p(c_i)} \tag{3.5}$$

Chi-square

Reduction by χ^2 is inspired by the probability distribution χ^2 with one degree of freedom. It measures the lack of independence between the term t_k and the class c_i. $|D|$ is the total number of documents in the corpus. According to several publications, the χ^2 statistic is known to be not reliable for features with a low occurrence rate [YP97, CMF01, GFS00, Fab02].

$$score_k = \max_{i \in \{1,2\}} \frac{|D| \cdot (p(t_k, c_i) \cdot p(\bar{t_k}, \bar{c_i}) - p(t_k, \bar{c_i}) \cdot p(\bar{t_k}, c_i))^2}{p(t_k) \cdot p(\bar{t_k}) \cdot p(c_i) \cdot p(\bar{c_i})} \tag{3.6}$$

NGL Coefficient

The NGL coefficient, also often referred to as correlation coefficient, is the square root of the χ^2 statistic measure. This change results from the observation that the original χ^2 does not emphasize the positive correlation between a feature t_k and a class c_i more than the negative correlation. Results from literature indicate that NGL performs better than χ^2 [RS02, GFS00, Fab02]. The experiments carried out for this thesis show exactly the same performance for both functions since a binary classification is performed.

$$score_k = \max_{i \in \{1,2\}} \frac{\sqrt{|D|} \cdot (p(t_k, c_i) \cdot p(\bar{t_k}, \bar{c_i}) - p(t_k, \bar{c_i}) \cdot p(\bar{t_k}, c_i))}{\sqrt{p(t_k) \cdot p(\bar{t_k}) \cdot p(c_i) \cdot p(\bar{c_i})}} \tag{3.7}$$

Relevancy Score

The relevancy score [WPW95, Fab02] judges terms according to how good they predict the class membership by themselves. The variable d has to be set according to the corpus. An empiric

evaluation on the Medline documents shows that $d = 0.006$ performs best for the test collection at hand.

$$score_k = \max_{i \in \{1,2\}} \log \frac{p(t_k, c_i) + d}{p(\bar{t_k}, c_i) + d} \tag{3.8}$$

Odds Ratio

Odds ratio was designed for binary classification with the goal of making good predictions for one class. Some features have an occurrence probability of 0 for a class c_i in the training corpus which is not expected in reality. In these cases a small probability d is introduced. Empirical tests showed best performance for $d = 0.0004$ on the Medline dataset. [CMF01, RS02, Fab02]

$$score_k = \begin{cases} \frac{d \cdot (1 - p(t_k, c_2))}{(1-d) \cdot p(t_k, c_2)} + \frac{p(t_k, c_2) \cdot (1-d)}{(1 - p(t_k, c_2)) \cdot d} & \text{if } p(t_k, c_1) = 0, \\ \frac{p(t_k, c_1) \cdot (1-d)}{(1 - p(t_k, c_1)) \cdot d} + \frac{d \cdot (1 - p(t_k, c_1))}{(1-d) \cdot p(t_k, c_1)} & \text{if } p(t_k, c_2) = 0, \\ \sum_{i \in \{1,2\}} \frac{p(t_k, c_i) \cdot (1 - p(t_k, \bar{c_i}))}{(1 - p(t_k, c_i)) \cdot p(t_k, \bar{c_i})} & \text{else.} \end{cases} \tag{3.9}$$

GSS coefficient

The GSS coefficient is a further improvement on NGL by removing the following three factors from the formula which have undesired effects on the score. Since the factor $\sqrt{|D|}$ is the same for all features it can be removed. $\sqrt{p(t_k) \cdot p(\bar{t_k})}$ in the denominator improves the score for low-frequency features and thus should be removed. $\sqrt{p(c_i) \cdot p(\bar{c_i})}$ on the other hand emphasizes low frequency classes which is also considered to have a negative effect on global performance [GFS00, Fab02].

$$score_k = \max_{i \in \{1,2\}} p(t_k, c_i) \cdot p(\bar{t_k}, \bar{c_i}) - p(t_k, \bar{ci}) \cdot p(\bar{t_k}, c_i) \tag{3.10}$$

3.6.4 Class Discrimination Ratio

The CDR is a new approach at feature selection for binary text classification. It follows the intuition that a feature is good for document representation if it discriminates well between the two classes. The CDR-score consists of two factors. The first is simply the frequency of the term in the corpus $tf(t_k)$. The second is the ratio of the occurrences of the term t_k in both classes, normalized to the occurrence rate of the class in the corpus. The square root decreases

the influence of the second factor which is necessary to keep both in balance. The motivation is to assign a score to a feature depending on how well it separates the classes and how often it actually occurs in the corpus. It is quite obvious that both factors are important. A term that only very rarely occurs in a single class makes an equally bad indexing features as one that occurs often but is distributed evenly over the two classes. The normalization is necessary if there is an unjustified class-skew in the training collection. The Medline and Embase collections, for instance, contain more samples which are part of the relevant class. However, in a real world scenario one can expect much less relevant samples than irrelevant.

Many terms in the collection actually only occur in one of the two classes which would result in an infinite term frequency ratio. To avoid this problem, the normalized ratio is set to a constant value d. A value of 5000 for d has shown the best results for the Medline dataset. As demonstrated in the empirical evaluation, see Section 4.1.2, the CDR-score outperforms the other feature selection functions for reasonable sized feature vectors.

$$score_k = \begin{cases} tf(t_k) \cdot \sqrt{d} & \text{if } tf(t_k, c_1) = 0 \text{ or } tf(t_k, c_2) = 0, \\ tf(t_k) \cdot \max_{i \in \{1,2\}} \sqrt{\frac{tf(t_k, c_i) \cdot p(\bar{c_i})}{tf(t_k, \bar{c_i}) \cdot p(c_i)}} & \text{else.} \end{cases} \tag{3.11}$$

3.7 Indexing

After the selection of a set of n appropriate features, an index of all documents is created. The constructed index is a n by $|D|$ matrix in the vector space model. The weight of a feature for a document depends on its relevance for the document as well as the feature weight used. Weights for tfidf and LM are calculated straight forward from corpus and term statistics. The implementation of LM used for the prototype application uses the formulas cited in Section 2.4 with the following two modifications. In the experiments performed with the LM indexer on the Medline dataset, the first factor of the Formula 2.3 significantly increased the risk (decreased the risk-factor $\hat{R}_{t_k,d}$) for popular terms. The result is that the estimate for $\hat{p}$ for terms which occur often in the same documents, is calculated almost exclusively from the single document instead of the whole collection. Since this effect showed a negative impact on performance on the Medline training set, $\frac{1}{1+\bar{f}_{t_k}}$ was removed from the formula 2.3. In the original work on indexing text with LM, the assumption was made that the estimated probability $\hat{p}$ for a term is greater than zero even if it does not occur in the training document [PC98]. While the assumption by itself seems reasonable, empirical evaluations on the Medline dataset showed that $\hat{p}(t_k, d) = 0$ for $tf(t_k, d) = 0$ achieves better results for the classification task examined in this thesis. The two formulas were modified accordingly and are depicted in Formula 3.12 and Formula 3.13.

$$\hat{R}_{t_k,d} = \left(\frac{\bar{f}_{t_k}}{1 + \bar{f}_{t_k}} \right)^{tf_{t_k,d}} \tag{3.12}$$

$$\hat{p}(t_k|M_d) = \begin{cases} p_{ml}(t_k,d)^{1-\hat{R}_{t_k,d}} \cdot p_{avg}(t_k)^{\hat{R}_{t_k,d}} & \text{if } tf_{t_k,d} > 0, \\ 0 & \text{else.} \end{cases} \tag{3.13}$$

LSI depends on singular value decomposition (SVD) which is a factorization of the original index matrix. The SVD is calculated with the open source library colt[14] which is developed by CERN. Singular value decomposition is a computationally very expensive operation with $O(m \cdot n^2)$ floating point operations, m being the number of vectors $|D|$ and n the number of features. Since the cost increases quadratically with the size of the feature vector, a reduction of features has to be performed before the SVD. The result of the factorization is a new set of features of which only a subset of k features, with $k < n$, is chosen for indexing. The final index is converted to the attribute-relation file format (ARFF)[15] which is used by WEKA to store instance data. For every record the unique ID and the assigned class label is added.

3.8 Classification with WEKA

The Waikato Environment for Knowledge Analysis (Weka)[16] is a popular toolkit for machine learning which is developed and maintained by the university of waikato. Weka offers an easy to use graphical interface as well as the option to run all stages of the processing directly from the command line. The prototype application integrates Weka directly in the form of a jar-library. Many classifiers are supported by the toolkit. However, since SVM classifiers seem to perform best for text classification [Joa98, YL99, Fab02], only the SMO classifier is used by the prototype application. Sequential Minimal Optimization (SMO) is a resource efficient implementation of an SVM which is integrated into Weka.

The previously created index file can be loaded by Weka as a set of data instances. The unique ID attribute can be removed before the training with a filter that is added to the SMO classifier object. For k-fold cross validation, the data is randomized with a static seed to ensure reproducible results. All runs carried out for this thesis were performed with 10 folds and a static seed of 1. In the next step, the data is split into a training and a test set for every one of the k-folds. Optionally the prototype application trains a model on the complete dataset provided and saves the model to disk as a serialized Java object together with the indexing features and activated options. The prototype application can also perform a classification run on new documents with a previously saved model. In this case all previously set options necessary for reduction and indexing are loaded from the saved model together with the SMO classifier.

[14] `http://acs.lbl.gov/software/colt/`
[15] `http://www.cs.waikato.ac.nz/ml/weka/arff.html`
[16] `http://www.cs.waikato.ac.nz/~ml/weka/`

3.9 Parameter Optimization

Several of the internal parameters have to be tuned by empirical evaluation of the effectiveness. This optimization process is performed using the Medline dataset only to decrease the effect of overfitting parameters to the training set.

3.9.1 Thresholds for DF Reduction

The effectiveness of reduction by DF-thresholding depends on the choice of the two thresholds *minDf* and *maxDf*. Table 3.7 shows the results of a grid search for the two parameters performed on the Medline dataset. The best performance (F1 measure) was achieved with $minDf = 0.007$ and $maxDf = 0.6$ which reduced the document index to 1746 terms.

3.9.2 SVM Kernel Function

A support vector machine (SVM) can either perform linear classification or non-linear by the use of different kernels. The most prominent kernels are radial basis function (RBF) and polynomial kernels [CVBM02, HCL03]. The choice of the kernel depends on specific classification problem. To find the optimal kernel for the classification at hand, an empirical evaluation of all kernels on the collection has to be performed.

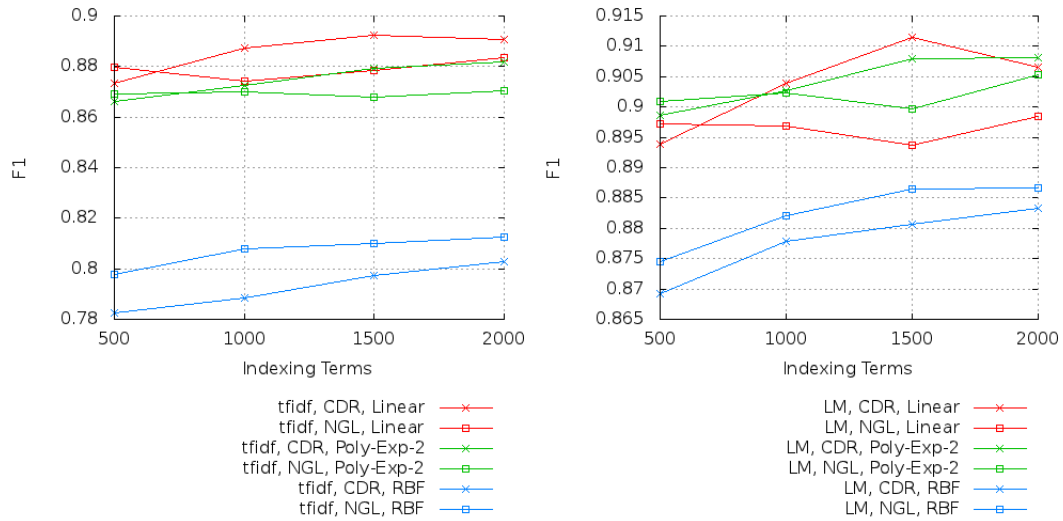

Figure 3.8: RBF, linear and quadratic kernel functions with tfidf (left) and LM (right) indexing.

Table 3.7: Grid search for optimal DF-thresholds with minDf between 0.003 and 0.2 and maxDf between 0.4 and 0.8.

maxDf	0.4		0.6		0.8	
	precision	F1	precision	F1	precision	F1
minDf	recall	#features	recall	#features	recall	#features
0.003	0.86228	0.85708	0.86750	0.86710	0.86560	0.86799
	0.85194	2957	0.86670	2961	0.87039	2963
0.004	0.86239	0.85900	0.86716	0.86716	0.86615	0.86734
	0.85563	2500	0.86716	2504	0.86854	2506
0.005	0.87006	0.86114	0.87049	0.86928	0.86863	0.87043
	0.85240	2186	0.86808	2190	0.87223	2192
0.006	0.86971	0.86120	0.87279	0.86997	0.87125	0.87105
	0.85286	1891	0.86716	1895	0.87085	1897
0.007	0.87080	0.86596	**0.88011**	**0.87685**	0.87362	0.87362
	0.86116	1742	0.87362	1746	0.87362	1748
0.008	0.87236	0.86485	0.87697	0.87413	0.87489	0.87609
	0.85747	1596	0.87131	1600	**0.87731**	1602
0.009	0.86863	0.86278	0.87115	0.87215	0.87070	0.87330
	0.85701	1453	0.87315	1457	0.87592	1459
0.01	0.86934	0.86430	0.87275	0.87295	0.87155	0.87235
	0.85932	1351	0.87315	1355	0.87315	1357
0.012	0.86803	0.86481	0.86965	0.87025	0.87339	0.87419
	0.86162	1178	0.87085	1182	0.87500	1184
0.014	0.86861	0.86113	0.86948	0.86647	0.86987	0.86967
	0.85378	1057	0.86347	1061	0.86946	1063
0.016	0.86854	0.86087	0.86700	0.86800	0.86432	0.86849
	0.85332	937	0.86900	941	0.87269	943
0.018	0.86698	0.86033	0.86928	0.86868	0.86594	0.87090
	0.85378	852	0.86808	856	0.87592	858
0.02	0.86465	0.86105	0.87309	0.87127	0.87281	0.87321
	0.85747	771	0.86946	775	0.87362	777

Figure 3.8 shows the results of classifications performed with RBF, linear and quadratic kernels. As expected, linear kernel clearly performs best for tfidf indexing and also produces the best result for the LM indexer with 1500 terms. It is generally considered that linear kernels are best for text classification since the dimensionality is already high enough and use of polynomial kernels does not improve the ability of the SVM to separate the data. It should also be mentioned that the linear kernel is by far the most resource efficient. Computation time for a RBF kernel is up to 10 times higher than for the linear kernel while classification with the quadratic kernel takes up to 5 times longer. With this in mind, the linear kernel seems the most appropriate for this application.

3.9.3 SVM Parameters

There are two parameters C and ϵ that affect the classification performance of a SVM. The round-off error ϵ has a default value of 10^{-12} in WEKA. Changing this value showed only a negative impact on classification performance. The penalty factor C however has a significant effect on performance. The best value for C depends on the size of the feature vector and on the reduction function. To find the best value for the parameter, a grid search with k-fold cross validation should be performed [HCL03]. Figure 3.9 shows the results of a grid search with the CDR and NGL reduction functions and a tfidf indexer. It is quite unexpected that the reduction function has such a big influence on the best value for the penalty factor. However, while the effectiveness (F1 measure) for NGL is achieved with 2000 terms and $C = 0.8$, tfidf with reduction by CDR performs best with $C = 4$ and only 1250 terms.

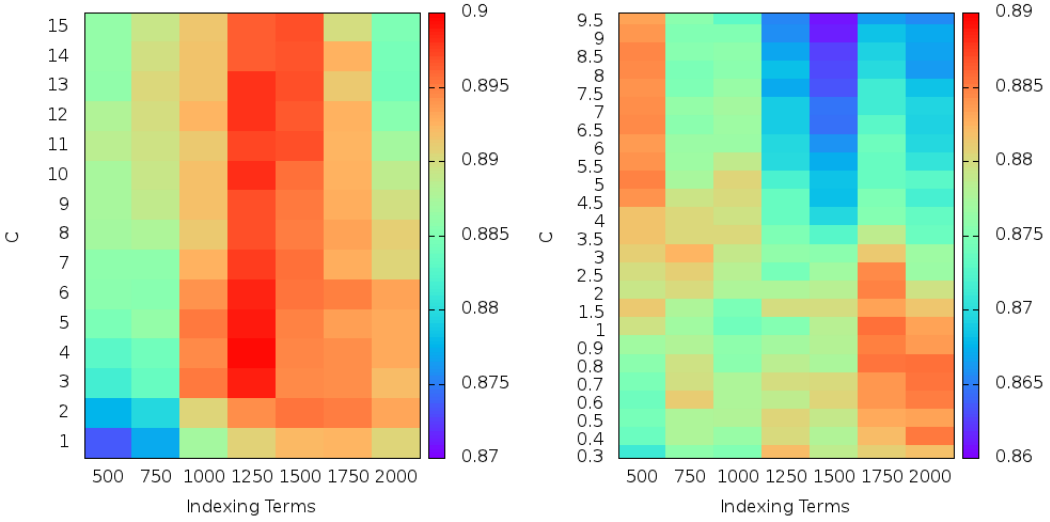

Figure 3.9: Grid search for optimal C with tfidf indexing and CDR (left) and NGL (right) reduction (F1 measure).

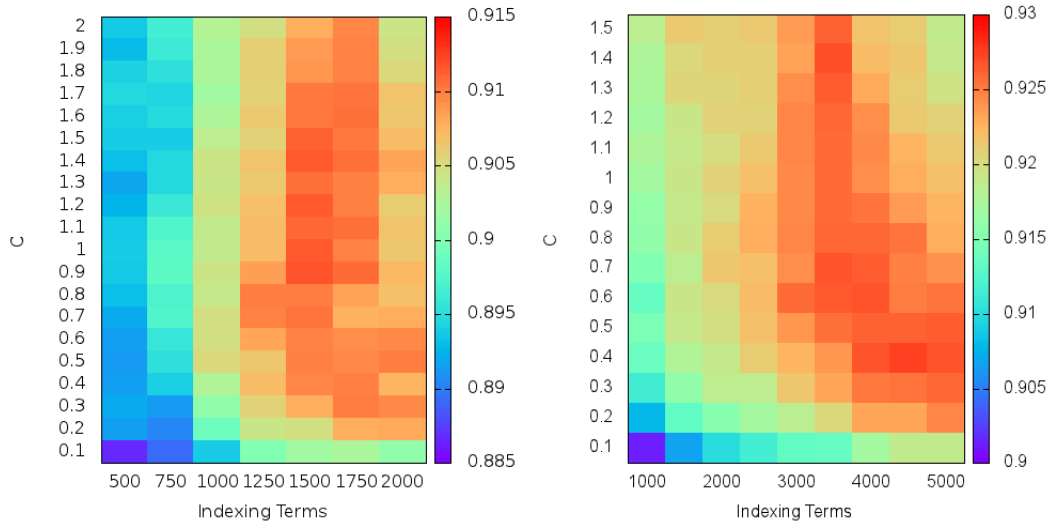

Figure 3.10: Grid search for optimal C with LM indexing and CDR reduction on Medline (left) and combined dataset (right) using the F1 measure.

In addition to the reduction function, the choice of the indexing method also has a big impact on the optimal parameter selection. Figure 3.10 shows a classifier with LM indexing and CDR reduction on the Medline and the combined (Medline+Embase) datasets. In both experiments the best results were achieved for a penalty factor below 1. The tendency towards lower values for C for higher feature vector sizes can be observed. This is because the increase in dimensionality also increases the problem of separating the instances. Therefore the penalty for non-separable points has to be decreased to prevent overfitting the SVM to the training data. The best choice for C is 0.6 for feature vector sizes between 3000 and 4000 and 0.4 above.

3.10 Summary

The merged training collection, consisting of records from the literature databases Medline and Embase, contains 8118 samples of which 4236 are relevant to the topic of off-label drug use. Many of the records contain no abstract and are only represented by the title, MeSH tags and the field, *'Name of Substance'*. 1525 of the instances contain below 50 terms before any reduction occurs and are therefore underrepresented in the index. The majority of the instances of the training set are recently published articles. About 80% of the training samples have been published after 1998. The validation set consist of 500 records from 2011 of which 79 are labeled as relevant.

The classification system is split into several processing steps which are carried out by basically independent modules. After parsing, the documents are augmented with statistical phrases or

UMLS concepts or both. The use of the UMLS as source of background knowledge allows the detection of synonyms, the resolution of ambiguous concepts, as well as the augmentation of the text with more generalized concepts (hypernyms). The WSD algorithm, described in this chapter, uses the distance between concepts in the ontology as a measure of semantic similarity to predict the most likely intended sense for ambiguous concepts. Hypernyms, on the other hand, can directly be inferred from the *'isa'* relationships in the Semantic Network. While the program MetaMap uses a POS tagger and the SPECIALIST Lexicon to identify concepts in free text, the application described in this thesis relies on a simple string matching approach. A direct comparison between the two techniques shows that the simple approach identifies more concepts and is much faster. Also many of the concepts identified by MetaMap are not considered a 'perfect match' and would introduce noise when used for document expansion.

Aside from stemming and removing predefined stopwords, several common feature selection functions have been presented in this chapter. The results of an evaluation of these functions and the novel CDR-score are compared in the next chapter, see Section 4.1.2.

A grid search, employed to optimize the parameters for DF-thresholding, revealed $minDf = 0.007$ and $maxDf = 0.6$ as optimal boundaries. Experiments also showed that the classification problem investigated is best handled by a SVM with a linear kernel. The optimal value for the penalty factor C seems to depend on the size of the used feature vector. The results of a grid search, depicted in Section 3.9.3, will be used to adjust the classifier in the final evaluation.

CHAPTER 4

Results and Analysis

After the selection and implementation of the most promising techniques, an empirical evaluation is necessary to assess the expected performance of the classifier. Initial experiments are performed on the Medline dataset only. The merged Medline and Embase datasets are used for a final cross validation which shows the effectiveness of statistical phrases and the UMLS Metathesaurus for document expansion. The best performing classifiers are also trained with the combined dataset and evaluated against the validation set in Section 4.8. The baseline classifier was developed prior to this thesis. It uses reduction by DF-thresholding with all terms below a document frequency of 0.7% and above 60% being removed. The index is built using tfidf weighting. This chapter will compare several more sophisticated reduction functions, indexing methods and document expansion techniques to the results achieved with the baseline classifier.

4.1 Feature Space Reduction

Choosing the right terms for indexing is regarded an important first step in any IR task as shown in Section 3.6. Beside the obvious improvement in performance achieved by reducing the over 47 000 distinct terms of the Medline+Embase dataset to below 5000 terms, dimensionality reduction also decreases the effect of overfitting the classifier to the training data. This section shows the results of experiments performed on the Medline dataset with and without stemming and removing terms from a predefined stop word list. Also the selection functions introduced in Section 3.6.3 are evaluated and compared to the novel CDR-score.

4.1.1 Reduction by Stemming and Stop Word List

An evaluation of the impact of stemming and the removal of stop words from a predefined list is presented in Figure 4.1. While the results are probably not as definite as one would expect, stemming and the removal of predefined stop words improves the classification performance in most cases for reasonably sized feature vectors. Since the initial experiments indicated a positive impact on effectiveness, stemming and stop word removal have been activated in all further experiments.

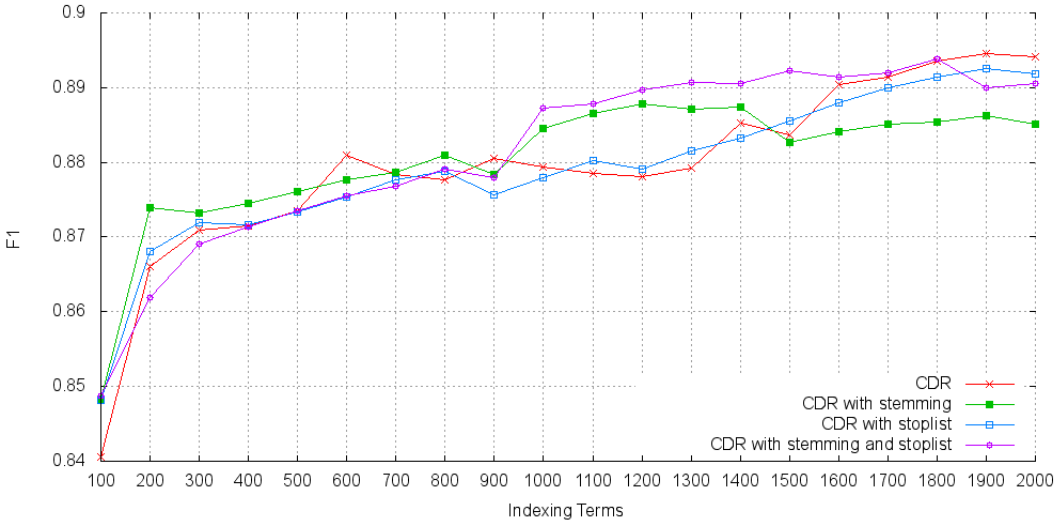

Figure 4.1: Impact of stemming and stopword removal on classification (CDR, tfidf).

It seems very odd that stemming and stopword removal by themselves produce worse results but improve the overall performance when combined. While the idea behind stemming is to improve the stochastic dependence between terms and reduce the feature vector size, it has sometimes been reported to have a negative impact on performance [Fab02]. Table 4.1 shows the TF for three stems from the Medline dataset. In this example, the individual words have quite a different distribution in the two classes and will discriminate better than their stem. Considering this, it could prove beneficial to perform a selective stemming which first calculates the CDR-score for each term and only performs stemming for the variants which distribute similar in collection.

Table 4.2 shows several words from the SMART stoplist which have quite a different distribution in the two classes and appear to be important for classification. The most dramatic example is the word *'me'* which seems to be very indicative for non-relevancy. In fact *'me'* is the abbreviation for the MeSH term *'metabolism'* which explains its importance. In general, it should be assumed that stop words contain no information and any relevancy indicated by a feature score is just coincidental.

Table 4.1: Problematic terms for stemming (Medline dataset).

Term	Stem	TF in irrelevant	TF in relevant	Ratio[1]
cell	cell	1109	209	3.62
cells	cell	1462	153	6.52
	cell	2571	362	**4.84**
factor	factor	367	1176	0.21
factors	factor	855	501	1.16
	factor	1222	1677	**0.5**
use	use	3068	6542	0.32
used	use	1063	1272	0.57
uses	use	52	392	0.09
using	use	1493	384	2.65
	use	5676	8590	**0.45**

Table 4.2: Stop words that appear to be useful (Medline dataset).

Term	TF in irrelevant	TF in relevant	Ratio[1]
me	1879	153	**8.38**
its	474	648	0.5
has	904	1114	0.55
been	920	1042	0.60
between	1002	381	1.79

4.1.2 Reduction by Term Selection

Of the evaluated reduction functions, mutual information (MI) performed worst followed by odds ratio (OR). Both functions favor terms which separate the classes best without adequately considering their appearance rate. Because the collection contains many terms with a high class to class ratio that occur only rarely, they are not suitable for this task. Peak accuracy of mutual information was only 66.8% with 2000 terms. Because of their obviously poor performance for this application, odds ratio and mutual information will be excluded from further analysis.

Figure 4.2 depicts the results from a 10-fold cross validation on the Medline collection with a linear SVM and a default soft margin parameter *C=1.0*. Also stemming and stopword removal were switched on. Even a quite aggressive feature reduction by a factor of over 100, with only

[1]The ratio is normalized to the number of documents per class. The Medline training set contains 3179 irrelevant and 2168 relevant documents. $Ratio = TFirrel * \frac{2168}{2168+3179} / TFrel * \frac{3179}{2168+3179}$. A ratio of 1 shows a term to be evenly distributed between the two classes.

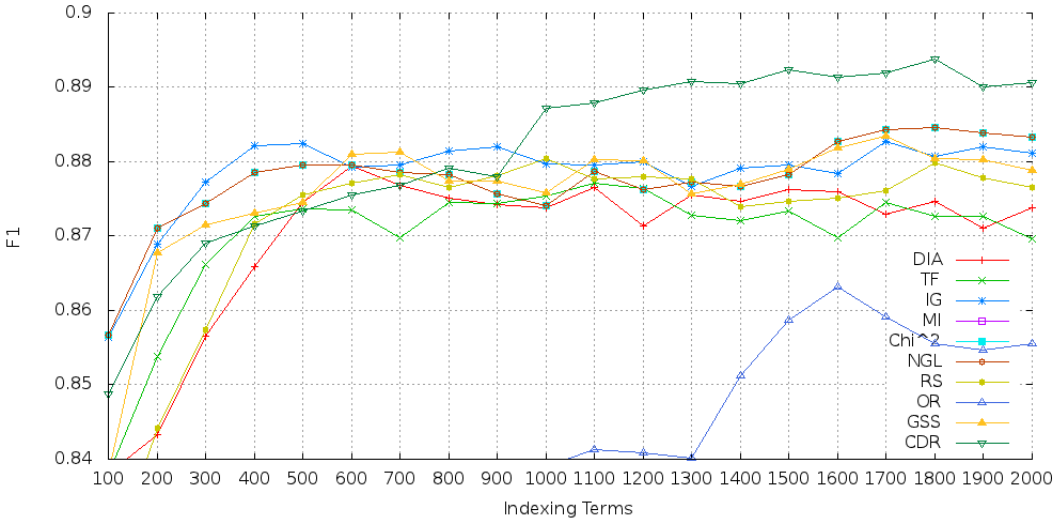

Figure 4.2: Comparison of different reduction functions (with tfidf indexing).

300 features remaining, brings about but a small loss in classification performance. However, it's reasonable to assume that this fact will change with the introduction of previously unseen data that is not present in the training set. While the other reduction functions do not improve the performance of the classification by allowing more than 500 distinct indexing terms, CDR shows up to 1% gain with higher feature vector sizes. The next best reduction functions are Chi^2 and NGL which overlap for all tested feature vector sizes. This is because NGL is an extension of Chi^2 but the difference is only relevant for non-binary classification.

4.2 Indexing

Figure 4.3 shows an evaluation of the implemented indexing techniques tfidf, language modeling (LM) and latent semantic indexing (LSI) on the Medline dataset. LSI seems a promising technique and has previously been used with some success. Its main advantage is the high compression of the feature space which is achieved by performing a singular value decomposition (SVD) on the feature-document matrix. The computational cost of this calculation is $O(m \cdot n^2)$ which makes it not suitable to compress very big feature vectors. In the empirical evaluation, the feature space was reduced from 27217 (after stemming and stopword removal) to only 3000 terms by CDR before compressing it further by the LSI process. Even after this reduction to only 11% of the original terms, the SVD took over an hour on a state of the art computer [2]. In the empirical evaluation performed on the Medline dataset LM clearly performs best followed by tfidf which performs about 2% worse. LSI also delivers poor performance with very low feature

[2] Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz, 16GB Ram

vector sizes. Considering the difference in performance to tfidf and especially LM indexing, it seems not practicable for this classification task. These results also indicate that there is no gain to be expected from using more than 1500 terms on the Medline dataset without any document expansion.

Figure 4.3: Comparison of tfidf, LM and LSI indexers.

4.3 Expansion with Statistical Phrases

The experiments with statistical phrases, shown in Figure 4.4, indicate that multi-word phrases can contribute a lot to the document representation. Many concepts, which are represented by more than one word, are lost if only single terms are used for indexing. For most multi-word concepts, the individual terms which form the concept have quite a different meaning by themselves. Replacing two words that belong conceptually together also improves the stochastic characteristic of the individual terms. For instance is the phrase *'use unlabel'* very indicative for irrelevant documents. However, *'offlabel use'* is much more frequent in relevant documents. By replacing the two individual words with a phrase, the term *'use'* becomes a less noisy feature.

Experiments performed with documents augmented by statistical phrases show a rather big increase in precision with no or only a small loss in recall. Table 4.3 lists the results from the classification run with 5000 features. An analysis of the top ranked phrases, which were used for indexing, shows that they separate the two classes very well. For instance are the stemmed phrases *'offlabel drug'*, *'dermatolog agent'* and *'botulinum toxin'* quite important for relevant documents while *'use unlabel'*, *'drugelut stent'* and *'stent thrombosi'* indicate an irrelevant document. While it is obvious that statistical phrases improve document representation, the chance

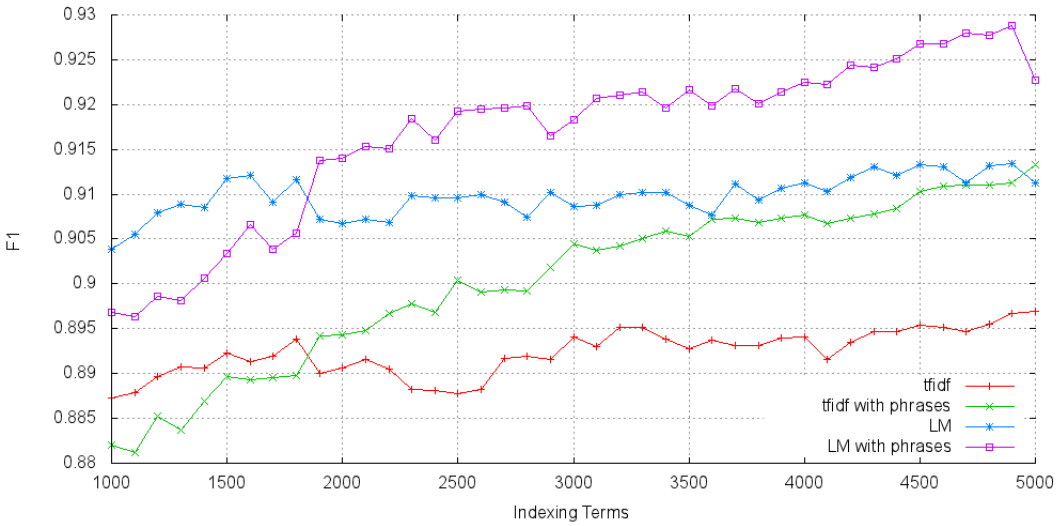

Figure 4.4: Document Expansion with statistical phrases on the Medline dataset.

of overfitting the classifier to the training collection increases. It is very probable that the continuous increase in performance for larger feature vectors, noticeable in Figure 4.4, is partially due to this effect.

Table 4.3: Performance measures for tfidf and LM indexer with and without statistical phrases.

techniques	precision	recall	F1	FP	FN
tfidf	0.889	0.905	0.897	246	205
tfidf with phrases	0.926	0.901	0.913	157	214
LM	0.896	0.927	0.911	232	159
LM with phrases	0.92	0.926	0.923	175	161

4.4 Expansion with UMLS Concepts

With the introduction of the UMLS Metathesaurus as source of background knowledge, there are many techniques for document expansion. Figure 4.5 shows and evaluation of a selected combination of expansion techniques for several feature vector sizes and compares them to classification without any document expansion ('Ontology=false'). There are three implemented concept-strategies which determine how the identified concepts are introduced to the text. The first strategy, i.e. 'replace', is to replace the original terms of the identified concept with its identifier. The strategy 'add' appends the concept-identifier without removing any of the terms.

The third strategy 'only' is to ignore all original terms and only use the identified concepts as indexing features. In addition, the three algorithms to handle WSD 'first', 'all' and 'context' are evaluated. The algorithm 'first' always uses the first sense, i.e. the one with the lowest ID. The strategy 'all' selects all possible senses. The third strategy takes the context in which the concept appears into account. The algorithm is explained in detail in Section 3.5.3. The evaluation is performed with concept-generalization depth of up to 2. A depth of 0 means that only the identified concepts are used.

The evaluation clearly shows that using only the identified concept as indexing features ('C-Strategy=only') delivers inferior results. The best result with only indexing the concepts was $F1 = 86.56\%$ (5000 features, generalization depth of 1, polysemy-strategy 'first'). Another rather obvious conclusion that can be drawn from the evaluation is that the introduction of background knowledge demands for more indexing features. While the classification without document expansion seems not to profit from using more than 1500 features, the best results with expansion are achieved with a feature vector size of 4000. The classifier performs best if the identified concepts replace the full designation in the free text instead of being added in addition to the original terms. It seems reasonable to assume that the improvement is due to the elimination of the stochastic dependence between the individual terms of the designation and the concept-identifier. Against expectation, the context sensitive WSD algorithm did not improve the classification performance against the simple 'first' strategy. In contrast to other experiments [HSS03, WHZC09], the evaluation also shows that introducing more generalized concepts does not improve the classification performance. In general it actually deteriorates the performance of the system. The best performance is achieved with the concept-strategy 'replace', the polysemy-strategy 'first', no concept-generalization and a feature vector size of 4000.

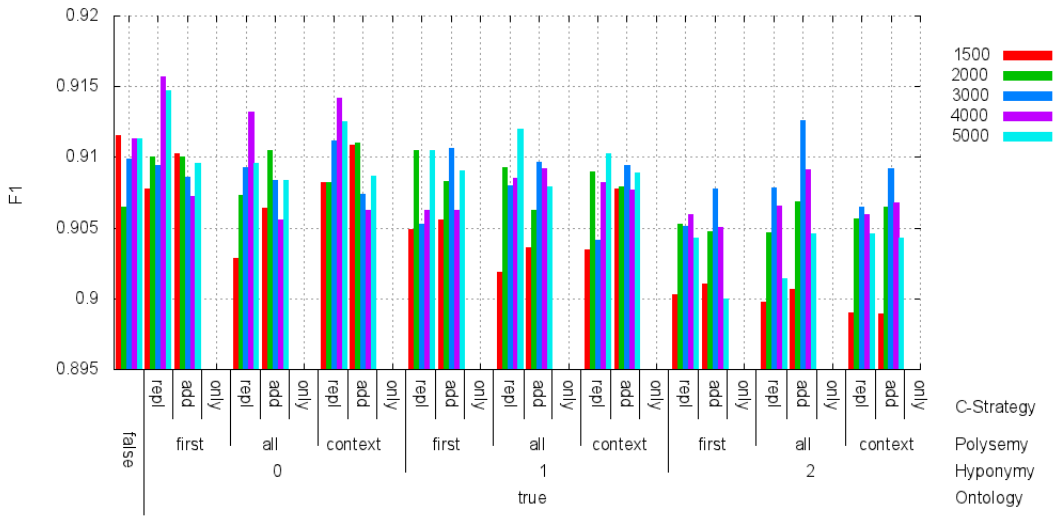

Figure 4.5: Document Expansion with the UMLS Metathesaurus using LM indexing on the Medline dataset.

4.5 Integration of UMLS Concepts and Statistical Phrases

Figure 4.6 shows the results of a classifier which uses statistical phrases (SP) and UMLS concepts for document expansion. Of the three implemented concept strategies only 'first' and 'replace' are evaluated since they showed the best performance in previous experiments (see Section 4.4). Since concept generalization seems to decrease performance in general, only the identified concepts were used to augment the documents. While the expansion with statistical phrases obviously delivers solid results, the addition of concepts seems to improve representation further if the size of the feature vector is increased. The evaluation also shows quite clearly that both variants of document expansion, UMLS concepts and statistical phrases, can increase classification performance. There seems to be no significant performance improvement through the use of context based WSD. In most of the experiments the simple 'first' resolution strategy performed equal or better than the other techniques.

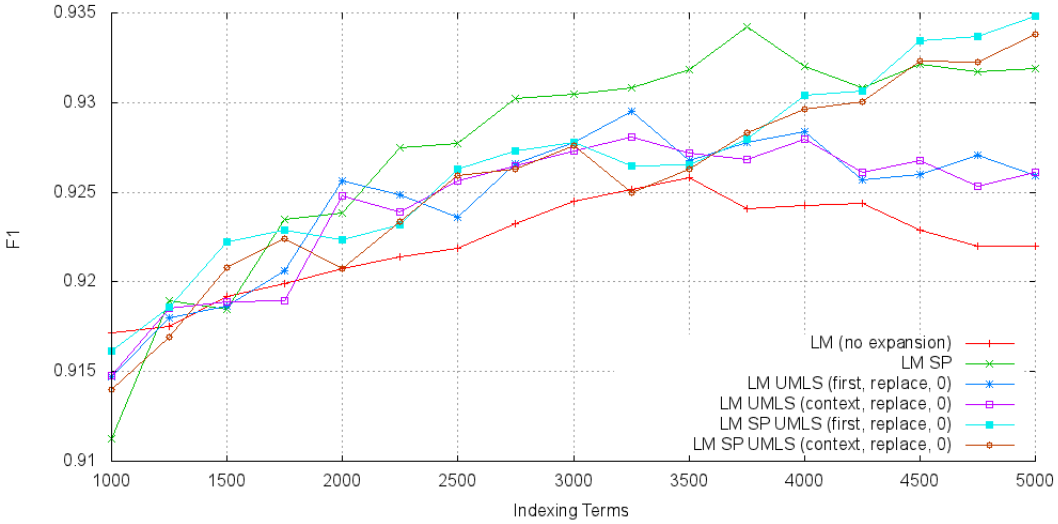

Figure 4.6: Expansion with statistical phrases and UMLS concepts on the Medline+Embase collection.

Previously only the F1 performance measure was use to show the performance of the classifiers. If we look at the individual measures recall and precision in Table 4.4, expansion with statistical phrases delivers the best precision and expansion with UMLS concepts the best recall. A classifier which combines both expansion techniques seems to achieve a better balance between the two measures.

Table 4.4: Effectiveness for different combinations of techniques on the Medline+Embase collection.

techniques	#features	accuracy	precision	recall	F1
baseline	2017	0.8857	0.8827	0.9006	0.8916
LM	3500	0.9208	0.9054	0.9471	0.9258
LM SP	3750	0.9310	0.**9294**	0.9391	0.9342
LM UMLS (FR0[3])	3250	0.9250	0.9121	**0.9476**	0.9295
LM UMLS (CR0[4])	3250	0.9234	0.9096	0.9474	0.9281
LM SP UMLS (FR0[3])	5000	**0.9315**	0.9283	0.9415	**0.9348**
LM SP UMLS (CR0[4])	5000	0.9303	0.9254	0.9424	0.9338

4.6 Stability

So far, all depicted results were obtained using 10-fold cross validation. For a k-fold cross validation, the collection is first partitioned randomly into k subsets. For each of the k classification runs, a classifier is trained on $k-1$ subsets and evaluated on the remaining subset not used for training. The result of the k-fold cross validation is the average of all k runs with each of the k subsets used exactly once for evaluation. While the average of a cross validation gives a good estimate of the expected performance of the classifier, other characteristics are also important as well. The *worst case* and the *variance* of the 10 runs shows how stable a classifier performs when presented with previously unseen input data. Figure 4.7 depicts the results of the individual runs from the previous experiment. The baseline classifier shows the highest scattering followed by the LM classifier utilizing statistical phrases. It also produced one result that was much worse than the average performance while also delivering the best overall run. The classifier which produced the best average F1-score (LM indexing, expansion with SP and UMLS (FR0) with polysemy strategy 'first', concept strategy 'replace' and generalization depth 0) is also the most stable one and has the best worst case performance.

[3]Polysemy strategy 'first', concept strategy 'replace', generalization depth 0.
[4]Polysemy strategy 'context', concept strategy 'replace', generalization depth 0.

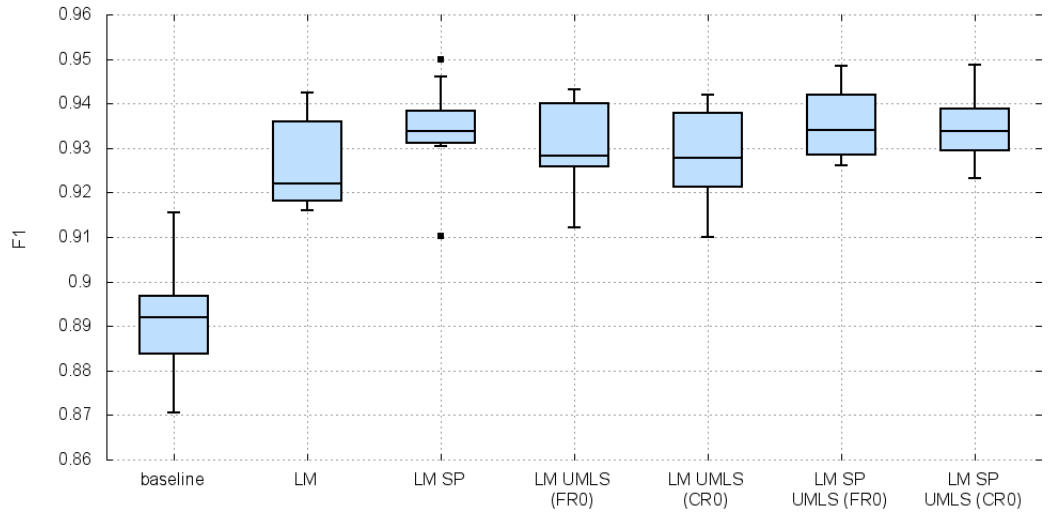

Figure 4.7: Stability of the techniques, depicting single runs of the previous 10-fold cross validation.

4.7 Precision vs Recall

While in practice both performance measures precision and recall are important, it is sometimes beneficial to improve one at the expense of the other. Figure 4.8 shows a ROC curve for the evaluation of the best combination of techniques against the baseline classifier. The ROC graph was generated by Weka and is an estimate of how the true positive rate would change if the rate of allowed false positives were to be changed. Weka optionally produces class predictions instead of discrete class labels if the option *'build logistic models'* for the SMO classifier is selected. The two classifiers depicted in the ROC graph are both near their BEP in an area of the ROC where the tradeoff between recall and precision is almost linear. Depending on the specific task it is possible to increase either precision or recall to a very high value with acceptable error rates. An estimate from the ROC shows that with the best classifier a recall of 97.25% would be possible at the expense of a precision of 90%. The AUC for the best classifier is 0.979 compared to 0.944 for the baseline classifier.

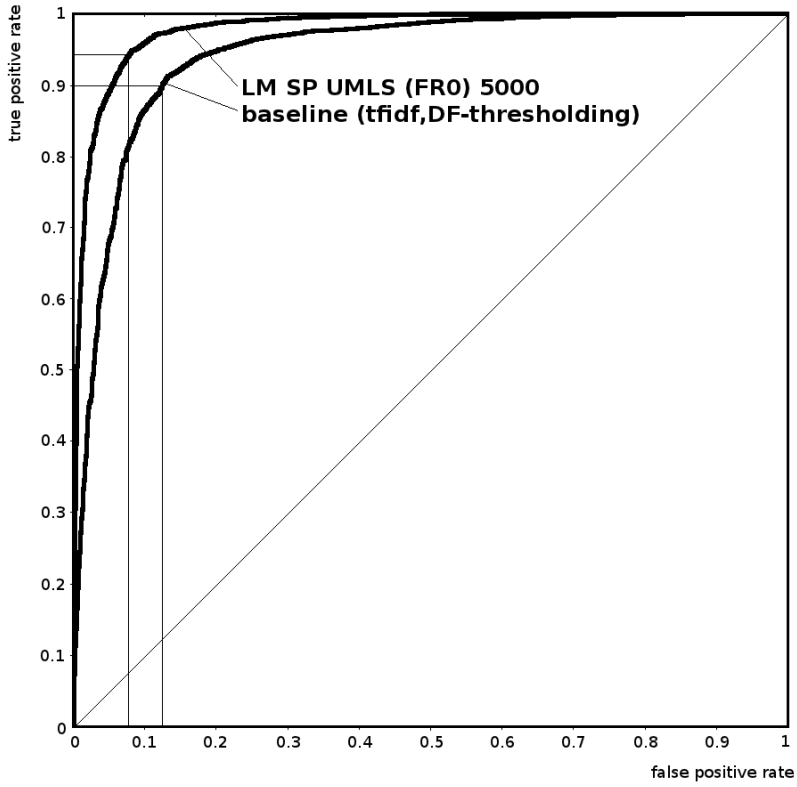

Figure 4.8: ROC curve for the baseline classifier and the best combination of techniques.

4.8 Validation on New Data

All previous experiments were carried out using 10-fold cross validation which prevents overfitting to a certain degree, but it is still only an approximation of the real performance of the classifier. More realistic results can be obtained by evaluating the classifier against previously unseen data which is not present during parameter optimization and training of the classifier. The validation set contains 500 samples of which 79 are relevant and 421 are irrelevant to the topic of off-label drug use. The classifiers are trained with the best combinations of techniques previously observed in Section 4.5 and the optimal penalty factor C in respect to the size of the feature vector, identified in Section 3.9.3.

Table 4.5 shows the results of the validation runs on the different classifiers. The improvements achieved by document expansion decrease almost only the false positives. Also the misclassified documents are not, as could be expected, those with very few indexing terms. The average amount of features per documents after expansion with phrases and UMLS concepts was 44 while the misclassified documents were represented by an average of 59 indexing features.

While the accuracy achieved by the classification runs seems to be excellent, the precision and recall are worse than in the cross validation previously performed on the Medline and Embase dataset. The lower precision is expected due to the different sizes of the two classes in the test and validation set. The measure favors true positives over false negatives. A classifier which assigns class labels randomly would have a precision of 52% on the Medline+Embase training set (4236 relevant and 3883 irrelevant). The same classifier, however, would only achieve a precision of 15% on the validation set. The relatively small amount of positive samples in the validation set has also the negative side effect of producing rather big changes in the effectiveness measures. The misclassification of one relevant document decreases the recall by 1.27%.

It can reasonably be assumed that the 10-fold cross validation shows too optimistic results. An explanation for this is that the feature selection is performed before the cross validation on the whole corpus. This means that the feature selection function has data about feature occurrence it does not have in a normal setting. This explains why, contrary to the results from the 10-fold cross validation, the combination of phrases and concepts did not improve classification performance compared to expansion with only UMLS concepts. Also indexing with 5000 feature seems to overfit the classifier since the precision of the system is actually better when only using 3000 features.

Table 4.5: Evaluation of the previously trained classifiers on the validation set.

techniques	#features	C	accuracy	precision	recall	F1	FP	FN
baseline	2017	1	0.9380	0.7609	0.8861	0.8187	22	9
LM	3500	0.6	0.9500	0.8068	0.8987	0.8503	17	8
LM SP	3750	0.6	0.9620	0.8659	0.8987	0.8820	11	8
LM UMLS (FR0)	3250	0.6	**0.9660**	0.8781	**0.9114**	**0.8944**	10	**7**
LM UMLS (CR0)	3250	0.6	**0.9660**	0.**8875**	0.8987	0.8939	**9**	8
LM SP UMLS (FR0)	5000	0.4	0.9560	0.8353	0.8987	0.8659	14	8
LM SP UMLS (CR0)	5000	0.4	0.9620	0.8571	**0.9114**	0.8834	12	**7**
LM SP UMLS (FR0)	3000	0.6	0.9620	0.8659	0.8987	0.8820	11	8
LM SP UMLS (CR0)	3000	0.6	0.9640	0.8765	0.8987	0.8875	10	8

4.9 Time Dependency

The training collection contains articles from a rather long timespan, 1947 to 2011, as previously described in Section 3.1.2. While the definition of the topic off-label drug use has not changed during this time, the importance of many terms and concepts for it has. A drug can, for instance, be approved for the treatment of a specific medical condition by a regulatory agency changing

its relevancy for the topic practically instantly. Several examples for terms and phrases and their distribution in the two classes over time are given in Section 3.1.4.

With 80% of the documents from the training collection being published 1998 or later, the collection contains more than enough recent training samples. However, since there are many terms whose relevance for the topic changes over time, the reduction of old training samples could be beneficial for the performance of the trained classifier. Table 4.6 depicts the evaluation results of several classifiers which were trained only with documents published since 1980. The 267 oldest articles were removed from the training collection. The evaluation results show an increase in performance for the classifiers that augment the documents with UMLS concepts. This clearly indicates that old training samples, created before 1980, decrease the performance of the classifier in some cases. Removing articles published prior to 1982, however, resulted in a decline of classification performance.

Table 4.6: Evaluation of classifiers trained with articles published 1980 or later.

techniques	#features	C	accuracy	precision	recall	F1	FP	FN
baseline	2049	1	0.9340	0.7347	**0.9114**	0.8136	26	**7**
LM	3500	0.6	0.9480	0.7978	0.8987	0.8452	18	8
LM SP	3750	0.6	0.9600	0.8554	0.8987	0.8765	12	8
LM UMLS (FR0)	3250	0.6	**0.9680**	**0.8889**	0.9114	**0.9000**	**9**	7
LM UMLS (CR0)	3250	0.6	**0.9680**	0.8706	**0.9367**	**0.9024**	11	**5**
LM SP UMLS (FR0)	5000	0.4	0.9520	**0.8395**	0.8608	0.8500	**13**	11
LM SP UMLS (CR0)	5000	0.4	0.9500	0.8214	0.8734	0.8466	15	10

4.10 Summary

Empirical evaluation shows that stemming and the reduction of words from the predefined SMART stoplist in general improve the classification performance. However, it has also been demonstrated that both techniques have shortcomings in individual cases. Of the 10 feature selection functions evaluated in this chapter, the CDR performed best for reasonably sized feature vectors.

The best feature weighting function for the evaluated task is LM, which clearly outperformed both of the other tested functions, tfidf and LSI. Latent semantic indexing was only possible after a prior reduction to 3000 distinct terms since the deployed algorithm for SVD has a computational complexity that is quadratic with respect to the size of the feature vector.

An analysis of the individual runs of the 10-fold cross validation shows that the classifier with the best average performance also has the least scattering and the best worse case performance.

However, it has to be assumed that the 10-fold cross validation shows too optimistic results since the feature selection is performed before the cross validation on the whole corpus. This also gives the selection of statistical phrases from the corpus an unrealistic advantage over a real world scenario.

The ROC curve, which was generated by Weka, depicts the tradeoff between recall and precision. Depending on the requirements of the specific task, either recall or precision can be increased at the expense of the other. E.g., a recall of 97.25% would be possible at the expense of a precision of 90%.

A small increase in classification performance can be observed if the 267 articles which were published prior to 1980 are removed from the training set. This shows that the system is, to some degree, time dependent. However, if more recent articles are removed from the training set the performance of the system deteriorates.

CHAPTER 5

Conclusion

This chapter first revisits the experimental results and compares the performance of the implemented techniques to related work. While document enrichment with background knowledge in general improved the classification performance, concept generalization and the resolution of ambiguous concepts did not. Future work will have to be conducted to address these and other issues.

5.1 Evaluation and Comparison

Without any document expansion, the best results are achieved with approximately 3500 terms for the combined dataset (Section 4.5). If the documents are enriched with either statistical phrases, UMLS concepts or both, the classifier shows better results if more features are used to index the documents. This clearly shows that there is information present in the documents that cannot be represented by simple BOW indexing. However, by using concepts represented by more than one word, in addition to conventional term indexing, a noticeable improvement can be achieved. The validation on previously unseen data shows that the document expansion techniques primarily improve the precision of the classifier. This effect can also be observed in the cross validation on the training set however not in the same scale.

In contrast to results from related work, no improvement by concept generalization was observed in any of the experiments performed. Two related systems, by comparison, use generalization of concepts from WordNet [HSS03] and Wikipedia [WHZC09] and report improved performance for generalization depths of up to 5 for WordNet concepts and the expansion with direct hypernyms from Wikipedia. A problem encountered during the integration of the UMLS into the application was that many concepts from the Metathesaurus have a functional or navigational purpose only. Despite the elimination of many such concepts before enriching the documents,

the hierarchy obviously is still not strict enough to improve document representation through concept generalization. In the present implementation the generalization seems to introduce additional noise to the documents. A similar effect has been observed in related work. If too many alias symbols are added to a query, an effect described as *query drift* decreases performance [BCC04]. The original meaning of the query or document is somewhat overshadowed by the high amount of added concepts. The relatively large number of direct hypernyms in the UMLS Metathesaurus seems to have a similar effect on the documents in this application.

Contrary to expectations, the WSD algorithm implemented did not perform significantly better than the primitive solution of choosing the first concept retrieved. It is unclear why this is the case. However, polysemes seem to have a significant impact on the classification performance. One possible explanation is that the positive effect on document representation, gained by identifying different senses in a polysemous term, is lower than the negative effect resulting from splitting one concept into several near synonymous concepts. A more detailed analysis of the implemented WSD algorithm and its performance would be necessary to assess the usefulness for the TC task examined in this thesis.

According to the validation on previously unseen data it can be assumed that some form of overfitting effect shows too optimistic results for the expected recall in the 10-fold cross validation. One probable cause is that the term selection is performed before the cross validation on the whole corpus. This gives the feature selection function an advantage because it *knows* about features which are not present in the training set. This would also explain why document enrichment with statistical phrases performed better than enrichment with UMLS concepts in the cross validation but not on the validation set.

While the experiments performed clearly show the improvements through document enrichment, the gain is probably not as high as hoped for. Related work with thesaurus-based query expansion has shown that the addition of UMLS concepts improves retrieval performance for some queries and decreases it for others [HPDD00]. It is reasonable to assume that similar effects come into play in this task as well.

An interesting fact, previously shown in literature [MKSS04], is that abstracts seem to contain less specific keywords than full text and in general express information in a more compressed way. Experiments showed a better overall performance when using full text instead of only abstracts. The collection used in this thesis, however, only contained abstracts and single word subjects. Adding the full article texts would most likely further improve performance.

The validation in Section 4.8 shows that almost all improvements in performance are gained through reducing the false positives. While document enrichment improves the precision by up to 8%, practically no changes in recall can be observed. It should also be mentioned that for some of the documents of the dataset, there was too little text available to label them even for the human domain expert. For those documents, the full article texts were acquired which were however not available for the automatic text classification performed in this thesis. It is reasonable to assume that his has a negative effect on the classification performance reported, compared to human judgment.

The novel feature selection function class discrimination ratio (CDR) presented in this thesis outperformed existing functions. The selection is very aggressive towards highly class discriminating features. However, it also takes the frequency of the feature in the collection into account. It is up to further experiments to explore the performance of the CDR function for other applications.

The results presented in Section 4.2 clearly show that LM is the best performing feature weight for this application. The weak performance of LSI is somewhat disappointing. However, it is possible that the term reduction prior to compression with SVD is partially responsible for that. Considering the computational complexity involved even with only 3000 features and roughly 5000 training documents, the SVD implementation used for this thesis is not feasible for large scale use.

5.2 Summary

The goal of this thesis was to compare state of the art techniques for text classification systems and to explore the use of domain specific background knowledge to improve the performance of such a system. Besides simple DF-thresholding, many other feature reduction functions were evaluated. The CDR allows aggressive feature reduction and performed best of all evaluated functions. Of the three tested feature weights, language modeling (LM) clearly outperformed tfidf and latent semantic indexing (LSI). While LSI initially seemed to be a promising technique, the involved computational costs make it not suitable for indexing over 8000 documents and over 40 000 distinct terms. Aside from the typical BOW approach, the use of statistical phrases and UMLS concepts for indexing was explored. Adding common phrases as indexing features shows relatively big improvements, however, as expected UMLS concepts performed even better. Using domain specific background knowledge can resolve several of the problems arising from the complex medical terminology. While the elimination of synonyms obviously reduced noise in the dataset, the expected gain from ambiguity resolution and generalization could not be observed. The semantic connections in the UMLS Metathesaurus seem to be very loose which resulted in additional noise when adding hypernyms to the index. Further research of the semantic connections between the UMLS concepts will be necessary to make these two techniques beneficial for document expansion.

Starting from a simple but solid baseline, the techniques investigated in this thesis improved the classification performance significantly. Considering the semantic similarity of the documents in the collection resulting from the search query used to retrieve them, the performance of the system is quite satisfying. Since off-label drug use seems to increase in recent years, as indicated by analysis of the collection, a *monitoring system* could be beneficial. The application developed for this thesis could serve as a document filtering system to assist a human expert in monitoring new articles for their relevancy to a specific topic. Another possible application would be automated pandemic detection through the monitoring of social network services like twitter.

5.3 Future Work

The expansion of documents with background knowledge is the most promising method to improve performance of a domain specific classification system. While adding more generalized concepts can improve performance, as indicated by related work [WHZC09, HSS03], this could not be observed by the experiments performed for this thesis. Most likely, the high amount of hypernyms in the UMLS adds too much noise to the documents which causes a topic drift. Also despite manually removing many functional and navigational concepts there were still too many concepts not suitable for document expansion left. Further efforts in removing useless concepts and selecting only one hypernym instead of using all parent concepts should be made. It is also not clear if the strategy to resolve ambiguous concepts actually improves the performance. The implementation was not extensively tested on its own. A performance analysis of the algorithm, especially in comparison to related work [HSS03, AR96], could help to improve the ambiguity resolution further.

The overall impact on performance due to stemming and removing of stop words from a predefined list seems to be positive. However, there also seems to be the need for a domain specific stoplist and possibly even an adapted stemming algorithm as indicated by examples from the collection provided in Section 4.1.1.

All experiments were only performed on article abstracts together with some key phrases. It is open to question how the system would perform when presented with full article texts. Since the documents were rather short and the length did not vary much, no document length normalization was performed. In practice an application should be able to classify abstracts and full text reasonably well at the same time. Further experiments as well as document length normalization are needed to ensure that the classification system can handle arbitrary input documents.

It is indicated by related work that expansion through UMLS concepts improves performance in some cases and decreases it in others. No detailed investigation of this assumption has been carried out in this thesis. It is up to further research to investigate this issue and possible develop an algorithm to identify which concepts are beneficial for expansion and which actually hurt performance.

There is clearly potential for further improvements in the field of medical text classification. Several areas which could profit from detailed examination have been pointed out in this thesis. Since digitally available information will continue to grow in the foreseeable future, it is also reasonable to assume that IR will grow even more important in the years to come.

APPENDIX A

Acronyms

AUC area under the ROC curve
ARFF attribute-relation file format
BEP break-even point
BMI body mass index
BOW bag of words
CDR class discrimination ratio
DF document frequency
DOM document object model
DR dimensionality reduction
FP false positive
FN false negative
IG information gain
IR information retrieval
LM language modeling
LSI latent semantic indexing
MeSH medical subject headings
MI mutual information
ML machine learning
NLP natural language processing
OR odds ratio
POS part of speech
RBF radial basis function
RNN reciprocal nearest neighbor
ROC receiver operating characteristic
SMO Sequential Minimal Optimization
SVD singular value decomposition

SVM support vector machine
TC text classification
TF term frequency
TN true negative
TP true positive
UMLS Unified Medical Language System
Weka Waikato Environment for Knowledge Analysis
WSD word sense disambiguation

APPENDIX B

Index

List of Figures

List of Tables

APPENDIX C

Listings

C.1 SMART Stoplist

a, as, able, about, above, according, accordingly, across, actually, after, afterwards, again, against, aint, all, allow, allows, almost, alone, along, already, also, although, always, am, among, amongst, an, and, another, any, anybody, anyhow, anyone, anything, anyway, anyways, anywhere, apart, appear, appreciate, appropriate, are, arent, around, as, aside, ask, asking, associated, at, available, away, awfully, b, be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, believe, below, beside, besides, best, better, between, beyond, both, brief, but, by, c, cs, cmon, came, can, cant, cannot, cant, cause, causes, certain, certainly, changes, clearly, co, com, come, comes, concerning, consequently, consider, considering, contain, containing, contains, corresponding, could, couldnt, course, currently, d, definitely, described, despite, did, didnt, different, do, does, doesnt, doing, dont, done, down, downwards, during, e, each, edu, eg, eight, either, else, elsewhere, enough, entirely, especially, et, etc, even, ever, every, everybody, everyone, everything, everywhere, ex, exactly, example, except, f, far, few, fifth, first, five, followed, following, follows, for, former, formerly, forth, four, from, further, furthermore, g, get, gets, getting, given, gives, go, goes, going, gone, got, gotten, greetings, h, had, hadnt, happens, hardly, has, hasnt, have, havent, having, he, hes, hello, help, hence, her, here, heres, hereafter, hereby, herein, hereupon, hers, herself, hi, him, himself, his, hither, hopefully, how, howbeit, however, i, id, ill, im, ive, ie, if, ignored, immediate, in, inasmuch, inc, indeed, indicate, indicated, indicates, inner, insofar, instead, into, inward, is, isnt, it, itd, itll, its, its, itself, j, just, k, keep, keeps, kept, know, knows, known, l, last, lately, later, latter, latterly, least, less, lest, let, lets, like, liked, likely, little, look, looking, looks, ltd, m, mainly, many, may, maybe, me, mean, meanwhile, merely, might, more, moreover, most, mostly, much, must, my, myself, n, name, namely, nd, near, nearly, necessary, need, needs, neither, never, nevertheless, new, next, nine, no, nobody, non, none, noone, nor, normally, not, nothing, novel, now, nowhere, o, obviously, of, often, oh, ok, okay, old, on, once, one, ones, only, onto, or, other, others, oth-

erwise, ought, our, ours, ourselves, out, outside, over, overall, own, p, particular, particularly, per, perhaps, placed, please, plus, possible, presumably, probably, provides, q, que, quite, qv, r, rather, rd, re, really, reasonably, regarding, regardless, regards, relatively, respectively, right, s, said, same, saw, say, saying, says, second, secondly, see, seeing, seem, seemed, seeming, seems, seen, self, selves, sensible, sent, serious, seriously, seven, several, shall, she, should, shouldnt, since, six, so, some, somebody, somehow, someone, something, sometime, sometimes, somewhat, somewhere, soon, sorry, specified, specify, specifying, still, sub, such, sup, sure, t, ts, take, taken, tell, tends, th, than, thank, thanks, thanx, that, thats, thats, the, their, theirs, them, themselves, then, thence, there, theres, thereafter, thereby, therefore, therein, theres, thereupon, these, they, theyd, theyll, theyre, theyve, think, third, this, thorough, thoroughly, those, though, three, through, throughout, thru, thus, to, together, too, took, toward, towards, tried, tries, truly, try, trying, twice, two, u, un, under, unfortunately, unless, unlikely, until, unto, up, upon, us, useful, usually, uucp, v, value, various, very, via, viz, vs, w, want, wants, was, wasnt, way, we, wed, well, were, weve, welcome, well, went, were, werent, what, whats, whatever, when, whence, whenever, where, wheres, whereafter, whereas, whereby, wherein, whereupon, wherever, whether, which, while, whither, who, whos, whoever, whole, whom, whose, why, will, willing, wish, with, within, without, wont, wonder, would, would, wouldnt, x, y, yes, yet, you, youd, youll, youre, youve, your, yours, yourself, yourselves, z, zero

C.2 Removed UMLS concepts

Table C.1: Functional UMLS concepts which were removed from the ontology.

CUI	Concept name	*isa* relations
C1285556	Navigational concept	638
C1285536	Procedure categorized by device involved	720
C0450973	Assessment scales	867
C1274015	Erroneous concept (inactive concept)	1152
C1274014	Outdated concept (inactive concept)	1498
C2584795	Parenteral dosage form product	2068
C2586094	Oral dosage form product	2712
C1276325	Reason not stated concept (inactive concept)	7529
C1274021	Moved elsewhere (inactive concept)	14457
C1274012	Ambiguous concept (inactive concept)	16110
C2733115	Limited status concept (inactive concept)	20930
C1274013	Duplicate concept (inactive concept)	37815
C1264758	Inactive concept (inactive concept)	7
C1298232	Special concept	3

APPENDIX D

Bibliography

[AMWZ09] Timothy G. Armstrong, Alistair Moffat, William Webber, and Justin Zobel. Improvements That Don't Add Up: Ad-Hoc Retrieval Results Since 1998. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 601–610, New York, NY, USA, 2009. ACM.

[AR96] Eneko Agirre and German Rigau. Word sense disambiguation using Conceptual Density. In *Proceedings of the 16th Conference on Computational Linguistics*, volume 1 of *COLING '96*, pages 16–22, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

[Aro01] Alan R. Aronson. Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program. In *Proceedings of the AMIA Symposium*, pages 17–21, 2001.

[BCC04] Stefan Büttcher, Charles L. A. Clarke, and Gordon V. Cormack. Domain-Specific Synonym Expansion and Validation for Biomedical Information Retrieval. In *Proceedings of the Thirteenth Text Retrieval Conference*, 2004.

[BCH06] Stephan Bloehdorn, Philipp Cimiano, and Andreas Hotho. Learning Ontologies to Improve Text Clustering and Classification. In *From Data and Information Analysis to Knowledge Engineering: Proceedings of the 29th Annual Conference of the German Classification Society*, volume 30 of *Studies in Classification, Data Analysis, and Knowledge Organization*, pages 334–341. Springer, 2006.

[Bod04] Olivier Bodenreider. The Unified Medical Language System (UMLS): Integrating Biomedical Terminology. *Nucleic Acids Research*, 32(Database-Issue):267–270, 2004.

[BZ11] Roi Blanco and Hugo Zaragoza. Beware of Relatively Large but Meaningless Improvements. Technical report, Yahoo! Research 2011-001, 2011.

[CMF01] Maria F. Caropreso, Stan Matwin, and Sebastiani Fabrizio. *A Learner-independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization*, pages 78–102. IGI Publishing, Hershey, PA, USA, 2001.

[CVBM02] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing Multiple Parameters for Support Vector Machines. *Machine Learning*, 46(1-3):131–159, 2002.

[DDF+90] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[Dol92] Lewis D. Dolan. An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, pages 37–50, New York, NY, USA, 1992. ACM.

[Fab02] Sebastiani Fabrizio. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34:1–47, March 2002.

[Faw06] Tom Fawcett. An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.

[GFS00] Luigi Galavotti, Sebastiani Fabrizio, and Maria Simi. Experiments on the Use of Feature Selection and Negative Evidence in Automated Text Categorization. In *Proceedings of ECDL-00, 4th European Conference on Research and Advanced Technology for Digital Libraries*, pages 59–68. Springer Verlag, 2000.

[HCL03] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A Practical Guide to Support Vector Classification. Technical report, Department of Computer Science, National Taiwan University, 2003.

[HPDD00] William Hersh, Susan Price, Larry Donohoe, and Larry Donohoe. Assessing Thesaurus-based Query Expansion Using the UMLS Metathesaurus. In *Proceedings of the 2000 American Medical Informatics Association (AMIA) Symposium*, pages 344–348, 2000.

[HSS03] Andreas Hotho, Steffen Staab, and Gerd Stumme. Wordnet Improves Text Document Clustering. In *Proceedings of the SIGIR 2003 Semantic Web Workshop*, pages 541–544, 2003.

[Joa98] Thorsten Joachims. Text Categorization with Support Vector Machines: Learning With Many Relevant Features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Heidelberg, 1998. Springer.

[LHM93] Donald A. B. Lindberg, Betsy L. Humphreys, and Alexa T. McCray. The Unified Medical Language System. *Methods of Information in Medicine*, 32(4):281–291, 1993.

[MCAC97] Mitra Mandar, Buckley Chris, Singhal Amit, and Cardie Claire. An Analysis of Statistical and Syntactic Phrases. In *Proceedings of the RIAO'97*, pages 200–217, 1997.

[MG98] Dunja Mladenic and Marko Grobelnik. Word Sequences as Features in Text-Learning. In *Proceedings of the 17th Electrotechnical and Computer Science Conference (ERK98)*, pages 145–148, 1998.

[MKSS04] Hans-Michael Müller, Eimear E. Kenny, Paul W. Sternberg, and Paul W. Sternberg. Textpresso: An Ontology-Based Information Retrieval and Extraction System for Biological Literature. volume 2, page 309, 2004.

[MMH12] Bita Mesgarpour, Markus Müller, and Harald Herkner. Search strategies - identified reports on 'off-label' drug use in MEDLINE. *Journal of Clinical Epidemiology*, 65(8):827–834, 2012.

[PC98] Jay M. Ponte and Bruce W. Croft. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.

[PPPC07] Ted Pedersen, Serguei V. S. Pakhomov, Siddharth Patwardhan, and Christopher G. Chute. Measures of Semantic Similarity and Relatedness in the Biomedical Domain. *Journal of Biomedical Informatics*, 40(3):288–299, june 2007.

[RS02] Miguel E. Ruiz and Padmini Srinivasan. Hierarchical Text Categorization Using Neural Networks. *Information Retrieval*, 5(1):87–118, 2002.

[Sal71] Gerard Salton. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[SB88] Gerard Salton and Christopher Buckley. Term-weighting Approaches in Automatic Text Retrieval. In *Information Processing and Management*, volume 24, pages 513–523, Tarrytown, NY, USA, 1988. Pergamon Press, Inc.

[Sta08] Randall S. Stafford. Regulating Off-label Drug Use - Rethinking the Role of the FDA. *The New England Journal of Medicine*, 358(14):1427–1429, 2008.

[WHZC09] Pu Wang, Jian Hu, Hua-Jun Zeng, and Zheng Chen. Using Wikipedia Knowledge to Improve Text Classification. *Knowledge and Information Systems*, 19(3):265–281, 2009.

[WPW95] Erik Wiener, Jan O. Pedersen, and Andreas S. Weigend. A Neural Network Approach to Topic Spotting. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, 1995.

[Yan99] Yiming Yang. An Evaluation of Statistical Approaches To Text Categorization. *Journal of Information Retrieval*, 1:67–88, 1999.

[YL99] Yiming Yang and Xin Liu. A Re-examination of Text Categorization Methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 42–49, New York, NY, USA, 1999. ACM.

[YP97] Yiming Yang and Jan O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.