**Budapest University of Technology and Economics**
**Department of Automation and Applied Informatics**

# SEARCHING FOR SIMILAR DOCUMENTS WITH LIMITED RESOURCES

---

# HASONLÓ DOKUMENTUMOK KERESÉSE KORLÁTOS ERŐFORRÁSÚ KÖRNYEZETBEN

Ph.D. Thesis

Kristóf Csorba

Advisor: István Vajk, D.Sc.

June 2010

# Abstract

The development of mobile devices is very fast in the last years: mobile phones and PDAs have more-and-more processing power and storage capacity. This leads to a rapidly increasing number of application areas. Nowadays it is not unusual that someone stores documents, like e-books for instance, on a PDA. Search engines allowing access to such locally stored public resources are still under development. This dissertation presents theoretical foundations for a search system designed to find similar documents in a network of mobile devices. By taking the limited processing power into account and maintaining low communication traffic, the system is searching for remote documents with topics similar to the locally stored documents. Assuming that a user is interested in the topic of the locally stored documents, the system can search for interesting documents without user interaction. The user is only notified, if a candidate document is found. In this case, the user is asked wether the document should be downloaded or not. This process consists of two main parts: creating the representation of the local documents for the comparison, and performing the similarity search. A low communication traffic is required to avoid depletion of the mobile device's battery in a few hours due to continuous transmission. To achieve this, a topic specific keyword-based, compact document representation is proposed where the presence of topic specific keywords is used to compare documents. The first part of the contribution contains the keyword selection. The second part uses this to define the compact document representation and the similarity search. Additionally, a document extension technique is proposed to take words closely related to the keywords into account. The third part of the contribution provides two improvements to the previously described methods: a decision tree-like document topic identification method aiming to decrease the number of comparisons during topic identification, and a serial cascade of 1-class classifiers to increase the number of detected documents. Several theoretical and experimental results validate the applicability of the proposed techniques for the similarity search performed between mobile devices.

# Összefoglaló

A mobiltelefonok és egyéb mobil eszközök fejlődése nagyon felgyorsult az utóbbi időben. A növekvő számítási teljesítmény és tároló kapacitás eredményeként rohamosan nő az alkalmazási területek száma is. Manapság nem ritka, hogy valaki elektronikus könyveket olvas a PDA-ján. Viszont jelenleg még fejlesztés alatt állnak azok a kereső rendszerek, melyek lehetővé tennék az ilyen, mobil eszközökön tárolt, nyilvános információk keresését és elérését. Mobil peer-to-peer kliensek vannak ugyan, de a keresés alapját még mindig kulcsszavak képezik, melyeket többnyire a fájlnevekben keres a rendszer. Ez a disszertáció egy olyan keresőrendszer elméleti alapjait tartalmazza, mely hasonló dokumentumokat keres a mobil eszközök között. Feltételezve, hogy egy felhasználót érdeklik a saját dokumentumainak a témái, egy hasonló dokumentumokat kereső rendszer teljesen automatikusan kereshet a felhasználó számára érdekes anyagokat. Csak akkor értesíti a felhasználót, ha talál egy esélyes dokumentumot és a felhasználónak el kell dönteni, hogy tényleg letöltse-e. A javasolt rendszer komoly hangsúlyt fektet az adatforgalom nagyságára, mivel a folyamatos adatátvitel egyrészt nem mindig ingyenes, másrészt gyorsan lemeríti az akkumulátort. Ezért egy tömör dokumentum reprezentációs eljárást használ, mely témaspecifikus kulcsszavak jelenléte vagy hiánya alapján hasonlítja össze a dokumentumokat. Az új eredmények első eleme a kulcsszó kiválasztó algoritmus. A második rész erre építve definiálja a tömör dokumentum reprezentációt, valamint a hasonló dokumentumok keresését. A kulcsszavak szinonimáinak kezelésére tartalmaz egy dokumentum bővítéses kiegészítést is. A harmadik rész további két kiegészítést tartalmaz: az első a dokumentumok témájának meghatározásához javasol egy többlépcsős, döntési fa jellegű megoldást, mely hatékonyan csökkenti a szükséges összehasonlítások számát. A másik javasolt kiegészítés a keresés többszöri egymás után fűzésével kapott, kaszkádosított osztályozó, mely a sikeresen felismert dokumentumok számát hivatott növelni. A bemutatott módszerek hasznosíthatóságát számos elméleti levezetés és mérési eredmény vizsgálja és támasztja alá.

# Acknowledgments

This thesis could not have been created without the help of many people. First of all I am grateful to my parents and my sister for their patience and support. I am very thankful to Tücsi because of her infinite tolerance and understanding.

I would like to thank István Vajk, my scientific advisor, for the inspiring consultations and professional support. I am also indebted to Hassan Charaf for providing the financial conditions of the work.

I would like to thank Bertalan Forstner, Zsolt Berényi, and all my Colleagues at the Department of Automation and Applied Informatics (Budapest University of Technology and Economics) for their help.

Many thanks for the many valuable comments and questions to everyone I talked to about my research.

This work is connected to the scientific program of the "Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project. This project is supported by the New Hungary Development Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

# Preface

## Dedication

The content of this thesis is a product of the author's original work except where explicitly stated otherwise.

## Nyilatkozat*

Alulírott Csorba Kristóf kijelentem, hogy ezt a doktori értekezést magam készítettem, és abban csak a megadott forrásokat használtam fel. Minden olyan részt, amelyet szó szerint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Budapest, 2010. Június

(Csorba Kristóf)

*A bírálatok és a védésről készült jegyzőkönyv a későbbiekben a Dékáni Hivatalban elérhetőek.

# Table of Contents

# List of Figures

# List of Tables

# List of Symbols

In this dissertation, the most important notations are related to words, documents, document topics and topic sets. The most common notations are listed in the following:

- $w$, $v$: words or keywords. Keywords are words selected by keyword selection.
- $d$: document, a set of words.
- $D$, $W$: set of all documents and all words in the data set respectively.
- $T$: topic, a set of documents.
- $\mathbb{T}$, $\mathbb{U}$: topic sets, like the set of off-topic topics $\mathbb{U}$.
- $\mathbb{TS}$: set of topic sets, its elements are topic sets.
- $K$, $K_T$: keyword lists, its topic can be given in subscript.
- $S(w)$: selector or 1-class classifier. The result is the set of documents containing the word $w$. Details and definition are presented on page 24.

# List of Most important expressions

The most important expressions used in my dissertation are the following:

Chapter 1

# Introduction

The development of mobile devices is very fast in the last years: mobile phones and PDAs have more-and-more processing power and storage capacity. This leads to a rapidly increasing number of application areas. Nowadays it is not unusual that someone stores documents, like e-books for instance, on a PDA. Search engines allowing access to such locally stored public resources are still under development. Peer-to-peer file sharing systems are already implemented for mobile devices[Forstner and Charaf., 2005], but the search is still based on search phrases entered by the user and it is usually looked for in the filenames.

The rapidly increasing amount of available information makes searching tools more-and-more important: instead of substring based searches, semantic search engines are introduced which allow more sophisticated searches. In semantic search, the goal is to capture as much as possible semantic information of the documents (or other media), and to allow searching for documents based on these meta-data. This may be implemented for example as automatic tagging and searches based on the tags assigned to the documents, but also as question answering systems presenting textual results which contain the answer on the question entered by the user.

According to the meta-data content of the documents, there are two main areas of research: the one storing semantic meta-data in the documents, like the semantic web approach, and the other processing pure natural language documents and apply various statistical and natural language processing (NLP) tools in order to retrieve the semantic information. Natural language processing involves

many techniques like morphologic analysis, part-of-speech tagging, and grammatical analysis for example. Beside these, many approaches have been proposed which capture the semantic relations using statistical methods like various classification techniques. Of course, these statistical approaches also often employ preprocessing steps like stemming or part-of-speech tagging.

Defining the scope of information someone wants to retrieve is another important question: a simple approach is to expect the user to define search phrases and to assume that documents containing these search phrases are related to the query. This may be enhanced by estimating the relevance of phrases with respect to a given document, and even by estimating the relevance of the documents themselves. These improvements can be achieved with one of the many TFIDF weighting schemes and the PageRank method for instance. Another way to define the scope of required information is to present examples, as in the topic-by-example approaches. In this case, documents similar to some predefined documents are searched for. If the users own documents are assumed to represent the interest fields of the user, searching for documents similar to the locally stored ones allows finding documents which are probably interesting for the user, and there is no need to define any search criteria explicitly. A mixture of the example and search phrase based methods is the query expansion where the user defined queries are automatically extended with additional search phrases improving the description of the users own interest profile.

The environment containing the information the user is looking for can also be very different: the most common space we search information in, is the world wide web. But company intranets, e-mail mailboxes, mailing list archives, or even peer-to-peer networks also require some sort of search mechanisms where the use of semantic information may significantly improve the quality of search results.

The methods proposed in this dissertation are based on keyword statistics and are designed for mobile peer-to-peer environment where the documents stored on various mobile devices are searched for by other mobile devices. Considering the limited resources available in such scenarios, the methods use easy-to-calculate algorithms and maintain low communication traffic to reduce communication costs and energy consumption. The search is based on the locally stored documents so

the user is only notified by the background searching process, if there is a probably interesting remote document available.

Of course, search among the locally stored documents is only allowed for documents marked public by the user. Furthermore, the user also has to be given the option to limit the number of parallel connections and transfer speeds in the final application, as these are essential requirements.

### Thesis structure

The following chapters are organized as follows:

- Chapter 1 introduces the topic of the dissertation, illustrates the motivations of the theses, and summarizes the main contributions.
- Chapter 2 is devoted to the description of related work proposed in the literature.
- In Chapter 3, the keyword selection algorithm is presented, the key method of the dissertation and the basis of the further theses as well.
- Chapter 4 introduces the search for similar documents and the document extension method aiming to improve the retrieval results.
- In Chapter 5, the two-level document topic identification technique is presented together with the serial cascaded similarity search aiming to increase the recall of the similarity search.
- Chapter 6 evaluates the new results, discusses the questions of application of the results, summarizes the theses and outlines the directions of future work.

## 1.1 Motivation

In the last years, mobile devices are getting more advanced which makes them suitable for many new applications. More-and-more content is stored on them (photos, music, e-books etc.) and the amount of content dynamically generated by the users is increasing rapidly. Nowadays, the rate of content generated this

way is estimated around 50% and this is expected to increase to 70% by 2012. The increase of content is also demonstrated by the appearance of desktop search applications on mobile devices. This essentially decentralized system has many advantages over centralized solutions: search and information exchange in peer-to-peer and ad-hoc networks is very scalable in storage capacity and transfer speed, can be energy efficient, and is very robust in overloading situations (like new years eve for mobile phone operators) or even catastrophic situations when the centralized infrastructure may be down entirely. The huge amount of information stored and exchanged this way requires new, advanced search and information management techniques.

Nowadays, there are many search engines also for mobile peer-to-peer networks, but usually, they provide phrased based search: the user has to define search phrases and the system can search for it in file names [Ekler et al., 2008], or in the contents. The work proposed in this dissertation is part of a project aiming to support automatic semantic search in mobile device environments [Forstner and Charaf., 2005]. The key concept is that the mobile device is running a background process performing automatic search for documents which might be of interest for the user. If one is found, the user is notified and asked, if it should be retrieved or not. The automatic search is based on the assumption that the user is interested in the topic of documents stored on the mobile device locally. This means that the search process has to compare the remote documents to the local ones and notify the user if the similarity is above a user-defined threshold. This threshold can be a setting like *many documents* (low threshold), or *strict similarity* (high threshold) for example.

The first important requirement for the system is related to communication traffic: as the communication with the remote devices may not be free of charge, the system should maintain a low communication traffic. This is achieved with special, very compact document representations that allow the comparison of document topics with the transmission of around 10-20 bytes. Beside the financial communication costs, another drawback of high traffic would be the significant energy consumption: a background process depleting the battery in hours while searching the peer-to-peer network for interesting documents would not be very popular.

The second important requirement is the low rate of misclassifications: the user should not be notified about off-topic documents too often, otherwise the system will not be used. From the theoretical point of view, this means that a high precision classification is required, even if the recall is lower: finding less interesting documents is much less annoying than finding many documents which are often not even related to the locally stored ones.

This dissertation is intended to provide the theoretical background for the automatic search for remote documents similar to the local ones. The low communication traffic is maintained using the compact document representation based on the presence or absence of very topic specific keywords. Strong topic specificity of the keywords supports the high precision results with priority over the recall.

## 1.2 Contribution

This section is intended to summarize the contribution proposed in this dissertation. The key idea of the proposed solution is the following: the document comparison uses a representation based on topic specific keyword lists. Every rough topic (like *animals*, *history*, and *traffic*) have a keyword list. The document representation consists of a topic identifier which identifies the topic. This already enables a limited level of similarity measurement, but its application in a fine-enough similarity comparison would require a huge amount of topics which would make the topic identification difficult. This is why the document representation also contains a bitmask where the bits indicate the presence or absence of the topic specific keywords in the document. Using this representation, the similarity of two documents is measured with the number of common keywords. The size of the representation is the size of the topic identifier (16 bits are believed to be sufficient) and 1 bit per keyword.

The simplicity of creating the document representation should be emphasized as it is performed by the mobile devices: the document is parsed and every word is compared to the keywords in the keyword lists. The topic with the most keywords in the document is selected and the bitmask (binary keyword presence indicator vector) is created with the keyword list of the selected topic.

The search for similar documents is a very simple procedure: the mobile device downloads the compact representation of remote documents and calculates the similarity, the number of common keywords, using only the compact representations. If it exceeds a user defined threshold, the user is notified. The user can decide wether the document itself should be downloaded or not. The keywords of the remote document are shown to the user to support this decision.

The similarity measure based on common topic specific keywords has two drawbacks: it cannot handle synonyms and hypernyms (generalizations of words), and if the documents contain few keywords - like in short texts or descriptions - it leads to a low recall as many related documents will not have any common keywords.

The first drawback, not being able to handle synonyms and hypernyms, is handled by the document extension. This process adds generalizations of the already present keywords to the documents. For example if a document about *hawks* and one about *dolphins* are compared, adding *animal* to both documents increases their similarity measure. For the learning of such generalizations, two approaches are proposed: one based on unsupervised, word co-occurrence based learning, and one based on WordNet.

The second drawback, the low recall, is partially solved by the document extension because it increases the number of keywords in the documents. In order to increase the recall further, a serial classifier ensemble type solution is proposed: if the user requires more hits, further levels of the similarity search can be requested. These are like additional classifiers which use different keyword lists and are specialized on the cases not recognized by the previous levels. It should be noted that the classification and selection expressions are both used for the similarity search because selecting documents similar to the local documents can be considered a 1-class classification.

The document topic identification method, selecting the topic with the most keywords in the document, requires every keyword list to be checked for the number of present keywords. In order to decrease the number of checked (and thus, locally stored) keyword lists, a two-level topic identification method, similar to a decision tree, is proposed: topics are ordered into topic sets which have their own keyword lists as well. This enables the mobile device to skip the check of some keyword lists and limit the search for the best matching topic in the promising topic sets.

The reason for our system not being exactly a decision tree is the following: the keyword list based topic identification may make mistakes due to the noise involved in natural language documents. A false decision inside a decision tree may make the correct classification impossible. To overcome this limitation, the topic sets are only triggering the check of their topics but not limiting the search on them. All topic sets trigger the check of their topics if their keyword lists have common keywords with the document, and topic identification is performed among the triggered topics. This way, the aim of the ensemble is to exclude hopeless topics from the identification procedure and not to strictly limit the number of checked topics by always restricting the decision to the best direction on a given level of classification. This solution allows robustness against misclassifications but still reduces the number of checked keyword lists.

Chapter 2

# Related work

This chapter is intended to give a brief overview of proposed techniques relevant to the dissertation. First, the commonly used bag-of-words document representation approach and some of the many dimensionality reduction techniques are introduced. After these, various extensions aiming to add semantic information to the representations are discussed, and an overview on the most common document distance measures is presented. Based on these distance measures, some important techniques for clustering and classification are presented, together with classifier ensemble creation aiming to further improve the performance of classifiers. Finally, some special properties of mobile device environments, some standard evaluation document corpora, and related research areas in Hungary are shown.

## 2.1 Representation and dimensionality reduction

The most common document representation approach is the vector space model[Salton, 1987][Dubin, 2004] which represents the documents as vectors in a feature space. Usually, the features are initially the possible words and the coordinates represent the relevance [Weiss et al., 2005] of the given word to the given document. This is called the bag-of-words approach as the appearance order of the words is not taken into account.

Using the vector space representation of the documents, the number of features is usually huge. As most of the features are redundant or useless, many dimensionality reduction techniques have been proposed. There are two main approaches: feature selection and feature extraction. Feature selection aims to select some of the features and discard the remaining ones, feature extraction on the other hand aims to create new features by merging information from original features. An overview on common approaches is presented in [Mladenić and Grobelnik, 2003].

## Feature selection

Feature selection techniques aim to select a subset of the features to reduce the number of dimensions, avoid unnecessary noise, and prepare the comparison by emphasizing the important features. The simplest methods are based on information gain [Li and Chou, 2002], mutual information content between the document label and the words appearance, the words entropy [Garner and Hemsworth, 1997], or TFIDF (term frequency, inverse document frequency) type weighting schemes[Salton et al., 1975]. More sophisticated methods use for instance angular measures considering the rate of occurrences of words inside and outside the given target topics [Combarro et al., 2006]. Other proposed methods are based on the Kullback-Leibler divergence [Dobrokhotov et al., 2003][Büttcher and Clarke, 2006], or the Optimal Orthogonal Centroid Feature Selection [Yan et al., 2005]. Further techniques are the Lasso method [Tibshirani, 1996] or techniques employing artificial neural networks [Zvi Boger and Shapira, 2001]. In [Keerthi, 2005], a generalization of least angle regression for feature selection is proposed, [Carmel et al., 2001] proposes index pruning techniques based on the relevance scores of the words, and [Guo, 2008] presents a feature selection method based on document frequencies. For selecting features characterizing a given set of documents, [Lagus and Kaski, 1999] and [Azcarraga et al., 2004] propose suitable keyword selection techniques.

In cases where additional information are available, further improvements can be achieved: if statistics on query term usage frequencies can be accessed, one can select the most frequently used query terms as important features [Kwok, 1996]. If the documents are for example scientific papers and the abstract can be located,

one can assume that words often appearing in abstracts are more important than other ones [Bhowmik, 2008].

Keyword selection usually involves the removal of stopwords, words that do not hold any useful information about the topic of the document. Beside the many stopword lists commonly available, [Sinka and Corne, 2003] proposes a method for optimizing stopword lists using k-means clustering.

## Feature extraction

Beside feature selection, feature extraction is another way of dimensionality reduction. It aims to create new features based on the original ones and transforms the documents into the space of the new features. Many common approaches are based on Latent Semantic Analysis (LSA) which employs singular value decomposition (SVD) of the term-document matrix [Furnas et al., 1988] [Deerwester et al., 1990] [Ando, 2001]. As SVD creates a low-dimensionality feature space with predefined number of dimensions, the optimal number of dimensions is an important question [Dupret, 2003]. In [Efron, 2008], the Rocchio relevance feedback and LSA are compared as two well-known techniques for improving the vector-space-model, and an important generalization of LSA to support multi-type objects is proposed in [Wang et al., 2006]. Further possibilities are examined in [Yu, 2004] and the operation principles of spectral methods (like LSA) is investigated in [Brand and Huang, 2003].

Alternatives to LSA are based on polynomial filtering [Kokiopoulou and Saad, 2004] to decrease resource needs, Probabilistic Latent Semantic Analysis [Hofmann, 2001] based on expectation maximization using a generative latent document class model, and Least Angle Regression [Efron et al., 2002] which involves the learning method into the feature extraction process instead of creating the feature set completely independently of the methods actually using the new features.

Different approaches of feature extraction are based on Lexical Chains [H. Gregory Silber, 2002] [Ercan and Cicekli, 2007], graph-mining techniques [Turenne, 2003], and the method proposed in [Toutanova et al., 2004] is based on estimated word dependency distributions using Markov Chains and random walks.

## Adding semantic information

As described previously, feature extraction aims to create new features based on the original ones, and feature selection aims to select a subset of the original features. A significant improvement can be achieved if the representation of the documents in the new feature space is capable to take semantic information into account. The most common example is the case of synonyms: difference of documents caused by using different, but synonym, words can be avoided by adding the synonyms to the representation. Similar improvements can be achieved by using hypernym (generalization) relationships. Many proposed methods, including the one proposed in this dissertation, use the hypernym graph of WordNet [Miller et al., 1990] [Edmonds, 2007] [Du et al., 2007] [Schönhofen, 2008] [Snow et al., 2004] for such purposes. An interesting mixture is proposed in [Termier et al., 2001] which rewrites the documents by replacing words with concepts retrieved using LSA and WordNet. The representation of documents in the space of concepts is used in [Schönhofen and Charaf, 2004] too.

Beside the co-occurrence, frequency, and similar statistics of the words, natural language processing techniques are also used to improve the feature selection and extraction. These include for instance stemming like the inflectional stemming [Weiss et al., 2005], Porter stemmer [Porter, 2006], LSA based morphology learning methods [Schone and Jurafsky, 2000], and part-of-speech tagging [Brill, 1992]. The Leximancer system [Smith and Humphreys, 2006] aims to transform lexical co-occurrence information of natural language documents into semantic patterns with an unsupervised learning. Automatic annotation of simple grammatical relationships of words using classification methods is proposed in [Pradhan et al., 2004]. Similar classification methods are used in [Taskar et al., 2004] to improve parsing of natural language documents. In [Chowdhury and McCabe, 1998], a user query based information retrieval system is proposed which is enhanced with part-of-speech tagging to find the most relevant parts of the texts.

Beside WordNet, ontologies [Zú 2001] [Tun, 2006] [Choi et al., 2006] are also often used as a source of semantic information. Many works propose methods to support the creation of such ontologies. These methods use document markups [Kozlova, 2005], external ontologies [Novácek et al., 2007], dictionaries or vocab-

ularies [Maedche and Staab, 2001], terminology descriptions [Hamon et al., 1998], LSA methods [Fortuna et al., 2006b] [Fortuna et al., 2006a] or syntactically similar parts of documents [Bloehdorn et al., 2006]. Given multiple ontologies, these can be mapped into each other [Tang et al., 2006] [Mocan et al., 2006] in order to describe information in other ontologies. In [Sabou et al., 2006], an overview on ontology selection and evaluation techniques is presented, as the selection of the best ontology to use is not always a trivial decision. A case study about combining ontologies and document retrieval in an E-learning environment is presented in [Baumann et al., 2002]. A method for creating description profiles for disease representation to support information retrieval is presented in [Wedemeyer and Srinivasan, 2003].

## 2.2 Distance measures

One of the most important questions in classification and clustering is the applied distance measure. The most common distance measure in document classification is the cosine distance measure [Weiss et al., 2005], but there are many other sophisticated distance measures proposed in the literature. Earth Mover's distance, BM25, Jaccard and Dice are described in [Wan, 2007]. Further measures are edit distance [Cormode and Muthukrishnan, 2007] [Batu et al., 2003] and a measure based on common substructures in web documents [Flesca et al., 2007].

Similarity - or distance - can also be measured using semantic distances of words which can be retrieved from Wordnet [Jensen et al., 2008], or by using principal component analysis run on a dictionary of the target language [Kozima and Ito, 1996]. Probabilistic correlation models are also common choices for similarity measurement [Jia and Peng, 2007].

Similarity of shorter texts not having enough words for statistical measures can also be measured by using them as a search query and comparing the search results [Sahami and Heilman, 2006].

Phrase based similarity is a natural choice too: distance measures based on suffix trees [Chim and Deng, 2008], Document Index Graphs [Hammouda and Kamel, 2002] [Hammouda and Kamel, 2004], or word-to-word correlation factors [Lee and Ng, 2007] taking also the distance between appear-

ances of words in the text into account. In [Cooper et al., 2002], a method measuring document similarity based on the presence of common, relevant keywords is proposed. These methods are similar to the one proposed in this dissertation.

A special similarity measurement based on movie descriptions is proposed in [Fleischman and Hovy, 2003] where movies are considered to be similar if they are both similar to the same type of other movies according to their descriptions and classification.

In an on-line system, the system can observe the queries issued by the users and the top-ranked documents returned by the search system. Based on sets of documents top-ranked for the same query, the document similarities can be kept up-to-date [Na et al., 2007].

Kernel function based methods like support vector machines have successfully been used for text categorization. A standard choice of kernel function [Lehmann and Shawe-Taylor, 2006] has been the inner product of the vector-space representations of two documents, but there are many further kernel methods proposed in the literature like the ones taking string properties [Lodhi et al., 2002] or latent semantic information [Aseervatham, 2008] [Cristianini et al., 2002] based on LSA into account.

## 2.3 Clustering

Clustering [Jain et al., 1999] of documents means creating groups without prior knowledge of the classes to which the documents have to be assigned. One of the most common clustering algorithms is k-means [Oded Maimon, 2005] [Sevillano et al., 2006] and its variants like k-medoids [Oded Maimon, 2005], or constrained k-means [Wagstaff et al., 2001] which is capable to handle background knowledge too. Probabilistic approaches are based on non-negative matrix factorization [Farial Shahnaz and Plemmons, 2006] [Lee and Seung, 1999], distributional clustering [Niall Rooney and Dobrynin, 2006] and Independent Component Analysis [Isbell and Viola, 1999] for example.

Document clusters can be considered to represent topics, so hierarchical clustering methods [Pons-Porrata et al., 2007] can also be used to create topic hierarchies

[Tao Li and Ogihara, 2007] automatically. (It should be noted that the two-level topic identification proposed in thesis III. has similar goals.)

In [Chih-Ping Wei and Hsiao, 2008], a clustering method is proposed which is capable to incorporate the users prior, partial clustering. This allows the personal preferences to be taken into account. Another method for incorporating prior knowledge in hidden markov methods is presented in [Grenager et al., 2005].

Double clustering [Slonim and Tishby, 2000] is a widely used clustering method which first clusters the words and then clusters the documents in the space of word clusters. It has an iteratively improving version [El-Yaniv and Souroujon, 2001] designed to further decrease the noise.

The self-organizing-map approach is incorporated in [Lagus et al., 2004] which maps the documents into an easy-to-visualize two dimensional space where mutually similar documents get near each other.

Further clustering algorithms are presented in [Banerjee, 2005] and [Strehl, 2002], and an objective evaluation criterion for clustering is proposed in [Banerjee and Langford, 2004].

## 2.4 Classification

Document classification aims to estimate the class where a document belongs. There are many well-known classification methods [Qi and Davison, 2009] and often, the main difference between applications is the distance measure underlying the classifier method. Special cases are the 1-class classifiers [Hempstalk et al., 2008] which aim to select documents of a given kind and not do anything with the remaining ones. Typical classification methods are summarized in [Kotsiantis, 2007]: decision trees (DT), rule learning, single layer perceptron, multi-layer perceptron (MLP), radial basis function (RBF) networks, linar discriminant analysis (LDA), Naive Bayes classifier, Bayes networks, instance based learning (like nearest neighbor), and support vector machines (SVM). A generalized version of the naive bayes classifier, optimized for binary classification problems, is presented in [Larsen, 2005]. Dynamic bayesian networks applied for many statistical linguistic applications are investigated in [Peshkin and Pfeffer, 2003]. Hierarchical classification techniques are presented in [Chen et al., 2005].

Information retrieval (IR) [Salton, 1987] [Singhal, 2001] [Moffat et al., 2005] [Thompson, 2008] consists of many areas like document classification, document similarity measurement, and question answering. The search for similar documents is mainly a classification task as well. The goal in to find documents which are similar to a given set of documents, user defined queries, or other representation of interest. The key question in such applications is the distance measure. A set distance based document similarity measurement used in peer-to-peer networks is presented in [Wang and Yang, 2006]. In [Jamali et al., 2006], a web crawler incorporating link structure into document similarity measurement is presented. An application of document similarity for automatic pre-fetching of web documents is presented in [Xiao, 2005]. A system observing common interest of users, based on document similarities, in order to support collaborative web browsing is presented in [H. Lieberman and Vivacqua, 1999]. [Silva and Martins, 2003] proposes a system helping users formulating query phrases by automatically creating a word cluster hierarchy. An algorithm to calculate all the pairwise similarities is proposed in [Roberto J. Bayardo, 2007].

A system aiming to find only a subset of a large document collection which contains the relevant documents is proposed in [Blair, 2002]. This can be achieved with both less-specific queries, and precise queries identifying documents surely outside the set of relevant documents. This second strategy is incorporated in this dissertation too, in connection with serial similarity search cascades.

A special, focused type of information retrieval is the task-based IR [He et al., 2008] [Liu et al., 2003] [Marchionini, 2006] which aims to support exploratory information search focusing on a given topic. Such a task could be for example the retrieval of every available information about a given criminal case.

Important areas of text classification are E-mail specific applications which usually aim to support e-mail management and spam filtering [Moreale and Watt, 2003], and sometimes even e-mail thread summarization [Wan and McKeown, 2004].

More specialized applications are possible, if the documents are semi-structured, like XML documents. The differences in such environments are summarized in [Fuhr, 2004]. Similar problems arise if other than textual information, like both text and images, are also to be re-

trieved [M.T. Martín-Valdivia and Ureña-López, 2008]. The ALFA system [Vailaya et al., 2005] aims to provide a user interface representing heterogenous biological information for information retrieval.

There are also very different document representation approaches based on logical expressions [Losada and Barreiro, 2006] and databases [Lacroix et al., 1998] which allow querying the corpus using different query representations like logic expressions or database query-like expressions.

Evaluation of classification methods is usually based on well-known measures like precision, recall, and F-measure [Weiss et al., 2005]. The evaluation measures average-precision and R-precision are compared in [Aslam et al., 2005], both approximate the area under the precision-recall curve. The influence of search engine performance on usability is discussed in [Liaw and Huang, 2003].

Further details about the methods, and about alternative approaches not described here, can be found in standard IR textbooks like [Baeza-Yates, 1999], [Belew, 2000], [Tikk, 2007], and [Ferber, 2003]. An overview of data mining technologies is presented in [Bodon, 2009].

## 2.5 Classifier ensembles

Classifier ensembles aim to improve the classification performance by employing multiple, even weak, classifiers and combining their results with a consensus function into a stronger classifier. An overview on classifier ensembles like decision list and decision tree-like ensembles can be found in [Oded Maimon, 2005].

One of the most commonly used ensemble techniques is boosting, where the classification result is retrieved using a weighted sum of weak classifiers. The weight of the individual classifiers are set according to their classification performance, and misclassified documents get higher weight during the training of the next classifier of the ensemble [Freund and Schapire, 1995]. An application of boosting for semantic web technologies is presented in [Bloehdorn and Hotho, 2004]. A successful application of boosting to improve SVM performance is presented in [Dong and Han, 2005].

## 2.6 Query and document extension

In order to improve the classification results based on user queries, query or document extension (also called document expansion) is often used. Query expansion means that the query entered by the user is extended with additional terms to improve the retrieval results. Document expansion means that the documents are extended with additional terms to increase their relevance to a set of queries.

The main difference between query expansion techniques is the source of information which the selection of additional terms is based on. Additional terms can be selected from queries resulting in the same documents as the one defined by the user [Billerbeck et al., 2003] or the documents retrieved with the initial query [Vechtomova et al., 2003]. Word-document relationships can be retrieved using random indexing [Sahlgren and Karlgren, 2002], or from user relevance feedback [Hsi-Ching Lin and Chen, 2006], as the user selects the really relevant documents from the search results. Thesauri can also be used to add related words to the query terms [Qiu and Frei, 1993], or the co-occurrence statistics of words in a corpus, like the proposed method in thesis II., can be observed to retrieve word relationships [Bai et al., 2005]. If document summaries are available, further relevant words can be extracted from the summaries as well [Lam-Adesina and Jones, 2001].

Similarly to query expansion, there are proposed methods aiming to substitute some of the query terms to more topic specific ones. In [Jones et al., 2006], better query terms are retrieved by observing the queries a single user issues during a day. In order to decrease the cost of query evaluation, query pruning techniques [Anh and Moffat, 2006] are also employed which remove some query terms to decrease processing time but still provide good-enough results.

Personalized web search can also be supported by extending the queries of the users. The personal information can be retrieved for example from the search history of the user [Liu et al., 2004], from the PC desktop which is a rich source of information about personal interests [Chirita et al., 2006], or from the set of (hyperlinked) web documents visited by the user. The system proposed in this dissertation uses the set of locally stored documents as a source of user interest profile information.

Document extension [Tao et al., 2005] [Tseng and Juang, 2003] aims to improve the retrieval results from the document's side: additional terms are added to the documents to improve their relevance to some given set of queries. In [Amitay et al., 2005], the proposed system observes query reformulation sessions of individual users and extends the documents so that there is less need to reformulate the queries, as the desired results are retrieved without successive tries with reformulated queries. [Kazem Taghva and Condit, 2004] investigates the usefulness of keywords manually added to documents in order to improve the search for information about a huge project, as the authors want the users to be more informed, and want to help them in the search for documents relevant to their inquiries. A document extension technique is proposed in thesis II. as well.

## 2.7 Mobile devices

In the last years, more and more mobile device applications appear in the information society. The information retrieval tasks involving mobile devices have some special requirements, like the limited energy storage (continuous transmission depletes the batteries in hours [Hurson et al., 2006] [J.K. Nurminen, 2008]) and the communication traffic is not always free of charge which makes the maintenance of low communication traffic necessary [Zhu and Mutka, 2008]. Location-awareness [Cook and Das, 2007] [Göker and Myrhaug, 2008] and peer-to-peer applications [Delmastro et al., 2008] [Luo et al., 2007] also have to take these limitations into account. Similarly to these limitations, many applications are proposed to overcome the limitations of the smaller displays too, for example by employing text summarizing techniques [Otterbacher et al., 2008].

## 2.8 Datasets

Document processing techniques are usually evaluated on well-known document corpora like the 20 Newsgroups [Lang, 1995], the Reuters Corpus Volume 1 [Lewis et al., 2005], the Ohsumed corpus [Hersh, 1994], the Enron data set [Klimt and Yang, 2004] created for e-mail classification evaluations, or the data sets of TREC (Text Retrieval Conferences).

WordNet [Miller et al., 1990] is often used as a starting point of semantic analysis between words and concepts. The WordNet database organizes words into sets of synonyms (synsets), each of which represents an underlying concept and links these through semantic relations like hypernyms and hyponyms. The current version 2.0 of WordNet comprises a total of 115,424 synsets and 144,309 lexical index terms.

## 2.9 Related research in Hungary

In Hungary, some of the important research areas related to this dissertation are web spam filtering [Benczúr et al., 2009], creating a Hungarian version of WordNet [Prószéky and Miháltz, 2008], and other linguistic resources [Halácsy et al., 2004] [Szarvas et al., 2006] for Hungarian. [Iván and Ormándi, 2007] proposes a sentence parsing technique using support vector machines, and [Dudás, 2006] describes a system aiming to learn the morphological structures of the Hungarian language with statistical methods. [Tikk et al., 2005] introduces new approaches for searching those parts of the web which are not accessible through conventional search engines, and [Benczur et al., 2006] proposes techniques for the integration of knowledge from different domains. A method for detecting interpersonal relations using language processing techniques is presented in [Pohárnok et al., 2007], and an application of controlled natural languages for agent interfaces is introduced in [Mészáros and Dobrowiecki, 2009]. Many further information management approaches and technologies are described in [Magyar et al., 2007].

## 2.10 Summary of relationships between the proposed methods and related work

There are many techniques in the related literature which are similar to the ones proposed in this dissertation. In the following, these similarities are summarized.

- Document representation: the proposed methods represent the documents using the very common bag-of-words approach which does not consider the order of words in the document. The document models usually employ some

kind of relevance measure to indicate the importance of a word in a document. Due to limitations on the representation size, the proposed methods use only binary relevance, indicating the presence or absence of words. The special, precision oriented feature selection allows the representation to remain effective, even after this simplification.

- Feature selection: the proposed feature selection technique (Precision based Keyword Selection, defined on page 26) introduces a ranking of the words with a precision-related measure (individual precision, defined on defined on page 24). This allows the keyword selection to introduce a precision related constraint and thus have a priority towards high precision. Similar methods in the literature are the feature selections using TFIDF approaches or the mutual information between the word appearances and the class labels of the documents. Baseline measurements presented on page 48 compare the PKS algorithm to the mutual information based feature selection.

- Distance measures: the proposed distance measure is derived from the precision related property of keywords which makes it simple, and different from the common methods like the cosine distance. The normalization in the distance measure was omitted because common keywords indicate common topic with high precision which makes the rate of common keywords less relevant.

- Classification: the classification method proposed in this dissertation takes advantage of the special keyword properties, too. Commonly used classification method with similar complexity is the naive bayes classifier, to which the proposed method is compared to in the baseline measurements presented on page 48. The proposed method is related to maximum likelihood decision too, but the employed probabilities are only estimated lower bounds (under some assumptions), which makes it unsuitable for exact likelihood estimations. Exact information about the relationships of keywords are not retrieved due to complexity and storage size considerations.

- Using semantic information: the techniques proposed in thesis II. are related to document expansion techniques. The main difference is that the compact

document representation limits the range of keywords which can be added to a document, so a suitable learning method is proposed, together with a technique to employ information retrieved from WordNet.

- The classifier ensemble technologies proposed in thesis III. are related to decision trees and decision lists. The two level topic identification is similar to a "soft" decision tree, where the search for the final classification is not limited to the best matching subtree. The cascade structures are related to decision lists, but they employ further 1-class classifiers to remove negative cases. This allows better training possibilities for the further levels of the cascade structure.

Probabilistic approaches to problems similar to the ones addressed in this dissertation are discussed in [Goutte and Gaussier, 2005], where a probabilistic interpretation of precision, recall and F-measure are presented using gamma- and beta-distributions. These results provide an important basis for further research related to the technologies proposed in this dissertation.

Chapter 3

# Keyword selection

This chapter describes the keyword selection method used for the document topic identification and representation. The proposed method is a feature selection technique aiming to find a set of suitable words for the representation of documents of a given target topic. This set is called the keyword list of the topic and the selected words are called the keywords. Due to the nature of the keywords, their presence will indicate that the document probably belongs to the target topic, and thus they will allow a simple but effective topic based classification as well.

Selecting the features used for classification and topic comparison is a very important step which requires special attention. As the documents are represented in the space of words, feature selection means creating the set of keywords.

The aim of the keyword selection is to select a keyword list which allows easy recognition of a document's topic and compact representation of a given document. The key idea of the compact document representation, described on page 52 in details, is to store only the presence/absence of the keywords in the keyword list of the documents topic. As keywords of other topics cannot be indicated this way, the compact representation is created using the topic having the most keywords in the document. This is the most keywords (MKw) classification. So the main question is the following: in how far can a given word contribute to the topic identification using this classification (MKw) and representation method.

As the keyword lists alone are suitable for application as a 1-class classifier aiming to select the documents belonging to the topic of the keyword list, in the following discussions, the term *selection* and *selector* will be used as synonyms of

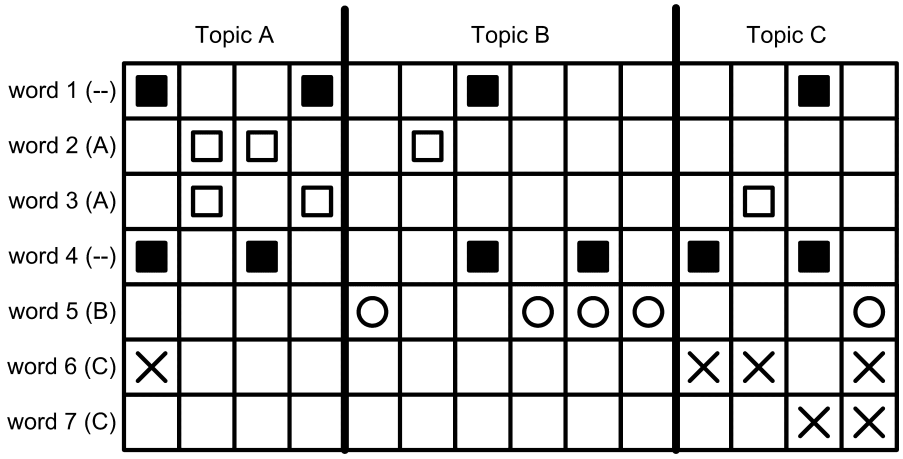


**Figure 3.1.** Keyword based selection. The topic of topic-specific keywords is given in brackets.

1-class classification and 1-class classifier, in order to emphasize the difference to a non-1-class classification.

## 3.1 Precision based keyword selection

The keyword based document selection is illustrated in Fig. 3.1. In the word-document matrix, rows represent words and columns represent documents. In this example, solid rectangles indicate the presence of two general words not specific to any topic. Other marks indicate the presence of words specific to one of the topics. If a document contains such a topic-specific word, the document belongs to the topic of the word with high probability. For example if a document contains a word marked with circle, the document is likely to belong to topic B. The proposed keyword selection technique aims to identify such topic-specific words for a given target topic.

In a real-world application, this algorithm requires a labeled training document set covering all topics the trained system is supposed to handle. For common topics, many document collections suitable for the training are commonly available, like the ones used in the section of experimental results.

### 3.1.1 The Precision based Keyword Selection Algorithm

The Precision based Keyword Selection (PKS) algorithm creates a keyword list $K_T$ for a given target topic $T$. Its input consists of labeled document vectors and the identifier of the target topic. The output is a list of keywords. In the following discussions, topics are handled as sets of documents and documents are handled as sets of words.

**Definition 3.1** (Document set selected by a keyword or keyword list)**.** The document set $S(w)$ selected by a keyword $w$ is the set of documents containing the word $w$: $S(w) = \{d \in D : w \in d\}$ where $D$ is the set of all documents. Similarly, for a $K$ keyword list, $S(K) = \{d \in D : K \cap d \neq \emptyset\}$.

Given a set of selected documents $S$ for a target topic $T$, the common measures precision, recall and F-measure can be used to evaluate the result. If $c = |S \cap T|$, $t = |T|$, and $f = |S \backslash T|$, then precision is $c/(c + f)$, recall is $c/t$ and F-measure is the harmonic mean of precision and recall.

A brute force method for keyword selection would be to check every possible keyword set and choose the one maximizing F-measure on the training document set, that is, for which $S(K_T)$ is maximizing F-measure. As this is computationally very hard to perform, the following trick is introduced: keywords are ordered according to their contribution to high precision, which is easy to estimate, and beginning with the best ones, words are added to the keyword list in a greedy way until an optimal F-measure is reached.

The PKS algorithm is a parameterless algorithm. The most important feature of the created keyword list $K_T$ is that if a document contains a keyword from it, then the document belongs to the target topic $T$ with high probability. A key concept of the PKS algorithm is the individual precision of words.

**Definition 3.2** (Individual precision, recall and F-measure)**.** Individual precision $iprec(w, T)$, recall $irecall(w, T)$ and F-measure $iF(w, T)$ of a word $w$ are the precision, recall and F-measure of $S(w)$ with respect to the target topic $T$.

The individual precision can also be interpreted as the estimated probability that the document belongs to topic $T$ if the word $w$ is present in the document.

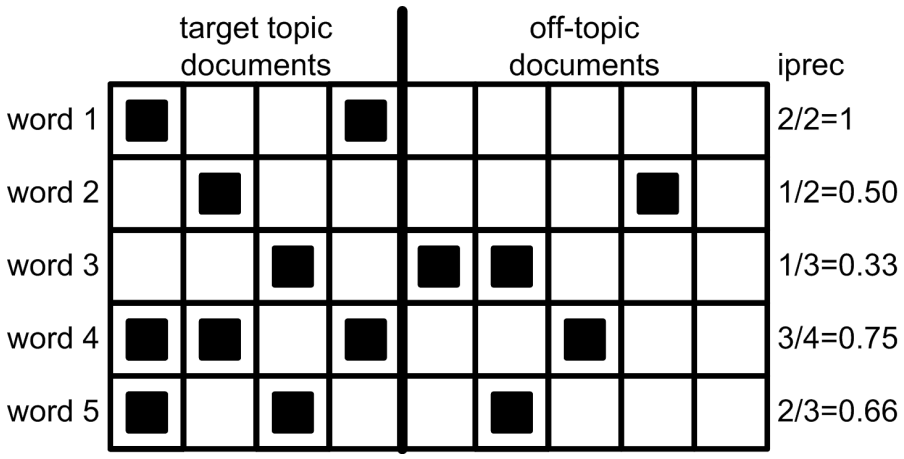


**Figure 3.2.** Individual precision of words with respect to a given target topic.

Formally: $iprec(w, T) = \hat{Pr}(d \in T | w \in d)$. A word having high individual precision is present almost only in the documents of the target topic which makes its presence suitable for topic estimation. Individual precision is illustrated in Fig. 3.2.

#### 3.1.1.1 The optimization on F-measure

The PKS algorithm creates a keyword list $K_T$ using a minimal individual precision limit defined as follows:

**Definition 3.3** (Minimal individual precision limit $mp_T$ of topic $T$)**.** The minimal individual precision limit $mp_T$ of topic $T$ is the lower limit for individual precision of the keywords of topic $T$. Formally,

$$w \in K_T \leftrightarrow iprec(w, T) \geq mp_T \tag{3.1}$$

The lower bound of individual precisions in the keyword list, expressed by $mp_T$, is the most important property of the keyword lists created by the PKS algorithm. PKS optimizes $mp_T$ to maximize the F-measure of the $S(K_T)$ selection using the resulting keyword list.

**Definition 3.4** (Precision based Keyword Selection)**.** The PKS algorithm is defined as presented in Algorithm 3.1. Given a $T$ target topic and a set of U off-topic documents, it returns a keyword list containing all words above the $mp_T$ mini-

mal individual precision limit. The value of $mp_T$ is optimized to achieve maximal F-measure with the keyword list.

**Algorithm 3.1** Precision based Keyword Selection

Input: $T$ target topic
**for** $x = 1$ to 0 step $-0.01$ **do**
  $K(x) = \{w \in W : iprec(w, T) \geq x\}$ // Create new keyword list for $x$
  $p(x) = \text{precision}(S(K(x)), T)$ // Calculate its precision with respect to $T$
  $r = \text{recall}(S(K(x)), T)$
  $f(x) = \text{fmeasure}(p, r)$
$mp_T = \arg\max_x \{f(x)\}$
$ep_T = p(mp_T)$
$K_T = K(mp_T)$
Output: $K_T$, $mp_T$, $ep_T$

The minimal individual precision limit $mp$ represents a balance between high precision and high recall, but high precision is maintained while F-measure is optimized. This gives high precision a priority over the high recall. The PKS algorithm also returns the $ep_T$ estimated precision of the keyword list which was measured in the training document set using all keywords together.

Fig. 3.3 presents an example on the curves of precision, recall and F-measure in PKS, as a function of $x$.

The resulting keyword list of PKS for topic $T$ satisfies the following important equation which describes that a word is a keyword exactly if it has an individual precision not lower than the $mp$ of the topic. In the probabilistic interpretation of the individual precision,

$$w \in K_T \leftrightarrow \hat{Pr}\left(d \in T | w \in d\right) \geq mp_T \tag{3.2}$$

**Remark**: the maximum of F-measure always exists, in the worst case at the borders of the $x = [0..1]$ interval. Very high $mp$ often leads to empty keyword list and thus 0 recall (and precision is considered to be 0 as well in this case). On the other hand, $mp = 0$ makes every word a keyword and thus recall is 1 and precision is the a priori probability of the target topic as all documents in the data set are selected.

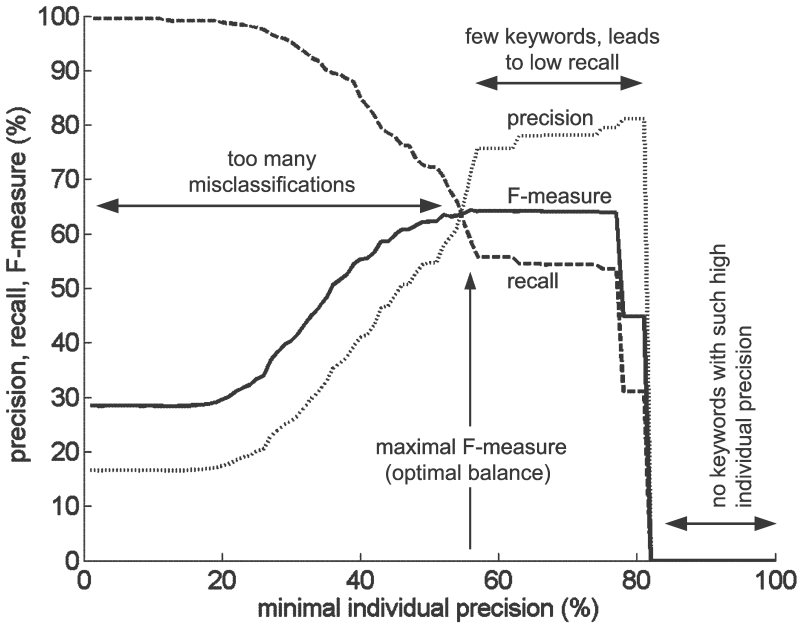


**Figure 3.3.** The PKS algorithm collects keywords starting from $x = 100\%$ minimal individual precision limit and decreasing it until 0%. After the maximum point of F-measure is reached, the decreasing precision (due to increasing number of keywords with lower individual precision) is no longer compensated by the increasing recall (due to the increasing number of keywords covering more and more documents).

### 3.1.2 Precision of document selection

By taking advantage of the nature of the document collections, I will make an important assumption about the behavior of the words. This assumption allows significantly simpler proofs of propositions about the proposed system. The validity of the assumption and the propositions in the real-world application are supported by several presented experimental results.

**Assumption 3.5** (Precision of keyword lists)**.** *Given a $K_T$ keyword list with the property described by Eqn. 3.2, I assume that every $W \subseteq K_T$ subset has a precision not lower than $mp_T$ too, when used for document selection: $\hat{Pr}(d \in T | W \cap d \neq \emptyset) \geq mp_T$.*

This assumption states that keywords do not decrease each other's precision below the *mp* value of the keyword list, so that *mp* can be used as a lower bound for expected precision, even if multiple keywords are present in a document.

Although this assumption is not always valid, experimental results show that due to the nature of the natural language document sets and the selected keywords, the rate of situations when the assumption is not valid, is very limited and can be omitted in real-world applications. The experimental results supporting this assumption are presented together with the other experimental results related to keyword selection.

Using this assumption, the $mp_T$ value is considered to be a lower bound for the expected precision of the document selection using the keyword list $K_T$. As the $mp_T$ value is a lower bound, it is mainly used in theoretical aspects. Another precision estimation is provided by the $ep_T$ expected precision returned by PKS, but as the experimental results show, $ep_T$ is usually a pessimistic estimation too.

### 3.1.3 Precision of keyword lists with same size

The following proposition emphasizes the quality of the keyword lists created by PKS among the possible keyword lists with the same size.

**Proposition 3.6** (precision of keyword list)**.** *The keyword list created by PKS allows the highest mp value among the keyword lists with the same size.*

The importance of this proposition is given by the $mp$ value being a lower bound for expected precision, according to assumption 3.5.

**Proof:** This proposition can be proven by assuming that assumption 3.5 is valid. The only possible keyword list with $|K_T|$ words and $\forall_{w \in K_T} iprec(w, T) \geq mp_T$ is $K_T$. The lower bound of expected precision, which is $mp_T$ according to the assumption, could only be increased by replacing a word to another word having higher individual precision. But according to Eqn. 3.1, all such words are already in $K_T$, so the improvement of $mp_T$ through replacing one or more words is not possible, only by changing the $|K_T|$ number of keywords, too.□

### 3.1.4 Linear execution time

The execution time of the PKS algorithm is linear with respect to the size of the feature-space representation of the documents in the training document set. This makes it applicable in mobile devices too, although making the keyword list

creation distributed and running on the mobile devices themselves is subject of further research.

**Proposition 3.7** (linear execution time of PKS)**.** *The execution time of PKS is* $\Theta(k \cdot n)$ *where* $k$ *is the number of words and* $n$ *is the number of documents in the training set.*

**Proof:** The transformations and data structures mentioned in the proof are illustrated in Fig. 3.4. PKS checks the F-measure for the various $x$ values between 0% and 100% in 1% steps. There are $m = 101$ iterations. The algorithm consists of the following steps:

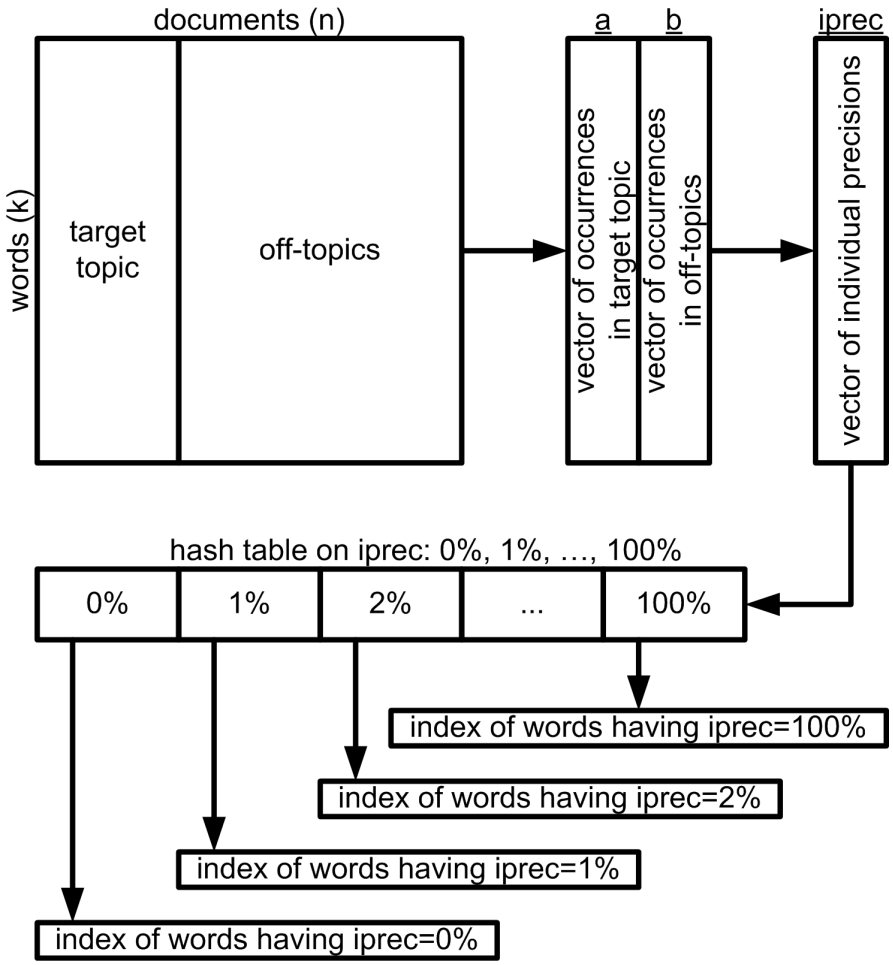


**Figure 3.4.** Linear execution time of PKS.

1. Creating two topics by merging all target and all off-topics by selecting documents of target and off-topics ($\Theta(n)$ decisions). The merge itself is a sum

of the corresponding (binary) document vectors ($\Theta(k \cdot n)$ steps). If the document vectors are not binary yet, the conversion can be performed within these $\Theta(k \cdot n)$ steps too. The resulting two vectors are $\underline{a}$ (sum of target topic documents) and $\underline{b}$ (sum of off-topic documents). Altogether, it is $\Theta(k \cdot n)$ steps.

2. Calculation of individual precisions: $iprec(w, T) = a_w/(a_w + b_w)$ which is $k$ addition and division. Altogether $\Theta(k)$ steps.
3. Creating a hash table for storing words for every possible *iprec* value (quantized on 1% slices). By using linked lists for the buckets and inserting new elements before the first element in the list, inserting $k$ words requires $k$ steps, assuming that jumping to the corresponding bucket requires one step. Altogether $\Theta(k)$ steps.
4. Creating a list of documents not covered by the keyword list. Initially, no documents are covered, so every document is contained in the *uncovered document list*. Initialization requires $\Theta(n)$ steps. Additionally, two counters are initialized for covered target and off-topic documents, both set initially to zero. Altogether $\Theta(n)$ steps.
5. For every $x$ value, that is, for every $m$ iterations: get words for the current $x$ value from the hash table and check coverage of previously uncovered documents (retrieved from the *uncovered document list*). Update counters for covered target and off-topic documents and remove newly covered documents from the *uncovered document list*. Update precision, recall and F-measure. During the iterations, every word-document pair is checked no more than once, which is $\Theta(k \cdot n)$.
6. Search for the maximal F-measure value: as there are 101 iterations, this requires $\Theta(1)$ steps.
7. Finally, all words over optimized *mp* value have to be collected, which requires $\Theta(k)$ steps.

The total number of required steps is $\Theta(k \cdot n + k + k + n + n \cdot k + 1 + k) = \Theta(k \cdot n)$. It should be noted that even reading the word-document matrix requires $\Theta(k \cdot n)$ steps.□

**Remark**: Stopwords usually cannot be keywords because of their low individual precision which makes stopword removal unnecessary. Despite of this observation, a minimal value for *mp* can be easily defined not to allow the minimal precision limit to have too low values.

## 3.2 Classification: The Most Keywords method

From the classifications point of view, a keyword list created by the PKS algorithm is a trained 1-class classifier: it is capable to select documents of its target topic and there is a lower limit for its expected precision as well. If there is a need to transfer a 1-class classifier selecting documents of a given topic, the transmission of the keyword list of the topic is sufficient. This makes the classification method simple, but still effective, because PKS has already created 1-class classifiers.

### 3.2.1 Classification method

The keyword lists created by PKS can be used to identify the topic of documents in the following: the *Most Keywords* (MKw) classification method selects the topic having the most keywords in the document:

$$\hat{\mathcal{T}}(d) = \arg\max_{T} \{|d \cap K_T|\} \tag{3.3}$$

where $\hat{\mathcal{T}}(d)$ is the estimated topic of the document $d$.

The most important reason of choosing this classification method beside its simplicity is that the compact representation of the documents (defined on page 4.1) should contain as many keywords as possible. This leads to a classification method choosing the topic having the most keywords in the document, as only the keywords of the document's topic can be indicated in the compact document representation.

## 3.3 Separability estimation

The PKS algorithm optimizes the minimal precision limit *mp* to achieve maximal F-measure.

**Observation** (suitability of *mp* to measure separability). If the target topic is hard to separate from the off-topics because there are few words with high individual precision, high *mp* would lead to low recall because there would be too few keywords. If two topics share many words, it is hard to find keywords separating them: words appearing in documents of both topics will have lower *iprec* for both topics, thus collecting keywords for sufficient recall requires a lower *mp* minimal individual precision limit.

This observation makes *mp* suitable to measure the separability of the target topic from a given set of off-topics. It expresses the estimated lower precision limit of the document selection and thus it can be used to create sets of topics that are easy or hard to separate which can be useful for constructing classifier ensembles.

In the previous discussions, a given topic set and a target topic $T$ was assumed. In the following, the set of off-topics is changing and several measures depend on the current off-topic set. The separability of the target topic $T$ from a set of off-topics $\mathbb{U}$ is measured with the $mp(T, \mathbb{U})$ minimal individual precision limit which, of course, depends on $\mathbb{U}$.

The key question is how to estimate $mp(T, \mathbb{U})$ for topic $T$ given the set of off-topics $\mathbb{U}$. If an off-topic set $\mathbb{U}$ has to be found that conforms to some conditions like minimal $mp(T, \mathbb{U})$, executing PKS for all possible $\mathbb{U}$ off-topic sets would be excessively time consuming. Instead, an approximation which can be calculated by using only the pairwise separability of the topics, is proposed. It requires the execution of PKS only $n(n-1)/2$ times where $n$ is the number of topics. This is illustrated in Fig. 3.5 for target topic $T$ and off-topics $A$ and $B$.

The way how this can be accomplished is presented by the following proposition. Its importance is based on the fact, that it allows the estimation of $mp(T, \mathbb{U})$ in acceptable time, even if it has to be estimated for many possible $\mathbb{U}$ off-topic sets.

**Proposition 3.8.** *Given a target topic $T$ and a set of off-topics $\mathbb{U}$, $mp(T, \mathbb{U})$, as optimized by PKS, can be approximated with*

$$\hat{mp}(T, \mathbb{U}) = \frac{1}{1 + \sum_{V \in \mathbb{U}} (\frac{1}{mp(T, \{V\})} - 1)} \tag{3.4}$$

**Proof:** The proof is based on the following two lemmas:

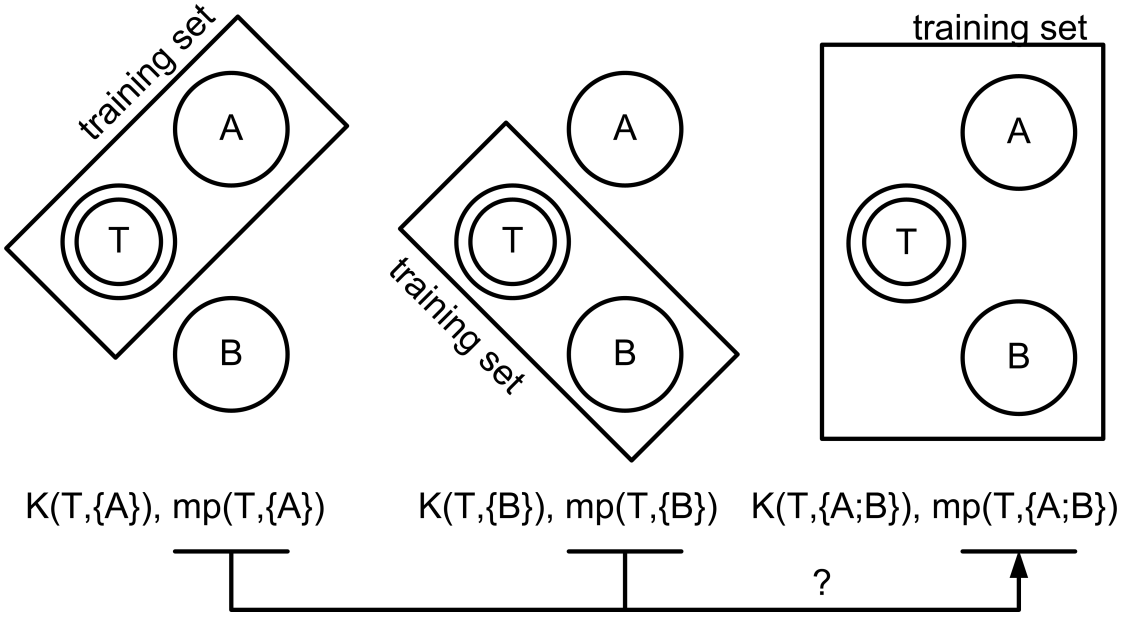


**Figure 3.5.** Estimating the separability of target topic $T$ from off-topics $A$ and $B$. The question is $mp(T, \{A; B\})$, using $mp(T, \{A\})$ and $mp(T, \{B\})$.

**Lemma 3.9.** *If all the documents containing a given $w \in K_T$ keyword are selected, the number of correctly selected documents $c(w, T) = |\{d : d \in T, w \in d\}|$ is independent of the set of off-topics $\mathbb{U}$.*

The lemma is true because $c(w, T)$ is only related to the documents in the target topic, as off-topic documents cannot be correctly selected.

In the followings, $f(w, \mathbb{U})$ is the number of false selections, that is, the number of off-topic documents selected by $w$, if $\mathbb{U}$ is the set of off-topics.

**Lemma 3.10.** *Given a set of off-topics $\mathbb{U}$, the individual precision of a given word $w$ with respect to the target topic $T$ is*

$$iprec(w, T, \mathbb{U}) = \frac{1}{1 + \sum_{A \in \mathbb{U}} \left( \frac{1}{iprec(w, T, \{A\})} - 1 \right)} \tag{3.5}$$

This lemma can be proven using

$$iprec(w, T, \mathbb{U}) = \frac{c(w, T)}{c(w, T) + f(w, \mathbb{U})} \tag{3.6}$$

$$= \frac{c(w, T)}{c(w, T) + \sum_{V \in \mathbb{U}} f(w, \{V\})} \tag{3.7}$$

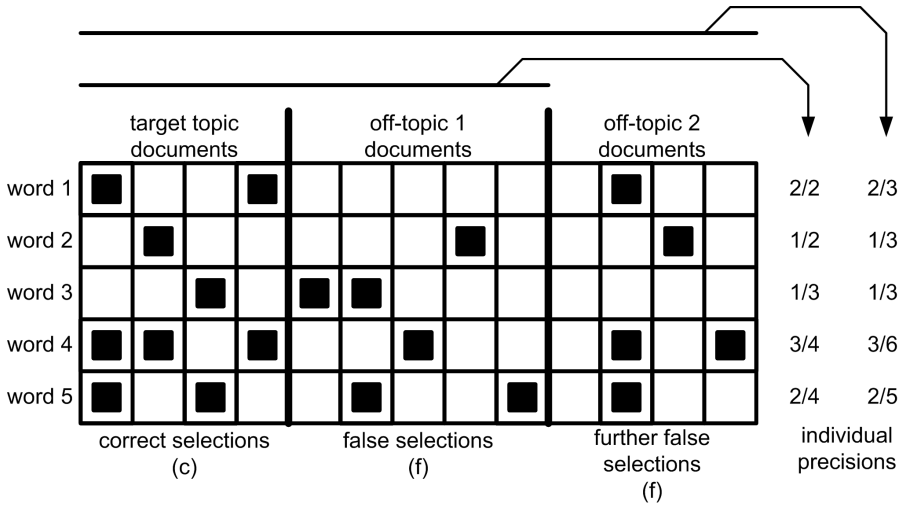


**Figure 3.6.** Individual precision, number of correct and false selections with different number of off-topics.

and $f(w, \mathbb{U}) = c(w, T)(\frac{1}{iprec(w,T,\mathbb{U})} - 1)$.

The number of correct and false selections, together with the individual precision changing due to changing number of off-topics is illustrated in Fig. 3.6.

Using these two lemmas, the proposition can be proven as follows: if a single $V \in \mathbb{U}$ off-topic is given, one can calculate $mp(T, \{V\})$ by executing PKS. The key idea of the proof is that if further off-topics are added, $mp(T, \mathbb{U})$ is estimated with the $mp$ value which would be necessary to get the same keyword list as with only the off-topic $V$. This requires the modification of $mp$ so that the keywords remain in the keyword list, although their individual precisions will decrease due to the new off-topics.

**Assumption 3.11** (worst keyword)**.** *The keyword $w_{worst}$ with the lowest individual precision with off-topic $V$, will have the lowest iperc with off-topic set $\mathbb{U}$, too.*

This assumption is illustrated in Fig. 3.7.

Taking the word with the lowest *iprec* in the keyword list, that is,

$$w_{worst} = \arg\min_{w \in K_T} \{iprec(w, T, \{V\})\}, V \in \mathbb{U}, \tag{3.8}$$

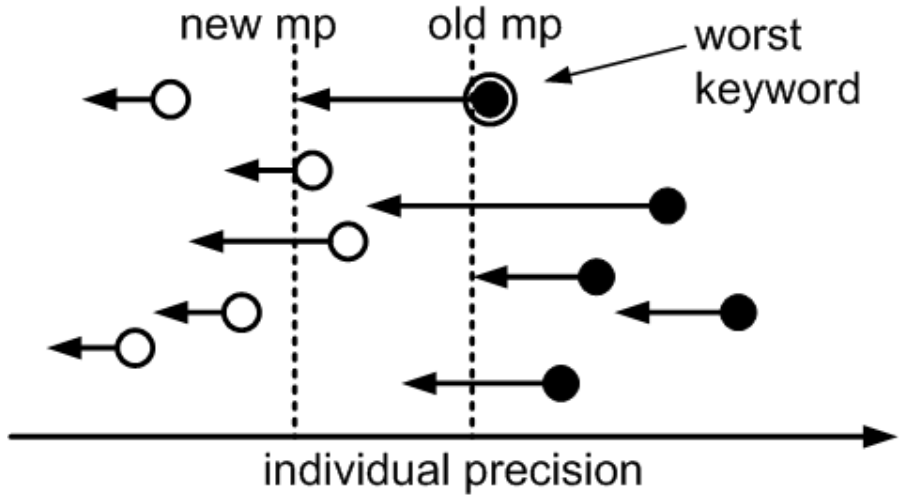


**Figure 3.7.** Worst keyword assumption. Solid circles represent keywords, empty circles stand for non-keyword words. Arrows indicate the change of *iprec*.

The condition $\hat{mp}(T, \mathbb{U}) \leq iprec(w_{worst}, T, \mathbb{U})$ is required to keep $w_{worst}$ in the keyword list $K_T$ which leads to the estimation

$$\hat{mp}(T, \mathbb{U}) = iprec(w_{worst}, T, \mathbb{U}) \tag{3.9}$$

With this assumption, $\hat{mp}(T, \mathbb{U})$ can be calculated from $mp(T, \{V\}) : V \in \mathbb{U}$ similarly to Eqn. 3.5 which proves the proposition.□

## 3.4 Comparison to related work

The results proposed in my first thesis have many connections to related work. The most important relationships will be discussed in the following.

The document model used as the basis of representation is the well known vector space model [Salton, 1987]. The key idea of the thesis is to select only the most topic specific keywords which is basically a feature selection. In my theses, this has a more important rule then usually, because the classification method takes advantage of it: using the keyword selection results, the assigning of weighting during the classification can be omitted. The keyword property is – to my knowledge – unique as the representation size is usually not so critical in related research areas. Common techniques like mutual information between words and documents

[Garner and Hemsworth, 1997], some TFIDF type weighting [Salton et al., 1975], or Kullback-Leibler divergence [Büttcher and Clarke, 2006] for example do not have to emphasize the classification precision over recall. Another difference is that my method contains a feature selection for every topic separately, while the common approach is to select the best separating words in one step for every topic.

The introduced classification method is built over the feature selection method's properties and thus it is not used together with conventional feature selection methods, as that would lead to very low performance. The common solutions [Qi and Davison, 2009] use feature weighting for example. The proposed classification method is loosely related to maximum likelihood decision, but the involved probabilities are only estimates which does not allow exact likelihood calculations.

The F-measure used as the main measure for keyword list quality was chosen because it takes the two measures precision and recall into consideration. I believe, these measures are the best choice to represent the user expectations in this application like low number of false notifications and after that, a possibly high rate of successfully detected documents. Of course, there are many other measures which could be used instead of F-measure: the 2x2 contingency table created by the properties of the documents selected or not, and being in target topic or not, can be evaluated in many ways. For example the Jaccard index, Chi-square or ROC (receiver operating characteristic) curve are all suitable to do this. The difference is in the property captured by them: the Jaccard index and Chi-square do not distinguish the false positive and false negative rates. The ROC curve uses the false positive rate on the horizontal axis, which is related to false notifications. Emphasizing this rate could make the ROC curve an alternative to the technique used in my approach, although I believe that precision is still better to express the user preferences in this application.

An important research area where the rate of false positive and false negative results do not have the same cost, is the spam filtering. A very good comparison is presented in [Zhang et al., 2004], where multiple classifiers (SVM, Naive Bayes, Boosting, memory-based, and Maximal Entropy based classification), and evaluation measures (Information Gain, Document Frequency, Chi-Square) are compared with respect to the number of selected features and the capability to handle the asymmetric costs. Further Naive Bayes classifier based approaches for spam filter-

ing are presented in [Androutsopoulos et al., 2000]. A minimum description length based method is presented in [Bratko et al., 2006] which is related to the compressibility of topic representation. [Sakkis et al., 2003] presents a document representation with binary keyword-presence vectors using information gain feature selection and k-Nearest-Neighbor classification, and explains that cost-sensitive evaluation is seldom emphasized in text categorization.

## 3.5 Experimental results

In this section, several experimental results are presented in order to evaluate the most important aspects of the proposed keyword selection, the classification and the topic separability estimation.

First, some results are shown which present some insights into the behaviour of the individual precision *iprec*, and minimal individual precision limit *mp*. After these, measurements related to assumption 3.5 are presented. Following these, classification results are investigated using the data sets 20 Newsgroups, RCV1 (LYRL2004 split) and Ohsumed. The topic separability approximation is evaluated using approximated and measured separabilities of the topics in the 20 Newsgroups data set. Finally, two measurements are presented investigating the mean number of keywords a document contains, and some examples of the reasons for misclassifications.

### 3.5.1 PKS-related measurements

First, changes of the individual precision of the words *engine* and *later* are presented in Fig. 3.8, while more-and-more off-topics are added to the data set. The target topic is *rec.autos*, *engine* is a keyword of this topic in the whole 20 Newsgroups data set. The word *later* is not topic specific at all, so the iprec is low with even one off-topic. On the other hand, *engine* is related to significantly less topics and has high *iprec* which begins to significantly decrease when the topic *rec.motorcycles* is added.

Fig. 3.9 shows all the words in the RCV1 data set. The figure is created by selecting a target topic (*GCRIM* in this case) and for every word, the number

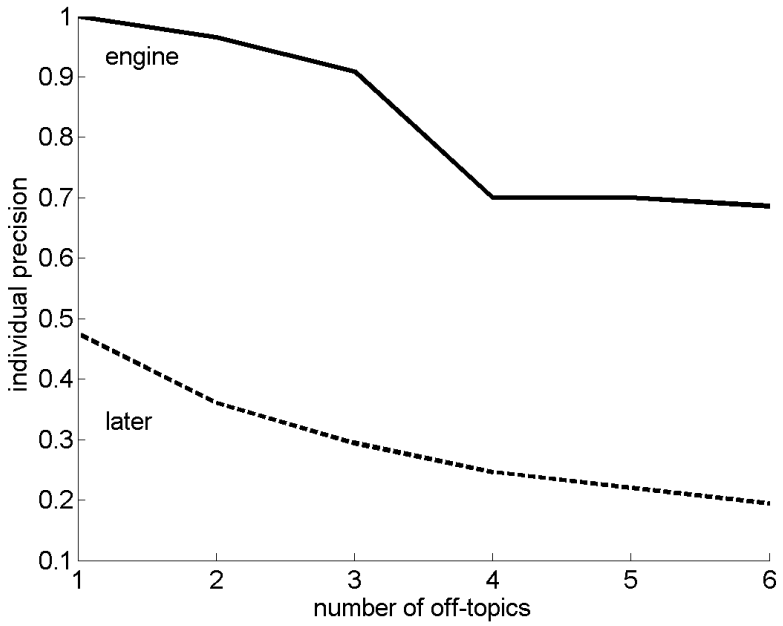


**Figure 3.8.** The change of individual precision of two example words while more-and-more off-topics are added to the data set. The off-topics in order of addition are: *talk.politics.guns, sci.crypt, sci.space, rec.motorcycles, sci.med, comp.graphics*

of containing documents are counted inside and outside the target topic. If a selector would select the documents for *GCRIM* which contain a given word, the two coordinates in the figure would be the number of correct and false selections. The line corresponding to $mp = 0.69$, optimized for this topic, is also indicated. Keywords are below this line.

Fig. 3.10 presents the precision, recall and F-measure curves which PKS is working on while creating the keyword list for the topic *GCRIM*. All the stemmed words are presented additionally with their individual precision and recall as co-ordinates. The final keyword list will consist of the words being on the right-hand side of the *mp* line.

As PKS assigns words to 1% wide individual precision intervals, there are cases when multiple words are assigned to the same interval. This is confirmed by Fig. 3.10. In marginal cases, if there would be a huge amount of words in one interval, that could prevent the fine tuning of the keyword list. Based on my experiences and the experimental results, this effect is not critical in the individual

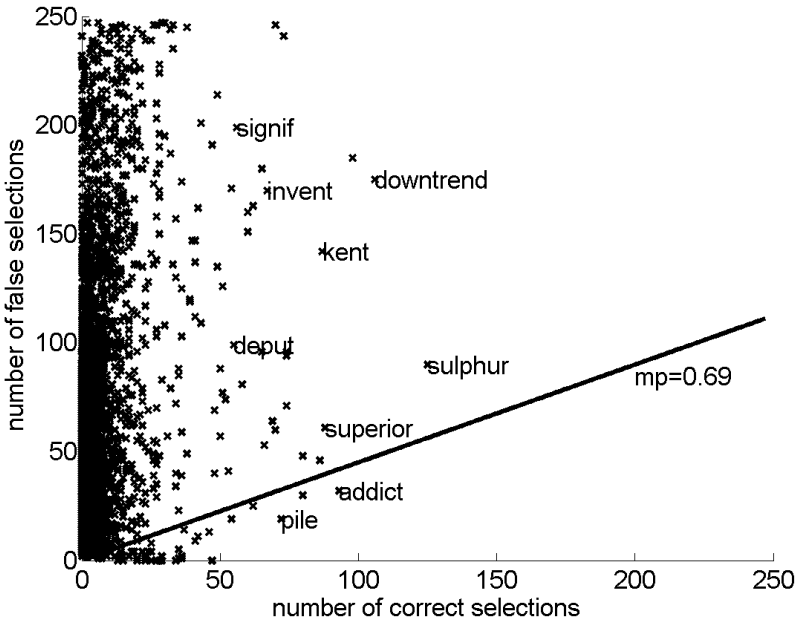


**Figure 3.9.** Words of RCV1 in the space of the number of correct and false selections with respect to the target topic *GCRIM*. Words causing more than 250 false selections have been removed to improve the visualization.

precision intervals really used for keyword list creation, only in intervals for low individual precisions.

Table 3.1 presents the detailed results of the PKS algorithm on all the topics of the 20 Newsgroups data set. Using the indicated keyword number, the communication traffic size of transmitting a compact document representation can be calculated: assuming 16 bit topic identifiers, for example a document in topic *comp.graphics* takes $16 + 36 = 52$ bits. In order to better observe the classification result estimation of the PKS algorithm, the precision of the classification is estimated using the minimal precision limit *mp*. Fig. 3.11 presents the results for all topics of the 20 Newsgroups separately. The topics are ordered in increasing *mp* order to improve the comparability. It is clear that *mp* does not overestimate the measured precision. Although it is sometimes a pessimistic estimation and significantly underestimates the results, by observing the tendency, *mp* can clearly identify the easy-to-identify and the hard-to-identify topics.

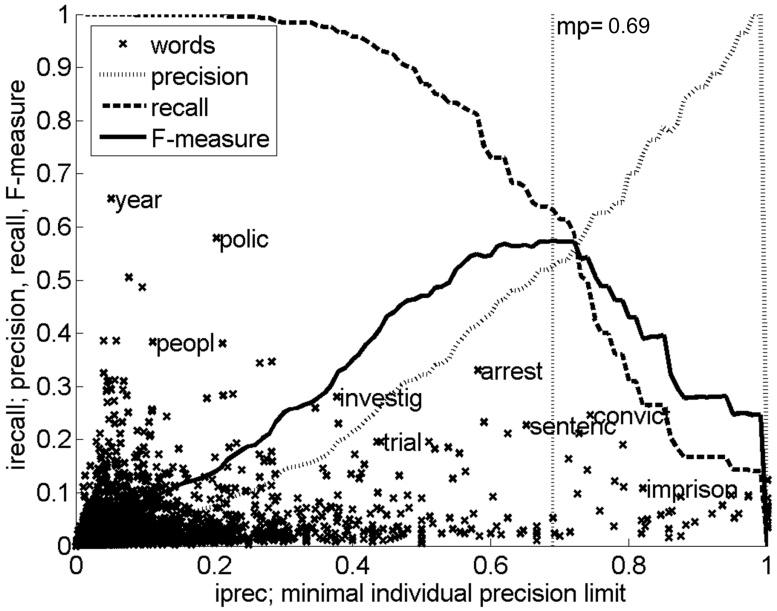


**Figure 3.10.** Words and PKS curves while creating a keyword list for the *GCRIM* topic of the RCV1 data set. Coordinates of words represent their individual precision and recall. The final keyword list contains the words on the right-hand side of the *mp* line.

### 3.5.2 Precision estimation measurements

Assumption 3.5 states that $mp_T$ is a lower bound for the expected precision, if the keyword list $K_T$ (or one of its subsets) is used for document selection. It states that if every keyword has an individual precision not lower than $mp_T$ (valid for all keywords), then using these keywords together leads to an expected precision not lower than this limit.

The following experiments investigate the relationship between the $mp_T$ minimal precision limit, the $ep_T$ estimated precision (provided by PKS), and the measured precision retrieved from document selections using the testing document set. The results confirm, that in most cases, $mp_T$ is a lower bound for the expected precision, and $ep_T$ is a usually higher, but still often pessimistic estimation.

The first related measurement is the one presented in Table 3.1. By comparing the columns *mp*, $P_{PKS}$ (which corresponds to *ep*), and $P_{eval}$, *mp* is always lower than the evaluation precision, thus the assumption is valid. The *ep* value is usually higher than *mp*, but it is still a pessimistic estimation, as the only topic where it

**Table 3.1.** PKS internal measures and classification results on the 20 Newsgroups dataset. For every topic, the keyword number and the *mp* value is presented, together with the precision, recall and F-measure during the simulation in PKS (at the maximal F-measure). ($P_{PKS}$ corresponds to the *ep* value returned by PKS.) The last three column present the same values retrieved with document selections from the test data set.

| topic name | $\|K\|$ | $mp$ | $P_{PKS}$ | $R_{PKS}$ | $F_{PKS}$ | $P_{eval}$ | $R_{eval}$ | $F_{eval}$ |
|---|---|---|---|---|---|---|---|---|
| alt.atheism | 4 | 38 | 43.89 | 38.57 | 41.06 | 44.21 | 56.20 | 16.30 |
| comp.graphics | 36 | 37 | 42.22 | 46.14 | 44.10 | 50.93 | 46.81 | 48.78 |
| comp.os.ms-windows | 34 | 39 | 47.86 | 67.38 | 55.96 | 56.79 | 54.12 | 55.42 |
| comp.sys.ibm.pc.hw | 34 | 40 | 45.34 | 44.92 | 45.13 | 46.39 | 35.71 | 40.36 |
| comp.sys.mac.hw | 35 | 36 | 44.61 | 53.54 | 48.67 | 54.17 | 41.43 | 46.95 |
| comp.windows.x | 34 | 36 | 51.29 | 52.24 | 51.76 | 58.20 | 29.58 | 39.23 |
| misc.forsale | 2 | 44 | 65.20 | 50.61 | 56.99 | 75.61 | 34.83 | 47.69 |
| rec.autos | 21 | 42 | 53.40 | 68.59 | 60.05 | 62.86 | 60.39 | 61.60 |
| rec.motorcycles | 20 | 75 | 84.18 | 70.71 | 76.86 | 90.23 | 53.33 | 67.04 |
| rec.sport.baseball | 20 | 37 | 46.56 | 60.26 | 52.53 | 51.39 | 57.33 | 54.20 |
| rec.sport.hockey | 22 | 57 | 73.24 | 59.75 | 65.82 | 80.19 | 31.84 | 45.58 |
| sci.crypt | 23 | 71 | 87.25 | 66.41 | 75.42 | 86.76 | 54.38 | 66.86 |
| sci.electronics | 28 | 19 | 24.82 | 31.81 | 27.88 | 47.06 | 21.71 | 29.71 |
| sci.med | 24 | 46 | 51.34 | 47.32 | 49.25 | 79.74 | 44.69 | 57.28 |
| sci.space | 23 | 46 | 50.79 | 59.08 | 54.62 | 69.74 | 44.17 | 54.08 |
| soc.religion.christian | 5 | 56 | 54.42 | 68.46 | 60.64 | 70.56 | 64.81 | 67.56 |
| talk.politics.guns | 29 | 43 | 44.62 | 72.92 | 55.37 | 54.47 | 57.61 | 56.00 |
| talk.politics.mideast | 28 | 67 | 79.83 | 75.80 | 77.76 | 86.84 | 66.53 | 75.34 |
| talk.politics.misc | 35 | 25 | 26.68 | 42.65 | 32.82 | 45.81 | 33.47 | 38.68 |
| talk.religion.misc | 34 | 26 | 27.56 | 51.19 | 35.83 | 53.06 | 27.37 | 36.11 |

overestimated the evaluated precision is *sci.crypt*. The relationship between *mp* and $P_{eval}$ is visualized in Fig. 3.11, too. This measurement confirms that the assumption can be considered valid for the whole keyword lists.

#### 3.5.2.1 Precision of keyword list subsets

The second experiment is designed to investigate the precision of the subsets of the keyword lists. A document selection is performed, but instead of using all the keywords, only a subset of the keyword list is employed. To investigate the keyword list subsets really present in the data set, only the subsets present in at least 0.1% of the documents, were considered.

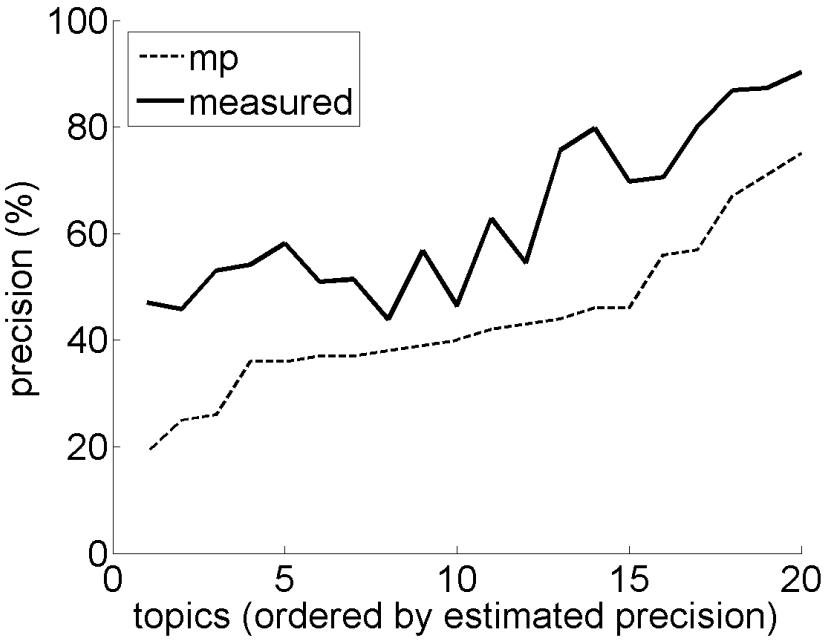


**Figure 3.11.** Estimation of classification precision based on *mp*.

**Table 3.2.** Estimation error of *mp* and *ep* with respect to measured precision.

| measure | OK rate | bias | std.deviation |
|---|---|---|---|
| *mp* | **0.9544** | -0.2479 | 0.1293 |
| *ep* | 0.8633 | -0.1913 | 0.1495 |

The results presented in Table 3.2 show that it is very rare that a keyword set has lower precision than its corresponding *mp* value. This is why I introduced the assumption that this condition is always satisfied. The *OK rate* shows the rate of the cases where the estimated value is lower than the measured precision. In the case of *mp*, it is higher than 95%.

The histogram of the difference between the *mp* and the measured precision is presented in Fig. 3.12. The assumption is invalid only in the cases with difference values over 0.

#### 3.5.2.2 Which topics are influenced?

This measurement investigates the frequency of documents in the various topics which contain a bad keyword subset. These are the documents involved in the violations of the assumption. In the following, I will use the following definition:

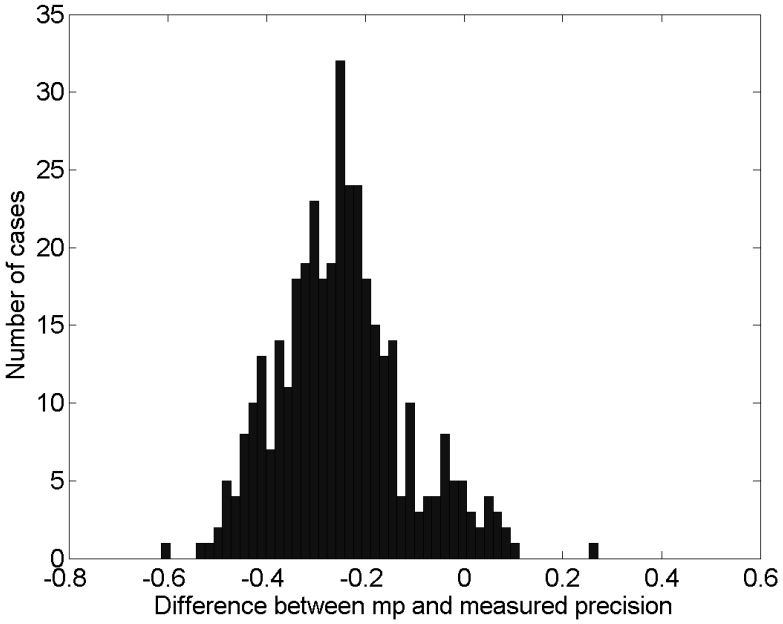


**Figure 3.12.** Histogram of differences between $mp$ and measured precision. In most cases (95.44%), the measured precision is higher than $mp$. In the remaining cases, the mean overestimation of the precision is 0.0614. Every possible keyword list subset was observed, which has at least 0.1% of the documents. (393 keyword list subsets conformed this condition).

**Definition 3.12** (Bad keyword subset)**.** A bad keyword subset is a subset of a keyword list which has a measured precision lower then the $mp$ value of the keyword list.

Bad keyword subsets do not behave according to assumption 3.5.

The critical documents containing a bad keyword subset and belonging to topic $T$ are

$$D_{crit}(T) = d \in D(T) : \text{isbad}(d \cap K_T) \tag{3.10}$$

where isbad() is true if the given keyword subset is a bad keyword subset, $D(T)$ is the set of all documents in topic $T$, and $K_T$ is the keyword list of $T$. It should be noted that not all critical documents are misclassified, they only contain a keyword subset leading to misclassifications more often than expected.

**Table 3.3.** Rate of documents containing bad keyword subsets. Global rate is 0.0321.

| Topic name | CriticalDocRate |
|---|---|
| alt.atheism | 0.0438 |
| comp.graphics | 0.0000 |
| comp.os.ms-windows.misc | 0.0118 |
| comp.sys.ibm.pc.hardware | 0.0040 |
| comp.sys.mac.hardware | 0.0000 |
| comp.windows.x | 0.0000 |
| misc.forsale | 0.0000 |
| rec.autos | 0.0000 |
| rec.motorcycles | 0.0044 |
| rec.sport.baseball | 0.0178 |
| rec.sport.hockey | 0.0637 |
| sci.crypt | 0.0323 |
| sci.electronics | 0.0000 |
| sci.med | 0.0037 |
| sci.space | 0.1292 |
| soc.religion.christian | 0.1245 |
| talk.politics.guns | 0.0576 |
| talk.politics.mideast | 0.0040 |
| talk.politics.misc | 0.0694 |
| talk.religion.misc | 0.0772 |

The rate of documents containing a bad keyword subset ($CriticalDocRate$) is presented in Table 3.3 for every topic and it is defined as

$$CriticalDocRate(T) = \frac{|D_{crit}(T)|}{|D(T)|}. \tag{3.11}$$

The rate of documents with bad keyword subsets in the whole data set (for all topics together) is 0.0321. Based on the results, the assumption is considered to be valid for most cases in the data set. Higher bad keyword subset frequencies are mainly caused by too similar topics, like the subtopics of *talk.politics* or *rec.sport*.

For a comparison, the similarity matrix of the data set is presented in Fig. 4.11 on page 70.

**Table 3.4.** Bad keyword subsets having a minimal frequency of 0.001. Topics not having any bad keyword subsets are not mentioned.

| keyword list | frequency | eff.prec-mp |
|---|---|---|
| Topic: rec.sport.baseball | | |
| league, players | 0.0030 | -0.0120 |
| Topic: sci.space | | |
| space, henry | 0.0020 | -0.0361 |
| space, orbit | 0.0032 | -0.0257 |
| space, orbit, henry | 0.0014 | -0.0167 |
| Topic: soc.religion.christian | | |
| church, sin | 0.0010 | -0.0267 |
| church, christ | 0.0044 | -0.0366 |
| church, christ, sin, rutgers, athos | 0.0010 | -0.0154 |
| Topic: talk.politics.guns | | |
| compound, bd | 0.0014 | -0.0603 |
| sw, bd | 0.0012 | -0.1065 |
| Topic: talk.politics.misc | | |
| clinton, taxes | 0.0026 | -0.0031 |
| clinton, bush | 0.0026 | -0.0080 |
| clinton, health | 0.0012 | -0.0266 |

#### 3.5.2.3 Keyword subsets causing low measured precision

Table 3.4 presents the bad keyword subsets having a frequency of at least 0.001. Based on the results, bad keyword subsets are usually caused by a few critical keywords: low measured precision is usually caused by multiple critical keywords appearing in a document together. Bad keyword subsets indicate that their topic is similar to another one, like *soc.religion.christian* and *alt.atheism*, or the subtopics of *rec.sports* or *talk.politics*.

Formally, the following keyword subsets are shown in Table 3.4 for every $T$ topic:

$$W \subseteq K_T : \text{isbad}(W), \frac{|D(W) \cap D(T)|}{|D(T)|} \geq 0.001 \tag{3.12}$$

where $D(W)$ is the set of documents containing the keyword subset $W$.

These results are also a precision check for all possible keyword subsets appearing in the test documents. Keyword subsets not appearing in Table 3.4 either correspond to the assumption and thus they are not bad, or they appear too rare in the test document collection.

#### 3.5.2.4 Summary of the measurements supporting assumption 3.5

Although the keyword lists created by the PKS algorithm do not guarantee a precision not lower than the *mp* parameter provided by PKS, the presented measurements confirm that in most cases, the assumption on minimal expected precision limit (assumption 3.5) is valid in the used data set. This allows simple formal proofs and does not negatively influence the practical application.

It is clear that there can be data sets leading to much worse results, artificially generating such a data set is relative easy. But in the data sets considered to be similar to the data in the target application of the proposed system, the assumption is considered to be valid in most cases and thus, it can be used for performance estimations.

I decided not to include the precision check of keyword list subsets into the keyword selection algorithm, as I believe the results would not be significant (rate of critical documents was below 5% in the measurements for 20 Newsgroups), but the procedure would be very resource consuming. Beside this decision, a checking step can be easily added if needed. Using a simulated selection performed on the data set used by PKS, the expected precision of every keyword list subset can be checked after each other. Even if all the subsets could not be checked due to their exponential growing number, the subsets with for example 2 or 3 elements can be checked. If a precision under the minimal individual precision limit is detected, a keyword may be removed from the keyword list. Identifying the keyword to remove may not be a trivial question and can be a subject of further research, but considering the removal of the keyword with the lowest individual precision (among the keywords in the bad keyword subset) may be a good starting point.

### 3.5.3 Classification measurements

In order to have an overview of the complexity of the classification problem, multiple baseline measurements were performed. Both the keyword selection method and the classification method are compared to a baseline method: PKS was compared to mutual information based feature selection, and the classification method *most keywords* (MKw) was compared to naive bayes (NB) classifier. Mutual

information-based feature selection selects a given number of words which have the highest mutual information with the topic of the documents.

Mutual information based feature selection was chosen because its simplicity making it similar to the PKS algorithm in resource consumption. The most significant difference is that high mutual information can require negative weighting which cannot be represented without weighting in the keyword lists, and thus, MKw cannot use its results.

Naive bayes classifier was chosen for baseline measurements and comparisons because of its simplicity and linear execution time which makes it similar to the MKw method with respect to resource consumption and execution time.

Results are presented in Table 3.5. It is clear that PKS significantly increases the precision and achieves higher F-measure with both the naive-bayes classifier and the MKw method. Using PKS, the MKw method achieves significantly higher precision and only slightly lower F-measure than the naive bayes classifier. For the small decrement in F-measure, a significant advantage (besides the higher precision) is provided: using the MKw classifier there is no need to transfer and store the weight vectors introduced by the naive bayes classifier, only the keyword lists themselves, represented with the list of keyword indices. The size of the keyword lists created by PKS are small (especially for the data sets 20NG and Ohsumed), and the baseline method could not outperform its performance even with 500 keywords.

### 3.5.4 Approximating *mp* for multiple off-topics

In order to check the suitability of *mp* for separability measurement, first, the pairwise separability of the topics in 20 Newsgroups are presented in Fig. 3.13. PKS was executed on all possible pairs of topics and the *mp* values are presented in the figure. The lowest (hardest) separability (0.69) belongs to the topic *alt.atheism* if the off-topic is *talk.religion.misc*. On the other hand, *talk.politics.mideast* is very easy to separate from *comp.graphics* (0.99). It should be noted that the separability is not symmetric, because of the different sets of topic specific words.

The estimation of the minimal individual precision limit *mp* was evaluated in the following way: *mp* was estimated with Eqn. 3.4 for randomly chosen target and

**Table 3.5.** Classification results using various data sets in terms of precision, recall and F-measure. Results with mutual information (Mut.Inf.) are taken using the same keyword number as PKS. The maximal achievable F-measure (with 1-500 keywords) is presented in brackets. The mean keyword numbers per topic returned by PKS are the following: 237.88 for RCV1, 24.55 for 20NG and 39.70 for Ohsumed.

| measurement | dataset | precision | recall | F-measure |
|---|---|---|---|---|
| Mut.Inf.+NB | RCV1 | 0.3541 | 0.4919 | 0.4118 (max 0.42) |
| PKS+NB | RCV1 | 0.4600 | 0.5161 | 0.4864 |
| PKS+MKw | RCV1 | 0.6355 | 0.4100 | 0.4746 |
| Mut.Inf.+NB | 20NG | 0.4296 | 0.5656 | 0.4883 (max 0.5) |
| PKS+NB | 20NG | 0.4781 | 0.5395 | 0.5070 |
| PKS+MKw | 20NG | 0.6152 | 0.4582 | 0.5024 |
| Mut.Inf.+NB | OHS | 0.3477 | 0.2789 | 0.3095 (max. 0.35) |
| PKS+NB | OHS | 0.3628 | 0.5266 | 0.4296 |
| PKS+MKw | OHS | 0.4342 | 0.4078 | 0.4065 |

off-topic sets for off-topic numbers 2 to 10. Several experiments were preformed for every off-topic number. The measured *mp* is shown in Fig. 3.14 together with the approximation error. The approximation is not significantly biased (mean bias of approximation is -0.0064) and the highest approximation error rate is 2.59%.

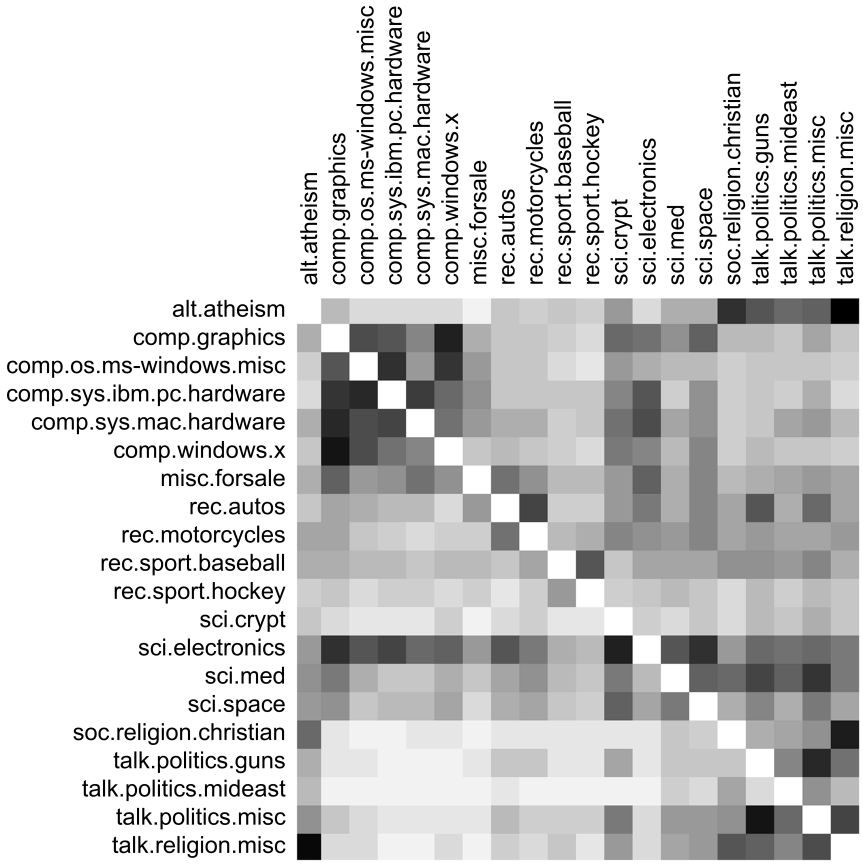


**Figure 3.13.** Pairwise separability of the topics in 20 Newsgroups. Black rectangles indicate hard, and white ones indicate easy separability.

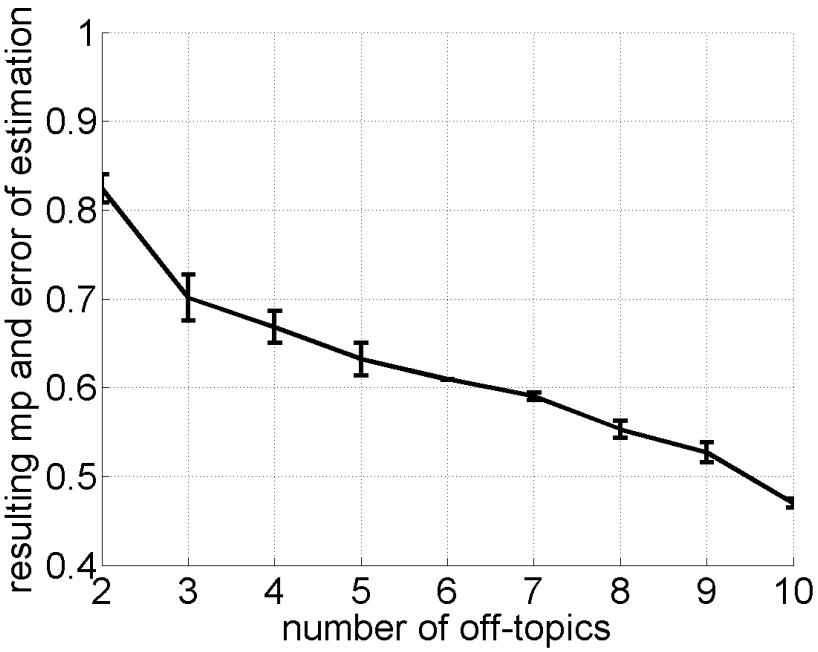


**Figure 3.14.** Results of *mp* approximation: the mean of measured *mp* value and the error of the approximation is shown for several off-topic numbers.

Chapter 4

# Searching for similar documents

This chapter proposes the searching method for mobile devices which want to find documents similar to the local ones. This is mainly a 1-class classification task: documents corresponding to some criteria are selected. The most important property is the communication traffic: documents are compared and the decision is made using only the proposed compact document representation to maintain low communication traffic. Otherwise, the process would be finantially unsuitable if the communication if not free of charge, and the battery of the mobile device would be depleted in a few hours.

## 4.1 Similarity search

The search for similar remote documents is the principal goal of the techniques proposed in this dissertation. If the mobile device detects a remote document which has similar topic to at least one of the documents stored locally, it notifies the user that a probably interesting document is available for download. (In order to support the decision of the user, wether to download the document or not, the system may show the keywords the remote document contains, or even download the beginning of the document.)

Deciding wether a remote document has similar topic to the local ones, is a 1-class classification task: the classifier selects the remote documents which are similar to the local ones. As the similarity of the topic of two documents is defined with the number of common keywords, a keyword mask (bitmask) is transferred

between the mobile devices, which will serve as a 1-class classifier: documents having common keywords with it may be interesting for the user.

Theoretically, the classification could be performed on both sides, but the classification helping one user to find similar documents should consume the resources of the mobile device of that user. The search for similar documents on a remote mobile device is illustrated in Fig. 4.1. After the compact document representations (defined in the next section in details) have been asked for and retrieved, the similarities to the local documents are calculated. If the similarity is high enough, the user is asked if the document should be downloaded or not. If the keyword list used by the compact representation is not known, it is retrieved from a central server storing all keyword lists. Otherwise, accessing the keyword list server is not necessary.

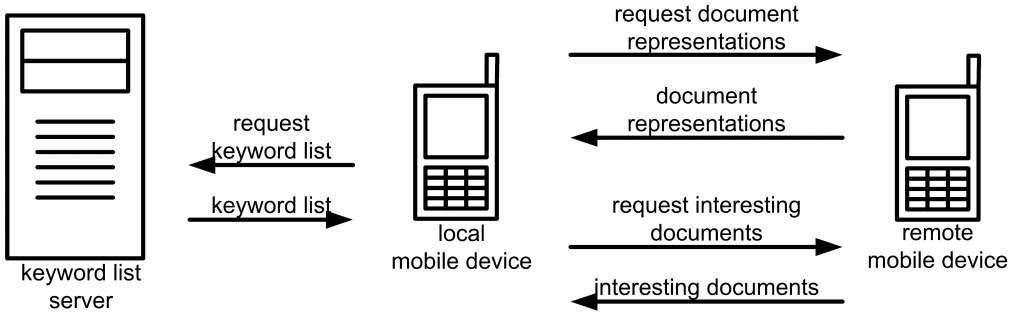


**Figure 4.1.** Searching for similar documents.

### 4.1.1 Searching for similar documents

The key idea of the compact document representation is that all documents are represented by the identifier of the best matching keyword list, and a binary vector indicating the presence or absence of keywords.

**Definition 4.1** (Compact document representation)**.** The compact document representation of a document $d$ is the pair $(\hat{T}(d), \mathbf{p}(\hat{T}(d), d))$ where $\hat{T}(d)$ is the estimated topic of the document $d$ and $\mathbf{p}(\hat{T}(d), d)$ is a binary vector indicating the presence or absence of the keywords of topic $\hat{T}(d)$ in the document $d$.

The compact document representation is sufficient to recover the document vector, if all the keywords not in the keyword list of the documents estimated topic

are considered not present in the document. The size of the compact document representation is only the size of the topic identifier (for instance 16 bits) and 1 bit/keyword.

Compact document representations using different topics are illustrated in Fig. 4.2. During the comparisons, these binary vectors are mapped into a global keyword space (using the keyword list identifier), where the scalar product returns the number of common keywords even of documents represented with different keyword lists. The number of common keywords is the similarity measure employed by the proposed system. The document similarity calculated for documents represented using different keyword lists is illustrated in Fig. 4.3.

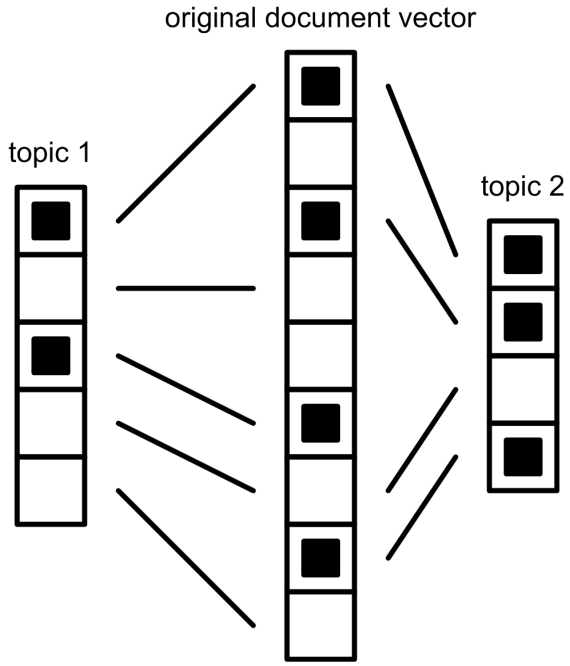


**Figure 4.2.** Document representations using the keyword list of different topics. Words not mentioned in the employed keyword list are considered not to be present in the document. The keyword list having the most common keywords with the document is used for the compact document representations (topic 2 in this example).

**Definition 4.2** (Similarity measure)**.** If $\underline{t}$ and $\underline{d}$ are binary document vectors in the global keyword space, than the similarity of the two documents is defined as

$$\text{similarity}(\underline{t}, \underline{d}) = \underline{t}^T \cdot \underline{d} \tag{4.1}$$

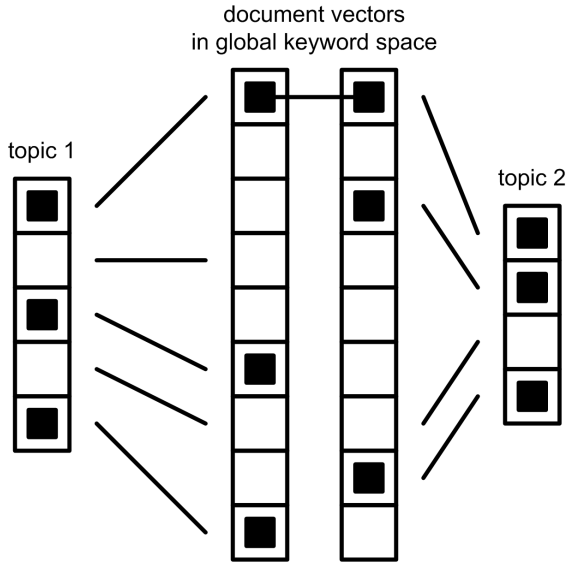


**Figure 4.3.** Documents represented with different keyword lists have to be mapped into the global keyword space prior to comparison. In this example, the documents have one common keyword.

Searching for documents similar to multiple base (locally stored) documents is performed with the merged base document vector which represents all base documents together:

**Definition 4.3** (Merged base document vector $\underline{b}$)**.**

$$\underline{b} := \mathrm{sign}\Big(\sum_{d \in B} \underline{d}\Big) \tag{4.2}$$

where $B$ is the set of base documents and $\underline{d}$ is the vector of document $d$ in the global keyword space. The creation of the base document vector is illustrated in Fig. 4.4.

Using the merged base document vector and the compact document representations of remote documents, the similarity search can be performed.

**Definition 4.4** (Similarity search)**.** The similarity search is the process of downloading the compact document representations of remote documents and calculating the number of their common keywords with the base documents. If this number exceeds a user-defined threshold, the user is notified.

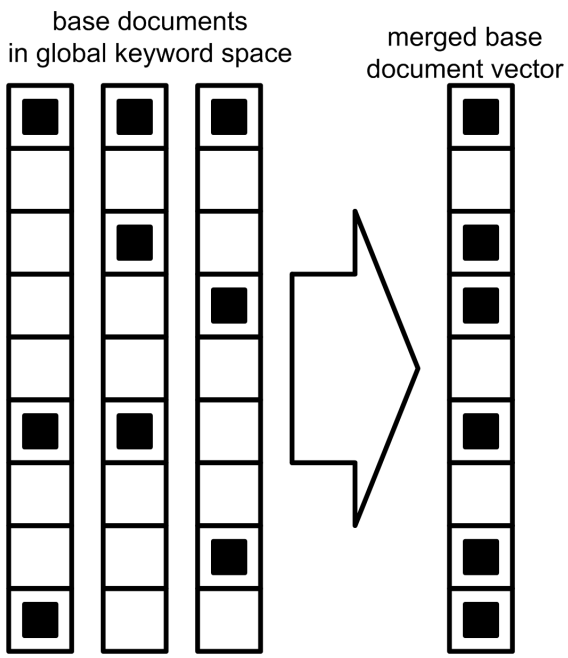


**Figure 4.4.** Creating the base document vector by merging all local (base) documents in the global keyword space. Remote documents are checked for similarity to this merged document.

The minimal similarity measure, for user notification, a document must have to the base documents, is the *threshold* parameter defined by the user. Based on the experimental results presented later in this chapter, it is suitable to control the balance between precision and recall: lower threshold allows the selection of more documents but increases the chance of misclassifications as well. According to the similarity search, the following proposition can be proven:

**Proposition 4.5** (Minimal precision of similarity search)**.** *Assuming that assumption 3.5 is valid, the search for documents similar to the base documents has an expected precision not lower than the lowest mp among the topics of the base documents.*

**Proof**: The similarity search is a 1-class classification like the document selection. The only difference is that keywords not present in the base documents are not used. This means that the lower limit of expected precision described in assumption 3.5 is valid here are well, because a $W \subseteq K$ keyword list is used for the selection. If there are multiple base documents, the result of the similarity search can be considered as the union of document selections using the keywords of the

base documents after each other. As the base documents may come from multiple topics, the lowest *mp* among these topics has to be used.□

### 4.1.2 Accessing available keyword lists

If a mobile device encounters an unknown keyword list, it has to download it from a central server storing all available keyword lists. As there will be many keyword lists which do not share any common keywords with the base documents, and thus documents represented with these keyword lists cannot have non-zero similarity with the base documents, it is sufficient to store only the identifier of such keyword lists and to remember not to download them again. In this way, the mobile devices only have to store the keyword lists of relevant topics.

### 4.1.3 Documents with multiple topics

A drawback of the proposed compact document representation is the unability to represent documents with more than one topic. It always requires the identification of a single, best-matching topic, which is then used. It is designed this way to minimize the representation size. If an extension to handle multi-topic documents is needed, two basic approaches can be followed:

- A document may have multiple compact document representations, one for every represented topic. This extension is easy to implement, but it does not allow a remote document to achieve the similarity threshold using keywords from multiple topics, as the topics are represented separately and not handled together.

- The representation can be extended for storing multiple "topic identifier - keyword presence vector" pairs. This allows a remote document to have the necessary number of common keywords with the based documents using all its topics, but it requires also the slight extension of the implementation of the similarity calculation.

## 4.2 Document extension

A system using the method presented in the previous section can search for documents similar to the base documents. Unfortunately if two documents had related topics, like *dolphins* and *hawks*, but they were not sharing any common keywords, their similarity measure would be zero and they would be considered to be completely different, just as any other documents with entirely different topics. The *document extension* procedure slightly increases the similarity measure of documents which have related topics. For example let's consider two documents without common keywords: one about hawks and one about dolphins. These documents are assumed to contain the keywords *hawk* and *dolphin* respectively. If the system recognizes *animal* to be a *related generalizing concept (RGC)* to both *hawks* and *dolphins*, the keyword *animal* can be added to both document representations rendering the similarity higher than zero. This would allow finding loosely related documents too.

The document extension can be performed on either the mobile device creating the compact document representation, or a globally accessible server can be employed which extends the compact document representations sent to it.

**Definition 4.6** (Related General Concepts Function (RGCF))**.** *RGCF* is the function returning the set of related generalizing concepts (keywords) $v_i$ for the keyword $w$: $RGCF(w) = \{v_1, v_2, ..., v_n\}$.

It should be noted that the RGCs have to be keywords as well, otherwise their addition could not be indicated in the document representation.

**Definition 4.7** (Document extension)**.** The document extension adds all the generalizations of the keywords of a document, to the document:

$$d^{ext} = d \cup \bigcup_{w \in d} RGCF(w) \tag{4.3}$$

where $d^{ext}$ is the extended document.

Two ways for creating the RGCF are presented in the following: an unsupervised RGCF learning method and a WordNet based approach.

### 4.2.1 Keyword co-occurrence (KCo) based RGCF learning

This RGCF learning method is an unsupervised method extracting the word relationships from a labeled data set having a topic hierarchy. For the sake of simplicity, a two-level hierarchy with upper and lower level topics is consider, but the methods can be easily generalized to more levels. The method is based on the assumption that keywords of upper level topics are more general than the keywords of lower levels, and that frequently co-occurring keywords are related to each other.

If two documents have a similar upper level topic (for example both are about animals), they are assumed to tend to contain lower level topic specific keywords for their own topic, like *hawk* and *dolphin*, but they also often contain more general keywords from keyword lists of upper level topics such as *animal*. This observation suggests that both *hawk* and *dolphin* are related to *animal*, but *animal* is the keyword of an upper level topic which means it is specific to something more general than *hawk* and *dolphin*. If a word is keyword of an upper level topic, that means that it is very specific to that upper level topic and it is more general than the keywords of the lower level topics.

The RGCF function returns keywords satisfying the following condition:

$$v \in RGCF(w) \leftrightarrow w \in K_G, v \in K_H : G \subseteq H, \frac{S(w) \cap S(v)}{S(w)} \geq mcr$$

where $K_G$ and $K_H$ are the keyword lists of topics $G$ and $H$ respectively, $G \subseteq H$ indicates that $G$ is a subtopic of $H$, $S(w)$ is the set of documents containing the word $w$, and $mcr$ is the minimal co-occurrence rate (like the minimal confidence limit in association rule mining). The first condition ensures the generalization and the second ensures the frequent co-occurrence of the keywords $v$ and $w$. An example is shown in Fig. 4.5.

Using these conditions, the RGCF can be learned by collecting the RGCs for every keyword. The evaluation of this RGCF function is performed with the help of related topics:

**Definition 4.8** (Related topics)**.** Two topics are related in a topic hierarchy, if they have a common parent topic.

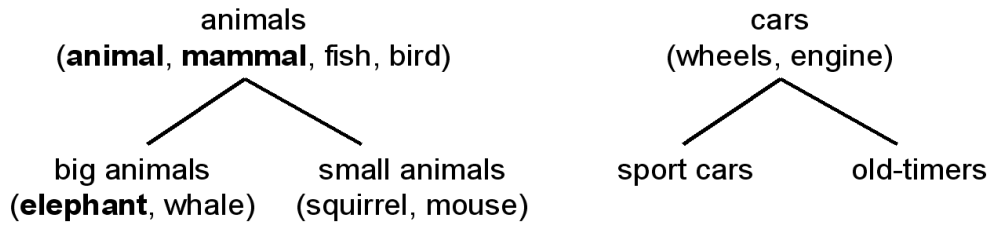


**Figure 4.5.** Example topic hierarchy and keywords. *Big animals* is a subtopic of *animals*. If the keywords *animal* and *mammal* appear often together with the keyword *elephant*, they will be RGCs of *elephant*.

Documents of related topics (for example subtopics of *animals* like *hawks* and *dolphins*) are considered to be a suitable test environment for the document extension, as the common parent topic ensures loose relatedness, but the documents are different enough to share few or no keywords. (It should be noted, that if the topic hierarchy contains a common root element, the definition of relatedness has to be extended with a limit on the distance of the common parent. For example, only direct parents are considered.)

The most important feature of the document extension using the unsupervised RGCF learning approach is summarized in the following proposition:

**Proposition 4.9** (Expected probability of relatedness of documents with increasing similarity measure due to document extension)**.** *If a keyword is added by document extension to two documents $d$ and $f$, and so the similarity measure of the two documents in increased, then a lower bound for the expected probability that $d$ and $f$ belong to related topics is $mp_d \cdot mp_f$ where $mp_d$ and $mp_f$ stand for the mp values of the estimated topics of $d$ and $f$.*

**Proof**: A keyword $v$ is added to both documents $d$ and $f$ if they contain the keywords $w_1 \in d$ and $w_2 \in f$ and $v$ is RGC of these keywords: $v \in RGCF(w_1) \cap RGCF(w_2)$.

If these conditions are satisfied and the two keywords $w_1$ and $w_2$ belong to topics $A$ and $B$, that is, $w_1 \in K_A$ and $w_2 \in K_B$, then based on the property of keywords (Eqn. 3.2),

$$\hat{Pr}(d \in A | w_1 \in d) \geq mp_A \tag{4.4}$$

$$\hat{Pr}(f \in B | w_2 \in f) \geq mp_B \tag{4.5}$$

As the *KCo* algorithm searches RGCs only in upper level topics, $v \in RGCF(w_1) \cap RGCF(w_2)$ means that $A$ and $B$ are related topics. And as the documents $d$ and $f$ belong to these topics with expected probability at least $mp_A$ and $mp_B$ respectively, the expected probability that $d$ and $f$ belong to related topics is at least $mp_A \cdot mp_B$. □

### 4.2.2 Creating RGCF using WordNet

Creating the RGCF using WordNet is much easier because WordNet already contains hypernym edges which indicate the generalizations (hypernyms) of the words. The nodes in WordNet are set of synonym words (called symsets), and many types of directed edges connect these synsets. One of the edge types points to hypernym symsets.

**Definition 4.10** (Hypernym distance of words in WordNet)**.** The $h(w, v)$ hypernym distance of words $w$ and $v$ is the length of the route along the directed hypernym edges from $w$ to $v$. If $w$ and $v$ are synonyms (they belong to the same synset in WordNet), $h(w, v) = 0$.

For example if *animal* is a hypernym of *mammal*, and *mammal* is a hypernym of *elephant*, then $h(elephant, animal) = 2$.

**Definition 4.11** (WordNet based RGCF learning)**.** The RGCF learned using WordNet is defined as

$$v \in RGCF(w) \leftrightarrow h(w, v) \leq dl \tag{4.6}$$

where $dl$ is the distance limit, a parameter of the learning method.

The distance limit is necessary because in WordNet, almost every word would be related through the word *entity*.

As the document topic representations can indicate only keywords, words retrieved from WordNet which are not keywords of the topic of the documents, have to be omitted.

### 4.2.3 Comparison to related work

The proposed similarity search technique is strongly related to the classification method proposed in my first thesis, as it also takes advantage of the precision-centric feature selection. This leads to similar relationships the the related work. The involved distance measure is using this property too by assuming a clearer separation of the topics in the space of selected features. Distance measures like cosine-distance are more general and do not assume much about the feature space.

The compact document representation is new in the aspect of representation size: common methods do not emphasize the size so much, so they usually use weighted feature vectors with all features in them [Kotsiantis, 2007] and not binary ones with only a subset of all the features.

The main advantages of the proposed RGCF based document extension over the techniques presented in the literature [Billerbeck et al., 2003] [Vechtomova et al., 2003] [Ghanem et al., 2002], are the following:

- The Keyword Co-occurrence based method takes advantage of the precision based keyword selection which is performed in advance. It makes it not need to check the topic specificity of words.
- Words added to the documents have to be represented using the compact document representation. The proposed RGCF learning methods add only keywords, so they can be used with the proposed representation technique.
- The WordNet based RGCF learning method demonstrates a possibility to add semantic information from external sources, if available. (If not, the keyword co-occurrence based method is still available.)

## 4.3 Experimental results

In this section, several experimental results are presented in connection with the proposed RGCF learning methods and the similarity search. First, the KCo and Wordnet based RGCF learning methods are compared and evaluated, and after these, the similarity search results and their changes due to document extension are presented. Finally, a document extension example is presented.

### 4.3.1 Experiments: Learning the Related General Concepts Function

The *RGCF* learning methods collect the RGC keywords for every keyword of lower level topics. 4 RGCF learning cases are investigated: the KCo method and the WordNet based method using distance limits 0, 1 and 2. Table 4.1 summarizes some example words and their generalizations according to the various cases. It is clear that all the methods capture correct generalizations in some sense, but the difference in the operation is clearly visible: the KCo method observes co-occurring words and does not take any meanings into account. This leads to topic dependent generalizations which really belong to the topic of the word (like *game* for *players*). On the other hand, the WordNet based approach captures generalizations based on real meaning and considers the current topic only so far that the generalization has to be a keyword as well. This leads sometimes to generalizations belonging to another sense of the word (like *soul* for *players*). The unsupervised approach seems to be more robust against special words (often not known by WordNet). For example WordNet does not know about *NHL* (National Hockey League) and so it returns no further generalizations. The KCo method recognized that *NHL* is a team game.

Regarding the RGCF learning methods in general, document pairs containing the mentioned generalizations are believed to have related topic with high probability. This does not necessarily mean topic equivalence but indicates a little more similarity than nothing which would be characterized by zero common keywords.

The mean similarities of documents are investigated in Table 4.2 during the document extension using KCo based RGCF. The document extension is meant to increase the similarity of related documents. The table presents the mean pairwise similarities of documents for three types of document pairs: *InterU* (inter-upper-level) is the similarity of documents from different upper level topics. Document extension should not increase these similarities significantly. *IntraU* (intra-upper-level) stands for documents inside the same upper level topic but from different lower level topics. Document extension should increase these similarities. Finally, *IntraL* (intra-lower-level) stands for documents of the same lower level topic. Document extension may increase these similarities as well, but that is not the main

**Table 4.1.** Comparison of RGCF learning results. WN$n$ stands for the WordNet based method where $n$ is the distance limit. The *mcr* minimal co-occurrance rate in KCo was 30%.

| original word | KCo | WN0 | WN1 | WN2 |
|---|---|---|---|---|
| graphics | graphics | art<br>graphics | art<br>graphics | art<br>graphics |
| controller | controller<br>mb | control<br>controller | person<br>someone<br>individual<br>control<br>soul<br>somebody<br>controller | person being<br>someone<br>cause<br>control<br>individual<br>soul device<br>somebody<br>controller |
| players | game<br>team<br>player<br>players | players<br>player | players<br>player | person<br>soul<br>someone<br>individual<br>somebody<br>players<br>player |
| encryption | encryption<br>key secure<br>chip keys | encryption | encryption | writing<br>encryption |
| nhl | nhl game<br>team | nhl | nhl | nhl |
| ball | ball<br>game | ball | ball<br>baseball<br>shot | ball party<br>equipment<br>baseball<br>shot<br>throw<br>player |
| cup | cup team | cup | cup hole | cup solid<br>hole |

goal. The three types of document pairs are illustrated in Fig. 4.6. It should be noted that these are the pairwise similarities of documents (number of common keywords), not the keyword number of documents. The results confirm that the document extension mainly increases the similarity of related documents and does not significantly increase the similarity of unrelated documents.

**Table 4.2.** Mean similarities of documents with different types of topic relationship. Extended documents are indicated with *. The RGCF is learned with the KCo method.

| DataSet | InterU | InterU* | IntraU | IntraU* | IntraL | IntraL* |
|---|---|---|---|---|---|---|
| 20NG | 0.0102 | 0.0137 | **0.1640** | **0.2098** | 0.2824 | 0.4274 |
| RCV1 | 0.0094 | 0.0474 | **0.0957** | **0.2604** | 0.0164 | 0.0527 |

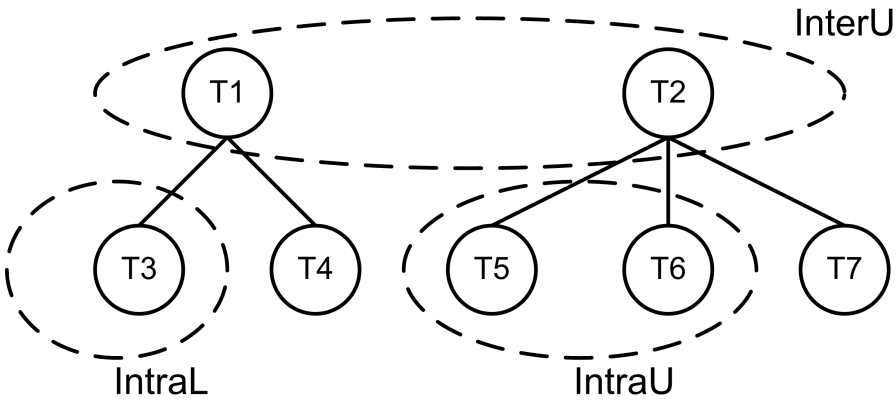


**Figure 4.6.** The three types of observed document pair similarities.

### 4.3.2 Searching for similar documents

The search for similar documents is evaluated together with the document extension in the following way: a small set of documents is selected from an arbitrary lower level topic and they are considered to be the base documents. Using these documents, a search for similar documents is started on the remaining part of the testing document set. As documents of related topics are considered loosely similar, the resulting set of selected documents is evaluated for precision and recall using their upper level topics only. By using lower level topics for the evaluation, document extension would drastically decrease the precision by making documents from other lower level topics similar. Although evaluation on upper level topics decreases the recall, the effect of document extension on detecting related documents can be observed on upper level.

Fig. 4.7 shows the results of a search for similar documents using the original document representations (shown in Fig. 4.11 in form of a document similarity matrix too) and Fig. 4.8 presents the results with extended document representations. The threshold (minimal number of common keywords for selection) is defined by the user. The measurements are performed for various base document numbers

(1, 5, 10, 15 and 20 base documents), and the precision and recall of the search is calculated as a function of the threshold.

The results confirm that increasing the threshold increases the precision and lowers the recall. Intuitive threshold settings such as "many documents" and "strict similarity" could mean threshold values for example 1 and 3 respectively. The increasing number of base documents increases the set of used keywords, thus it increases the recall but it makes more chances for misclassifications which lowers the precision. The significant recall increment due to document extension is confirmed by the results. This is a consequence of increasing the number of loosely related documents having similarity measure above the threshold. The degradation of precision is acceptable for small base document numbers. For more base documents, a stronger precision decrement is observable which is caused by the added RGC keywords and their additional chances to cause false selection. This can be compensated by increasing the threshold if many documents are stored on the mobile device.

Fig. 4.9 presents a comparison of mean performances of the searches using document extension with various RGCF learning methods or no document extension at all. More additional keywords (higher distance limit in the WordNet based method) obviously increase recall and decrease precision. The KCo method allows slightly higher precision than adding the synonyms based on WordNet. It should be emphasized that the effect of the KCo method is similar to the WordNet based extension with zero limit (synonyms only).

Table 4.3 shows the keywords of a concrete document about space shuttles. Table 4.4 shows the keywords added to that document during document extension using the KCo method. Documents containing the additional keywords might have a *loose relationship* to the content of the original document.

**Table 4.3.** Example for keywords representing a concrete document about space shuttles.

| |
|---|
| earth, access, protection, mass, landing, os, proposed, schedule, km, planned, fly, adams, bursts, evidence, orbital, space, universe, electrical, mars, predict, earth, vehicle, houston, training, scientific, baltimore, gravity, human, receiver, propulsion, thermal, engines, stanford, sky, satellite, nasa, mission, flight, bases, air, age, rocket, planets, launched, safety, solar, flight... |

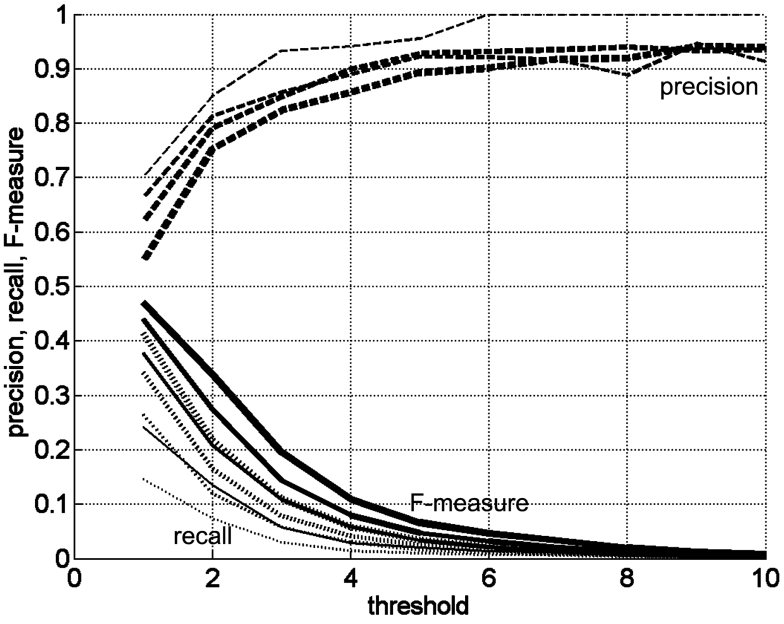


**Figure 4.7.** Evaluation of the search for similar documents in the data set 20 Newsgroups *without* document extension. The precision and recall of the selection is presented as a function of the threshold for base document numbers 1, 5, 10, and 20 (represented by line width in increasing order respectively). Results were evaluated on the upper level topics.

**Table 4.4.** Keywords added to the representation of the document on space shuttles during document extension using unsupervised (KCo) RGCF learning.

| |
|---|
| project, sci, phase, science, elements, objects, probe, radar, fuel, toronto, planet, zoo, cloud, solar, kelvin, henry, antenna, probes |

The results of the similarity search are different from the baseline measurements (Table 3.5 on page 48), as these results were achieved by using only the keywords of the base documents, and they were evaluated on the upper level topics for the better observation of the document extension. These differences lead to significantly lower recall and higher precision. But by comparing the results with and without document extension, the advantages of the document extension are clearly visible. Beside these, according to Table 3.1 (page 41), the similarity search requires only a few bytes per base document stored on the mobile device, and the size is independent of the number of topics (except the influence of topic numbers on keyword list sizes). The comparison of a remote document to the

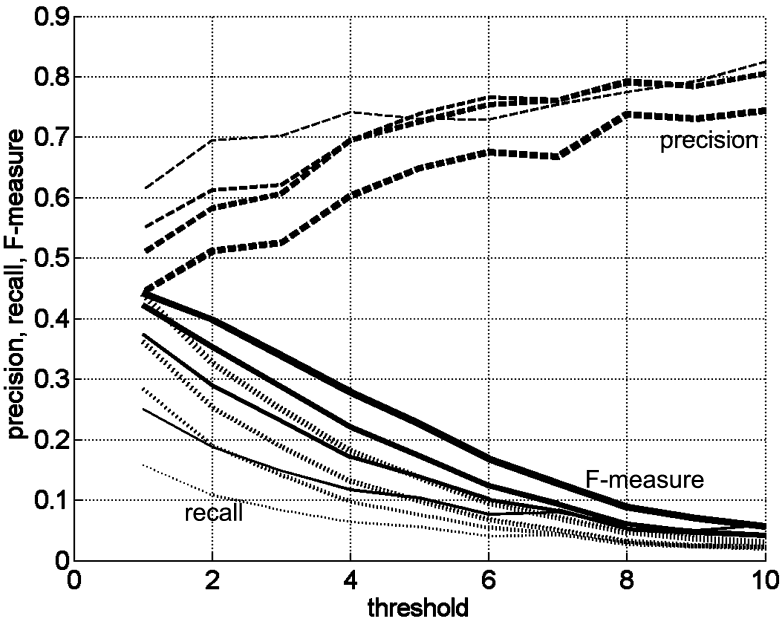


**Figure 4.8.** Evaluation of the search for similar documents in the data set 20 Newsgroups *with* document extension using WordNet based RGCF learning with depth limit 1. Results were evaluated on the upper level topics.

local ones requires only the transmission of the remote compact document topic representation having the size of about 10-20 bytes.

### 4.3.3 Number of keywords in a document

This measurement is a small calculation on the keyword numbers of the documents. As the user defined similarity threshold is the minimal number of common keywords, the number of keywords in a document is an important question. Fig. 4.10 presents the histogram of the keyword numbers of the documents. 20% of the documents does not contain any keywords, 66% contains few (maximum 5) keywords and 14% of the documents contain more than 5 keywords. The main reason for the low keyword numbers is the content of the data set: 20 Newsgroups contains Usenet submissions, which are usually short, just like most e-mails. These results confirm the significant decrement in recall if the threshold is increased.

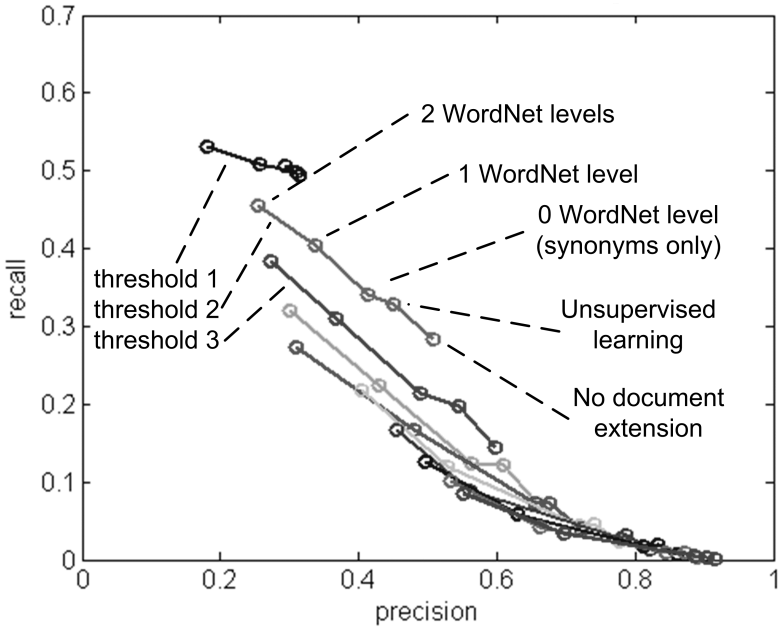


**Figure 4.9.** Performance of similar document search in terms of precision and recall using various RGCF learning methods for document extension, and threshold values between 1 and 10 on the 20 Newsgroups data set.

### 4.3.4 Keywords causing false similarities

This measurement investigates the keywords causing false high similarities during the search for similar documents. From the document similarity matrix presented in Fig. 4.11, the region where documents from topics *hardware.PC* and *science.electronics* are compared, were selected. This is shown in Fig. 4.12. It is clear, that the misclassifications are caused by a few number of documents in the *science.electronics* topic which are similar to many documents in the topic *hardware.PC*. This is indicated by the columns with many markers.

Among these documents, one was selected for detailed investigation: document *52729* (filename in 20 Newsgroups) has more than 3 common keywords with 6 documents from the other topic. The keywords appearing in the document and thus leading to the false similarity measures are *card*, *bus*, *cpu*, *dx* and *motherboard*. Document *52729* is the answer on a question on the ISA bus of an IBM PC computer. Considering the content, the misclassification is not surprising as the document could have easily fit into the other topic as well.

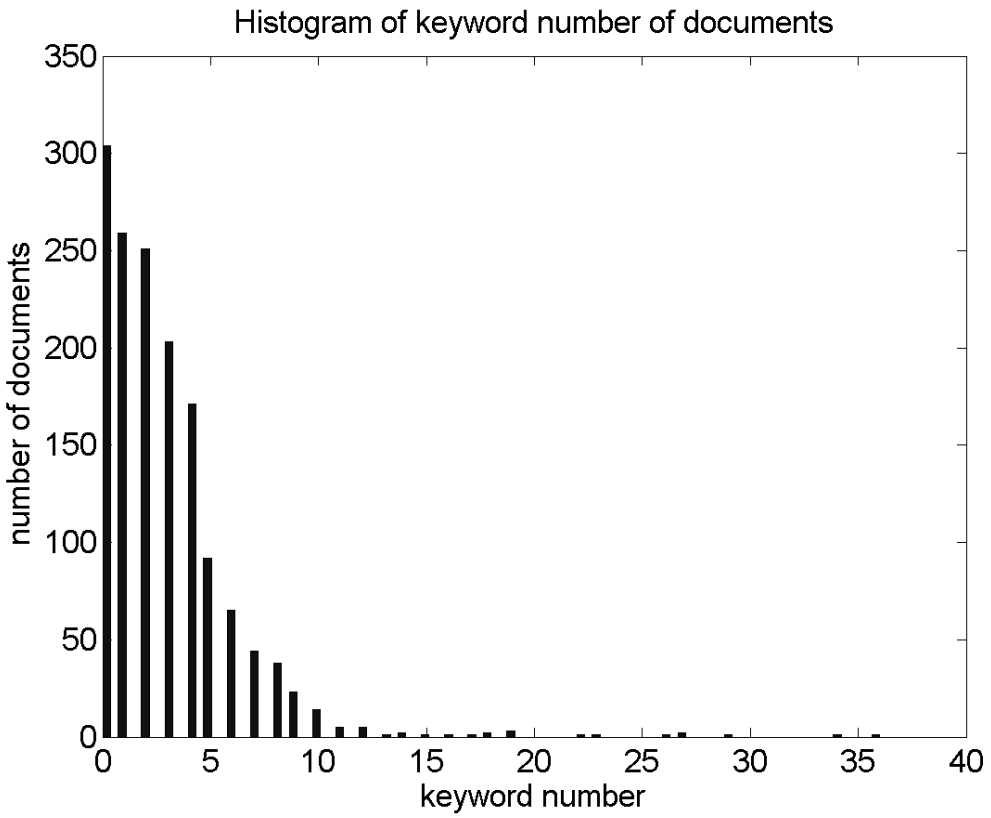


**Figure 4.10.** Histogram of keyword numbers of the documents in a randomly chosen testing subset of the 20 Newsgroups data set.

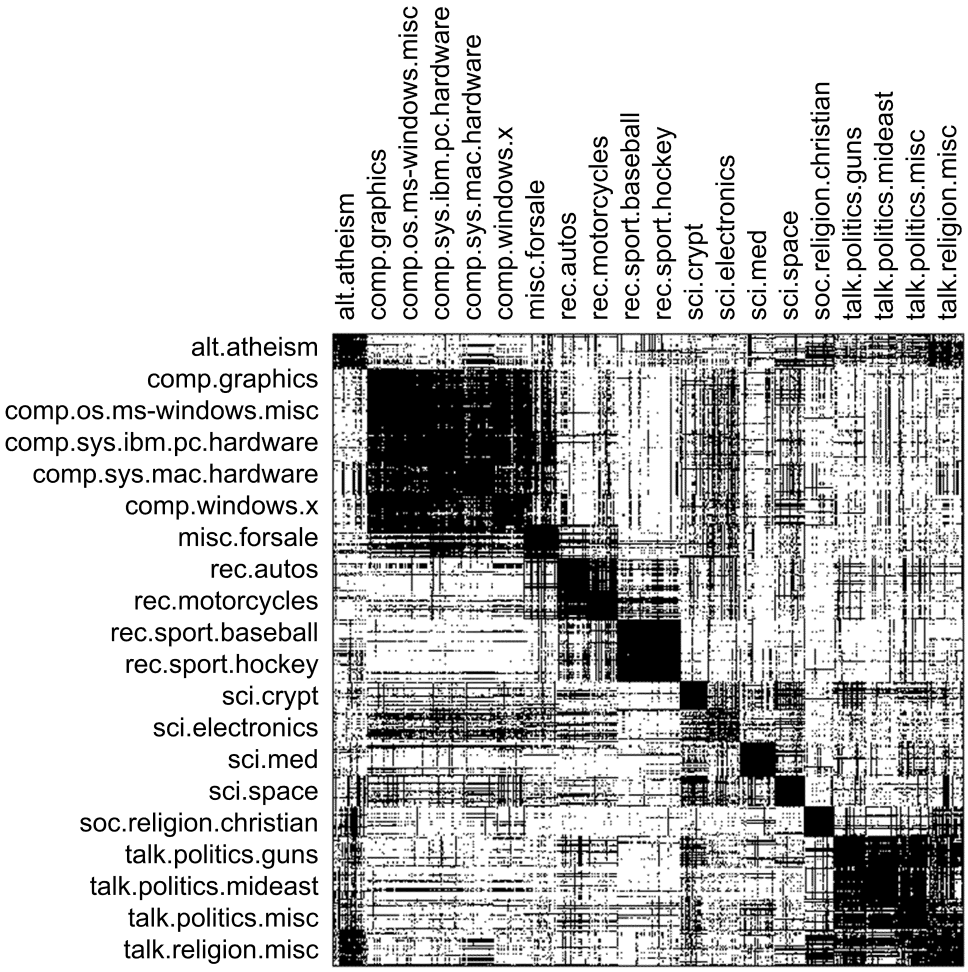


**Figure 4.11.** Document similarity matrix of the whole 20 Newsgroups data set. Topics can be seen as rectangles along the main diagonal as documents are ordered by topic along the axes. Dots indicate nonzero similarity measure between the documents corresponding to the row and column.

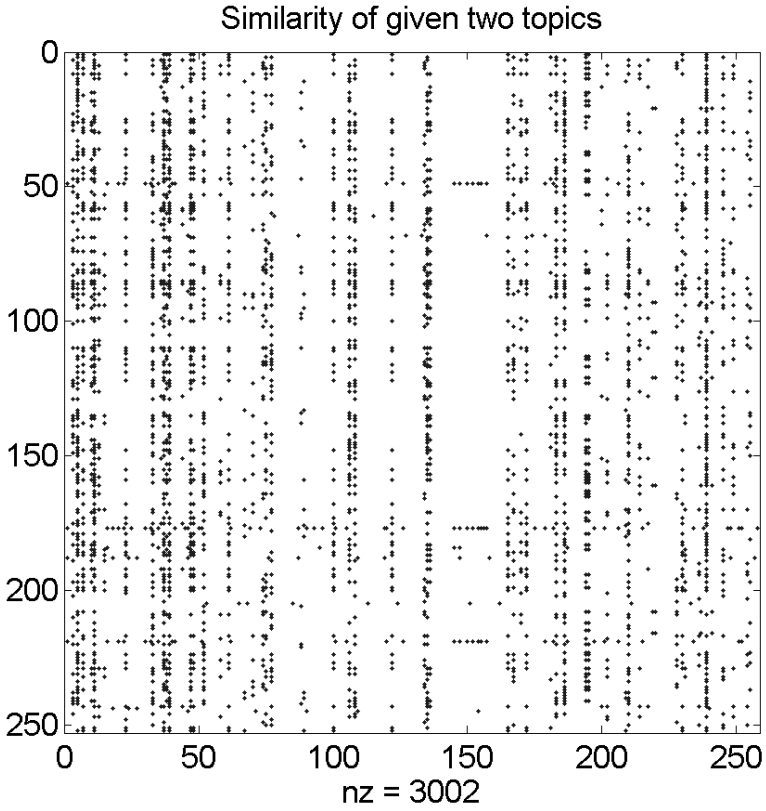


**Figure 4.12.** Similarity matrix of topics *hardware.PC* (rows) and *science.electronics* (columns).

Chapter 5

# Two-level topic identification and cascading

This chapter proposes two ways how classifier ensembles can improve the performance of the methods proposed in the previous chapters. The first one creates a two-level topic identification ensemble which reduces the number of keyword lists the mobile device has to check during the topic identification. The second one improves the recall of similarity search by creating further selectors running on documents not selected by the previous classifier levels. This allows selecting further similar documents, although it decreases the precision due to additional chances for misclassifications.

## 5.1 Two-level topic identification using topic sets

The topic identification aims at finding the topic which has the most common keywords with a given document. The keyword list of the best matching topic will be used to represent the document for other devices. A simple solution would be to calculate the number of common keywords with every available keyword list and find the best matching one. This is done by the Most Keywords classification in its form described earlier. The drawback of this solution would be the high number of keyword lists: if the mobile device has to use all the keyword lists for the topic identification, it has to retrieve all possible keyword lists which decreases the scalability of the solution. In order to reduce the number of keyword lists the topic

identification process has to check, a two-level classifier ensemble is introduced which uses sets of similar topics on the upper level to approximate the topic of a document. Using these topic sets, the number of checked topics can be limited by skipping the ones which have very low probability to be the best fitting one. Theoretically there is no limitation for the number of levels if the big number of topics makes more levels reasonable.

The structure of the solution is the following: during the training of the system, multiple initial topic sets are created. All of these are evaluated with a simulated document classification. This allows the removal of the useless topic sets such as those that cover almost every topic or those achieving very low recall. In the last step of the training, keyword lists are created for the topic sets using the PKS algorithm described earlier.

Fig. 5.1 shows the architecture of the classifier: documents are first compared with the keyword lists of the topic sets. The topic sets having at least one common keyword with the document are collected (these topic sets are the *triggered topic sets*) and finally, only the keyword lists of topics in triggered topic sets (*triggered topics*) are compared to the document. This solution is similar to a decision tree, but not only the topics of the best matching topic set are checked, in order to decrease the probability of misclassification due to a false decision on the topic set level.

The key goal of the topic set based topic identification is to limit the number of keyword lists which have to be checked during topic identification, while not decreasing the classification performance due to the internal classifications using the topic sets.

### 5.1.1 Creating easy-to-identify topic sets

Topic sets are created in three steps: initial topic sets are generated, initial topic sets are evaluated (and modified/removed if necessary), and further topic sets are created for every topic not covered by the topic sets.

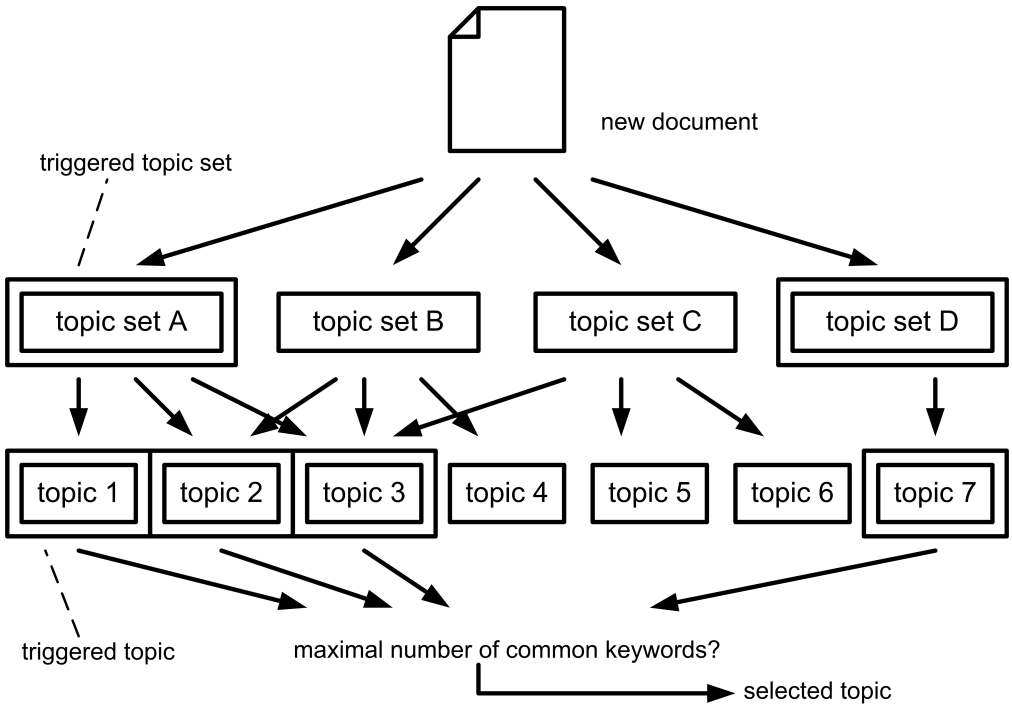


**Figure 5.1.** Topic sets. Only triggered topics (topics of triggered topic sets) are checked during topic identification. In this case, the total number of checked keyword lists is 6, although the number of topics is 7.

#### 5.1.1.1 Creating initial topic sets

The first step is to identify sets of topics which are easy to identify. A brute force method could be the generation of every possible subset of the topics, and let the topic set evaluation step discard the bad ones. This is not applicable due to the exponential growing number of subsets. The key idea behind the *F-measure based Topic Set Creation* (FTSC) is the identification of the topic set for every $w$ word, which is the easiest to identify using only $w$.

The F-measure based Topic Set Creation algorithm (FTSC) retrieves for every word the set of topics which that word can select with the highest F-measure. With other words, if one would select all documents the given word appears in, which target topic set would get the highest F-measure.

Using the individual F-measure (defined on page 24), every $w$ word is assigned the set of topics for which $w$ achieves the highest $iF$ individual F-measure (Fig. 5.2).

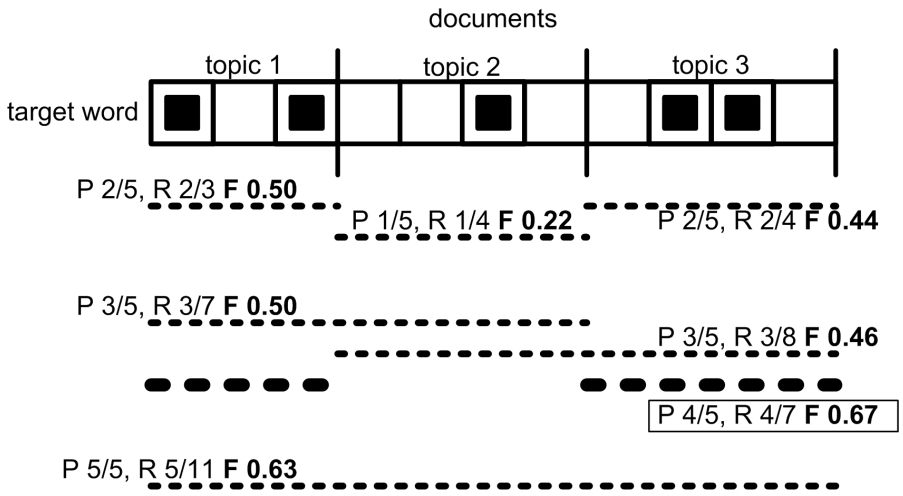


**Figure 5.2.** Example for calculating the easy-to-identify topic set of a given word. P, R and F stand for precision, recall and F-measure respectively. In this example, the individual F-measure is maximized by a topic set containing topics 1 and 3.

**Definition 5.1** (Easy-to-identify topic set of a word)**.** The easy-to-identify topic set of the word $w$ is

$$\mathbb{T}^{opt}(w) = \arg\max_{\mathbb{A}} \{iF(w, \mathbb{A}) : \mathbb{A} \subseteq \mathbb{T}\} \tag{5.1}$$

where $iF(w, \mathbb{A})$ is the individual F-measure of the word $w$ with respect to the topic set $\mathbb{A}$ as target topic and $\mathbb{T}$ is the set of all topics.

Individual precision is not suitable in this case as considering more topics as target can not decrease precision. The highest precision is achieved if all the topics are target topics. Individual recall is unsuitable as well because it does not take the precision into consideration which is still very important.

The FTSC algorithm is searching for the topic set $\mathbb{T}^{opt}(w)$ for every $w$ word in a greedy way: it adds the $T$ topics to the topic set in descending $iF(w, T)$ order until the individual F-measure is maximized (Alg. 5.1).

The set of initial topic sets consists of all topic sets returned by FTSC executed for every $w$ word (without duplicates of course).

Although FTSC is a greedy algorithm, it achieves optimal solution if the a-priori topic probabilities are equal for all topics:

**Algorithm 5.1** F-measure based Topic Set Creation

Input: $w$ word, $\mathbb{T}$ set of all topics
// Get $iF$ for every topic
// Order topics in $\mathbb{T}$ in descending $iF$-order
$\mathbb{L} = sort(\mathbb{T}, iF(w,T), 'desc')$ // $\mathbb{L}$ is an ordered list of topics.
// Choose $n$ so that the first $n$ topics maximize $iF$
$n = \arg\max_x \{iF(w, \mathbb{L}(1..x))\}$ // Return set of the first $n$ topics.
Output: $\mathbb{L}(1..n)$

**Proposition 5.2** (Optimality of FTSC)**.** *The FTSC algorithm selects the optimal topic set $\mathbb{T}^{opt}(w)$ for every word if the a-priori topic probabilities are equal for all topics.*

**Proof:** Let's consider a given word which selects documents of a given target topic. Let $c$ be the number of correctly selected documents, $s$ the number of selected documents, and $t$ the number of documents in the target topic. The precision is $p = c/s$ and recall is $r = c/t$.

$$F = \frac{2 \cdot p \cdot r}{p + r} = \frac{2 \cdot c \cdot c}{s \cdot t(c/s + c/t)} = \frac{2 \cdot c}{t + s} \tag{5.2}$$

If the optimal topic set for a given $w$ word is searched for, the $s$ number of selected documents is constant. The $t$ target document number is assumed to be the same for every topic (assuming equal a-priori topic probability). A topic set containing $|\mathbb{T}| = n$ topics and maximizing F-measure is maximizing

$$F = \frac{2 \cdot \sum_{T \in \mathbb{T}} c_T}{n \cdot t + s} \tag{5.3}$$

where $c_T$ is the number of selected documents in the topic $T$. Due to the constant denominator, $\mathbb{T}$ has to maximize $\sum_{T \in \mathbb{T}} c_T$. Considering that the individual F-measure of $w$ in every topic is

$$iF(w,T) = \frac{2 \cdot c_T}{t + s} \tag{5.4}$$

where the denominator is topic independent, $\mathbb{T}$ has to contain the $n$ topics with the highest individual F-measure regarding $w$. If the topics are added to $\mathbb{T}$ in

decreasing individual F-measure order, the $\mathbb{T}$ maximizing F-measure is a global optimum.□

If the topics do not have the same number of documents in the training set, the best choice of topic to be added next to the topic set depends on the set of already added topics.

Let's suppose that there are two topics which can be added to $\mathbb{T}$: $T_1$ and $T_2$, and $c_1 > c_2$, so $T_1$ seems to be a better choice to add. In the following, I will show a condition for the $t_2$ number of documents in topic $T_2$ which assures that despite $t_1 \neq t_2$, $T_1$ increases F-measure more than $T_2$ does. If $t_2 = t_1 + \epsilon$, $T_1$ is the better choice if

$$\frac{2(\sum_{T\in\mathbb{T}} c_T + c_1)}{\sum_{T\in\mathbb{T}} t_T + t_1 + s} > \frac{2(\sum_{T\in\mathbb{T}} c_T + c_2)}{\sum_{T\in\mathbb{T}} t_T + t_1 + \epsilon + s} \tag{5.5}$$

By expressing $\epsilon$, we get

$$\epsilon > \frac{(c_2 - c_1)(\sum_{T\in\mathbb{T}} t_T + t_1 + s)}{\sum_{T\in\mathbb{T}} c_T + c_1} \tag{5.6}$$

As $c_2 < c_1$ due to the starting condition, $\epsilon_{min}$ is a negative value. If $t_1 = t_2$, the condition is always satisfied. Otherwise, the $c_T$ based ordering of topics may lead to lower F-measure values. An alternative solution would be to calculate the F-measure in every step for every possible topics and choose the one leading to the highest F-measure. In the section of experimental results I will compare this solution to the presented one, and show, that the quality of the results is not significantly lower, even for a data set having different document numbers in the topics.

It should be noted that even if the FTSC algorithm does not find the optimal topic set for some words, the goal is to create topic sets which can be used to train a good two-level classifier system. Based on the results, this is successfully achieved.

#### 5.1.1.2 Evaluating and modifying initial topic sets

After the initial topic sets have been created, they have to be evaluated because some of them will not be useful, like a topic set covering all topics. In order to

use (or evaluate) a topic set, its keyword list has to be created. This is done using PKS just as it would be a single topic: PKS searches for keywords which appear often in the documents of the topic set and rarely in the documents outside the topic set.

The evaluation phase evaluates every initial topic set. It creates keyword lists to distinguish them using PKS and simulates the selection of documents in the training set. The keyword list created for an ideal topic set would select exactly the documents of the topics contained in the topic set. The precision and recall of the result is calculated and topic sets fulfilling the following conditions are preserved:

1. The topic set cannot contain topics for which too few document were selected. Such topics are removed from the topic set because they have too low recall. The topic set would unnecessarily trigger these topics every time the topic set is triggered. In the experiments, every topic had to have a 0.5 recall inside the topic set.

2. Sufficiently high precision and recall. If a topic set has too low precision or recall, it is discarded. In the experiments, the minimal limit was set to 0.5 for both precision and recall. It should be noted that the previous condition already guarantees the minimal necessary recall.

3. The topic set has to have topics satisfying the first condition. If all the topics of a topic set are removed due to the first condition, the empty topic set is removed entirely.

4. The topic set may not cover more than 50% of the topics. Otherwise there would be topic sets covering almost all topics using very common words. Such topic sets are believed to be useless because they trigger almost every topic and thus do not support the exclusion of topics having minimal chance to be the best fitting one.

#### 5.1.1.3 Creating additional topic sets

In the last step of the topic set creation, topics not covered by any remaining topic sets are moved into a separate topic set created for each of them individually.

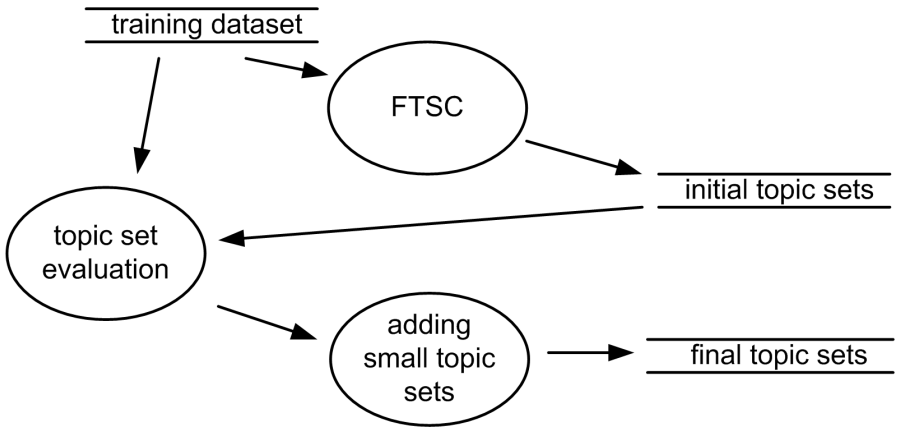


**Figure 5.3.** Data flow diagram of the topic set creation. The training set is used to create the initial topic sets and the evaluation phase creates the final topic sets by modifying or removing worse topic sets and adding new ones if necessary.

These additional topic sets contain only one topic. These topic sets are called *small topic sets* and the ones containing multiple topics are called *big topic sets*.

**Definition 5.3** (small and big topic sets)**.** A topic set is a small topic set, if it contains exactly one topic. Otherwise, it is a big topic set.

The data flow diagram of the topic set creation algorithm is presented in Fig. 5.3 and the algorithm is summarized in Alg. 5.2.

**Algorithm 5.2** Topic set creation for document topic identification

Input: $\mathbb{T}$ set of all topics, $W$ set of all words
// Create initial topic sets
$\mathbb{TS} = \bigcup_{w \in W} \{FTSC(w, \mathbb{T})\}$
// Get topic sets fulfilling the evaluation conditions
$\mathbb{TS} = GetSuitableTopicSets(\mathbb{TS})$
$\mathbb{C} = \bigcup_{\mathbb{F} \in \mathbb{TS}} \mathbb{F}$ // Create set of covered topics
**for** $T \in \mathbb{T} \backslash \mathbb{C}$ **do**
  $\mathbb{TS} = \mathbb{TS} \cup \{T\}$ // Create small topic sets for not covered topics
Output: $\mathbb{TS}$ set of topic sets

#### 5.1.1.4 Training the classifier ensemble

After the topic sets have been created, a keyword list is created for them using PKS just as they would be the topics. The second level of the ensemble is trained just as there would be no topic sets: a keyword list is created for every topic.

### 5.1.2 Using the classifier ensemble

After the training of the classifier ensemble (creating the keyword lists for all topic sets and topics), the classifier is ready to identify the topic of new documents according to Alg. 5.3.

If the topic of a new document has to be identified, it is first compared with the keyword lists of the topic sets. If the document has at least one common keyword with a topic set (that means that the topic set is triggered) the topics contained in the topic set are all triggered. After checking every topic set, the best fitting topic specific keyword list is searched for just as in the case without the topic sets, however not triggered topics are not checked because they are considered to be "hopeless".

**Definition 5.4** (Set of triggered topic sets)**.** Given a $d$ document, the $Trig(d, \mathbb{TS})$ set of triggered topic sets contains the topic sets having common keyword with the document.

$$Trig(d) = \{\mathbb{TS} : K_{\mathbb{TS}} \cap d \neq \emptyset\} \tag{5.7}$$

**Algorithm 5.3** Using topic sets for topic identification

Input: $d$ document
$\mathbb{TR} = Trig(d, \mathbb{TS})$ // Get set of triggered topic sets
$\mathbb{R} = \bigcup_{\mathbb{TRS} \in \mathbb{TR}} \mathbb{TRS}$ // Get topics in triggered topic sets
// Get the topic with the most keywords, among the triggered topics
Output: $\hat{\mathcal{T}}(d) = MKw(d, \mathbb{R})$

Unfortunately there are always topics which are not covered by the initial topic sets and they have to be placed in a small topic set. As this may increase the number of topic sets significantly, the following rule has been introduced: if a document triggers at least an $mb$ minimal number of *big topic sets*, the *small topic sets* are not checked because the real topic of the document is assumed to be covered by the big topic sets. This extension is called *Small Sets on Demand* (SSD) because small topic sets are only checked if there seems to be a need for it. The topic identification extended with SSD is presented in Alg. 5.4.

**Algorithm 5.4** Using topic sets for topic identification with SSD. $\mathbb{TS}^*$ represents the set of big topic sets.

Input: $d$ document
$\mathbb{TR} = Trig(d, \mathbb{TS}^*)$ // Get set of triggered big topic sets
**if** $|\mathbb{TR}| < mb$ **then**
  $\mathbb{TR} = Trig(d, \mathbb{TS})$ // Checking all topic sets
// Get topics in triggered topic sets
$\mathbb{R} = \bigcup_{\mathbb{TRS} \in \mathbb{TR}} \mathbb{TRS}$
// Get the topic with the most keywords, among the triggered topics
Output: $\hat{\mathcal{T}}(d) = MKw(d, \mathbb{R})$

## 5.2 Cascade structure for similarity search

Similarity search cascade structures are designed to improve the low recall caused by the many document pairs not having any common keywords. The key idea is the training of a selector specialized for the documents not having any keywords in the first similarity search. The second and any later levels are trained so that "easy" cases (recognized by previous levels) are already removed from the training set. The aim of the later classifier levels is to correctly select additional documents similar to the base documents. As the proposed solution creates multiple 1-class classifiers, the increasing resource need makes it less applicable in low-resource environments. But if the required resources are available, similarity search cascade structures are suitable to further increase the recall of the search.

**Definition 5.5** (Allowed and Excluded topic set)**.** $\mathbb{A}$ is the set of topics which have base documents: $T \in \mathbb{A} \leftrightarrow \exists d \in B \cap T$ where $B$ is the set of base documents. The set of excluded topics contains topics which do not have base documents: $\mathbb{E} = \overline{\mathbb{A}}$.

The aim of a cascade structure is the following: in every cascade element, the selector is trained assuming that every document arriving to its input was not selected by any selectors of previous levels, nor for allowed, neither for excluded topics. This means, that the document had no common keyword with any previously used keyword lists.

Alg. 5.5 presents the training of a cascade structure. It consists of successive keyword list creations for both allowed and excluded topics. After every training, the selected documents are removed, so the next level is trained only on the documents not selected by the previous levels. The documents recognized to belong

to excluded topics are removed too. This is the advantage of the next level: there are many off-topic documents which it does not need to care about. The results of the classifier training are the two set of keyword lists: one for the recognition of similar documents and one for the recognition of off-topic documents.

**Algorithm 5.5** Training a similarity search cascade structure

Input: $\mathbb{A}$ set of allowed topics, $\mathbb{E}$ set of excluded topics, $D$ set of all documents, $n$ number of cascade levels, $B$ set of base documents
$D^{(1)} = D$ // First element becomes all documents
$b = \bigcup_{d \in B} d$ // Merge all base documents into one word set.
**for** $i = 1$ to $n$ **do**
  $K_b^{(i)} = b \cap \bigcup_{T \in \mathbb{A}} PKS(T, D^{(i)})$ // Create keyword lists for all allowed topics. Use only keywords appearing in the base documents.
  $K_e^{(i)} = \bigcup_{T \in \mathbb{E}} PKS(T, D^{(i)})$ // Create keyword lists for all excluded topics.
  $D^{(i+1)} = D^{(i)} \backslash (D^{(i)} \cap K_b^{(i)}) \backslash (D^{(i)} \cap K_e^{(i)})$ // Remove recognized similar documents and the ones with recognized excluded topic.
Output: all $K_b^{(i)}$ and $K_e^{(i)}$ keyword lists.

The use of a classifier is presented in Alg. 5.6. Every level begins with the search for documents similar to the base documents. This comparison is based on the keyword list $K_b^{(i)}$ containing the keywords for allowed topics, but only those one appearing in base documents, too. Before moving to the next level, documents are checked for excluded topics, too. All documents similar either to base documents, or recognized to belong to excluded topics, are removed from the document set before moving to the next cascade level.

**Algorithm 5.6** Using a similarity search cascade structure

Input: $K_b^{(i)}$ and $K_e^{(i)}$ keyword lists, $D$ set of remote documents
$R = \emptyset$ // Result set of similar documents.
**for** $i = 1$ to $n$ **do**
  $R = R \cup S(K_b^{(i)}, D)$ // Get selected documents from the remaining document set
  $D = D \backslash S(K_b^{(i)}, D) \backslash S(K_e^{(i)}, D)$ // Remove selected or excluded documents.
Output: $R$ set of similar remote documents.

It should be noted that the cascade levels require the documents to be represented using different keyword lists. This means that the remote mobile devices have to be asked to send multiple document representations, one for every cascade

level. The storage of multiple representations for the same document is the cause of the increased resource need mentioned in the introduction.

As an extension of the cascade structures, the idea of similarity thresholds, introduced in connection with the similarity search, can be introduced for the selectors in the cascade elements as well:

**Definition 5.6** (Threshold and exclusion threshold)**.** Threshold and exclusion threshold are the user defined minimal similarity limits used by the selectors $S(K_b^{(i)}, D)$ and $S(K_e^{(i)}, D)$ respectively.

In order to assure that the similarity of a remote document to the base documents cannot decrease while moving to the next cascade level, $K_b^{(i-1)} \subseteq K_b^{(i)}$ for every $i > 1$. This way, if a remote document is just under the threshold in one level, it needs only a few more common keywords in the next level to be similar enough for selection.

The successive cascade levels are specialized on the cases not recognized by the previous ones. The following proposition states that there are words which are better keywords in the later levels, as in the earlier ones.

**Proposition 5.7** (Increasing *iprec* of words in cascades)**.** *By training the next level of a similarity search cascade structure, there can be words with increasing individual precision.*

**Proof**: Given a target topic $T$, if a word $w$ appears in $c$ target topic documents and $f$ off-topics documents, $iprec(w, T) = c/(c + f)$. In a marginal case, if this *iprec* is not enough to be keyword, but the $S(K_e^{(i)}, D)$ selector of the current cascade level removes all $f$ off-topic documents containing $w$, in the next level, $iprec(w, T)' = c/(c + 0) = 1$ which indicates a perfect keyword.□

Although all the off-topic documents containing $w$ are unlikely to be removed, *iprec* can increase if enough of them are removed.

## 5.3 Comparison to related work

The techniques proposed in my third thesis are related to classifier ensembles. The most common approaches are AdaBoost [Freund and Schapire, 1995], decision

trees, and decision lists [Oded Maimon, 2005] for example. The speciality of my two level topic identification method over conventional decision trees is that it takes the uncertainty of classification results into account and does not limit the successive decisions to the best matching directions. Beside this, the training method contains ideas focusing on the selection capabilities of individual words for creating the easy-to-separate topic sets.

The cascade structures are related to decision lists, as they contain 1-class classifiers after each other. One difference is that the later classifiers are specialized on the cases hard for the previous levels, which is a property related to boosting methods. The difference to boosting is mainly the goal of the classification: the selection of documents on one single topic. This makes the specialization of later classifier levels more focused by removing the surely off-topic documents from the training set.

Beside these, both proposed methods can also be approached from the point of data partitioning [Dong and Han, 2005], as documents are grouped into sets easier to classify. (Although not necessarily into disjunct sets in this case.) In the two-level topic identification, this is done by creating the easy-to-identify topic sets, and in cascade structures this is done by removing surely off-topic documents in the beginning of the next classifier level.

## 5.4 Experimental results

In this section, several experimental results are presented in connection with both topic set based classifier ensembles and similarity search cascade structures.

### 5.4.1 Experimental results with topic sets

This section presents measurement results according to various aspects of the topic set based document topic identification. The measurements were performed using the commonly used data sets 20 Newsgroups and Reuters Corpus Volume 1 (RCV1, LYRL2004 split).

First, results according to the classifier ensemble used for topic identification in the 20 Newsgroups data set are presented, because the interpretation is easier with this data set. After that, the evaluation on RCV1 is discussed.

#### 5.4.1.1 Evaluation of the topic set based classifier ensemble

The most important condition the two level classifier has to satisfy is the minimal degradation in the classification performance. Table 5.1 presents the classification performance of the system without the application of topic sets, with topic sets but without SSD and with SSD using $mb$ minimal triggered big topic set number 1 and 2. The data set has 20 topics, 5 big and 8 small topic sets were created.

Based on the results, the following conclusions can be made:

- The case without topic sets is the baseline measurement as this is a simple classification using the keyword lists created with PKS.
- Using topic sets does not significantly influence the classification results, but without SSD, all the 13 topic sets are checked for every document, followed by the check of the triggered topics. The number of triggered topics (mean value is 3.12) is presented in Fig. 5.4. This means that around 16-17 keyword lists (13 topic sets and 3-4 triggered topics) are still compared to the documents which is almost the number of topics (which is 20), thus it does not lead to significant improvement.
- By activating SSD, the classification performance decreases slightly because some documents belong to topics in small topic sets but still trigger enough big topic sets which makes their real topics not checked. But for an exchange, with $mb = 2$, altogether 54% of the documents with topics in big topic sets are classified without checking the small topic sets (thus checking only 5+3.12 keyword lists in average for this 54%).
- SSD with $mb = 1$ decreased the recall slightly more but it made the small topic sets skipped for every document which had a real topic in one of the big topic sets.

If a user stores documents belonging to big topic sets on the mobile device, setting $mb = 1$ can decrease the number of keyword lists compared to the document

**Table 5.1.** Classification results without topic sets, with topic sets and no SSD, and with SSD using $mb$ 1 and 2.

| | precision | recall | F-measure |
|---|---|---|---|
| without topic sets | 0.61 | 0.45 | 0.50 |
| with topic sets | 0.65 | 0.42 | 0.50 |
| SSD ($mb = 2$) | 0.64 | 0.41 | 0.49 |
| SSD ($mb = 1$) | 0.65 | 0.39 | 0.47 |

during topic identification from 20 (no topic sets, no SSD) to 8.12 in average. If the small topic sets are needed as well, this value is 16.12 in average.

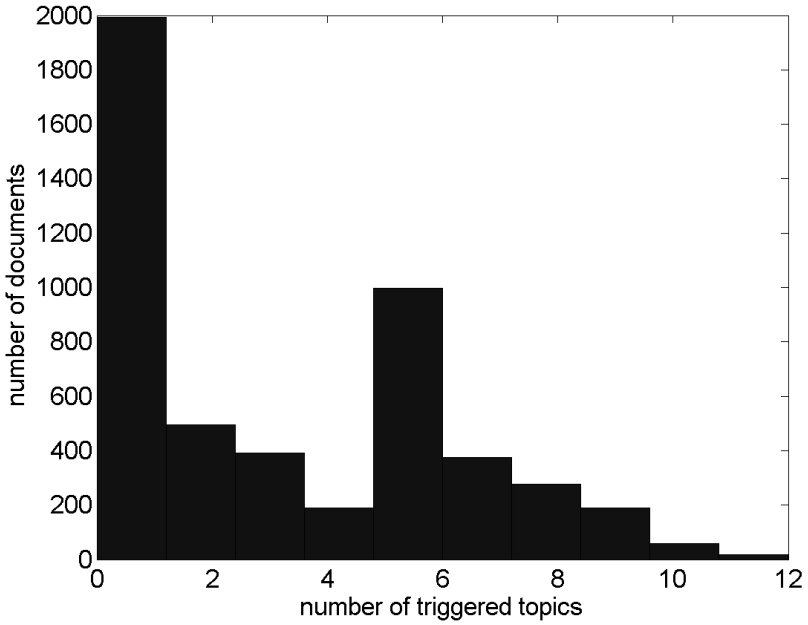


**Figure 5.4.** Histogram of the number of topics triggered by a document. The mean value is 3.12 topics.

Details about the topic sets are presented in Table 5.2. Some topic sets seem to be reasonable based on the name of the contained topics like merging *atheism* and *religion.christian*. Others may look strange in the first approach but after having a look at some keywords assigned to these topic sets, a connection can be recognized. Topic set 1 is based on connections with security and nation names, topic set 2 is about sports but nation names lead to the topic on the middle east as well. Topic set 3 is clearly about X-servers and MS-Windows, topic set 4 is based on security aspects of politics and computer science, and finally topic set 5

**Table 5.2.** Topic sets. Quality is shown in terms of precision (P) and recall (R). Topics not covered by any sets mentioned in this table have their own small topic set.

| ID | contained topics | quality | example keywords |
|---|---|---|---|
| 1 | soc.religion.christian<br>talk.politics.mideast<br>sci.crypt<br>sci.space | P 57<br>R 70 | christians church christ pgp soviet soul muslim escape turkish heaven spirit jews secret israeli secure arab israel orbit security mountain keys algorithm roads turks encryption |
| 2 | rec.sport.baseball<br>rec.sport.hockey<br>talk.politics.mideast | P 71<br>R 65 | teams team turkish israeli baseball wings arab israel league players season turks hockey layer fans nhl |
| 3 | comp.os.ms-windows<br>comp.windows.x | P 62<br>R 68 | server microsoft window windows motif |
| 4 | talk.politics.guns<br>rec.sport.hockey<br>sci.crypt<br>rec.autos<br>talk.politics.misc | P 63<br>R 69 | cars pgp citizens cup economic car gun criminal crime sw tax secret guns constitution clinton secure wings federal fbi police warrant security compound weapons keys agents enforcement pittsburgh hockey coverage encryption nhl |
| 5 | alt.atheism<br>soc.religion.christian | P 73<br>R 70 | atheist christians bible holy church god faith christianity christian christ belief morality jesus sin heaven |

is clearly about religions. Small topic sets are not mentioned here but every topic not covered by the presented topic sets is covered by a small topic set.

By evaluating the topic set creation method as a method for ordering topics into hierarchy, one can see that the created topic hierarchy is not the same as the original topic hierarchy of the 20 Newsgroups data set. The main reason for this difference is that the resulting "hierarchy" is created by merging topics which can be easier recognized using keywords if they are merged, than if they would have to be recognized separately. This is allowed by many common potential keywords shared between the documents of the topics. As there are many keywords, it is not surprising that some of them suggest different merging of topics than the merging defined by the original topic hierarchy of the data set. For example topic set 5 contains "alt.atheism" and "soc.religion.christian" together and it is reasonable as well, although the original hierarchy does not indicate this similarity.

The covering of topics by topic sets is visualized in Fig. 5.5. During the application of the system, documents of a given topic may trigger multiple topic sets.

The corresponding measurement results are presented in Fig. 5.6. The covering of the topic sets can be clearly recognized but there are false triggers as well. The average number of topic sets a document is triggering is 1.23, its histogram is shown in Fig. 5.7.

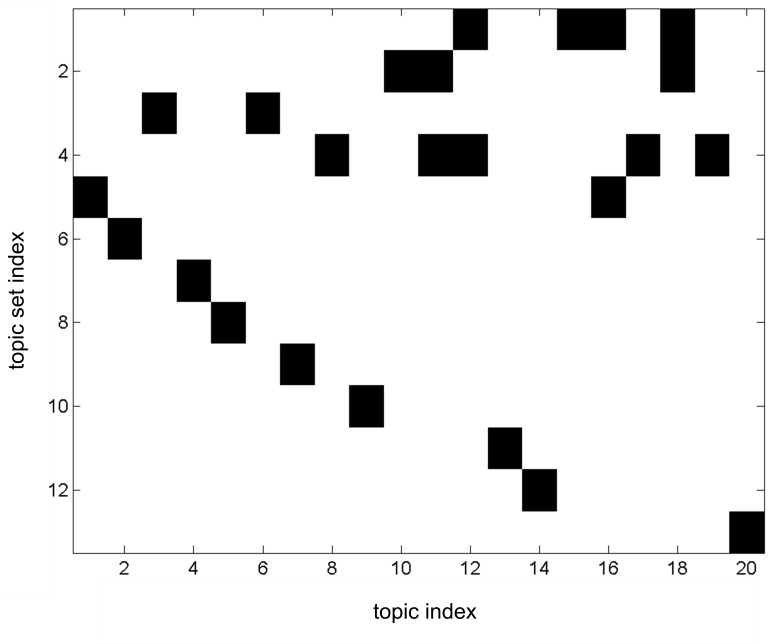


**Figure 5.5.** Topic sets retrieved for the 20 topics of the 20 Newsgroups data set.

#### 5.4.1.2 Evaluation on Reuters Corpus Volume 1

The 20 Newsgroups data set has only 20 topics. The Reuters Corpus Volume 1 (version 2) has 103 topics altogether and these are organized in a two-level hierarchy containing 4 topics on the upper level. The LYRL2004 split of the data set, which was used for the measurements, has an already prepared word-document matrix available on the World Wide Web. This prepared version of the data set has stemming already applied to it.

During the preparation of the data set, some topics containing too few documents were removed. 78 topics remained.

Exactly the same methods were applied to the RCV1 data set as to the 20 Newsgroups previously. The classification results were obtained without topic sets

**Table 5.3.** Classification results on the RCV1 data set.

| | precision | recall | F-measure |
|---|---|---|---|
| without topic sets | 0.64 | 0.41 | 0.47 |
| with topic sets SSD ($mb = 2$) | 0.62 | 0.47 | 0.52 |

and with topic sets using SSD with $mb = 2$. The results presented in Table 5.3 are similar to the ones for the 20 Newsgroups (Table. 5.1). The small decrease in performance with topic sets is caused by the imbalanced training set as the number of documents in the various topics in RCV1 is not the same.

As RCV1 has much more topics than 20 Newsgroups, so the capability of the topic sets to decrease the number of checked keyword lists is more significant: although there are 78 topics, the mean number of triggered topics is 39.7. If no small topics are required to check, only 12 big topic sets are checked and 4.4 of them are triggered by a document in average. This means that the classification of a document required the check of $12 + 39.7 = 51.7$ keyword lists in average, instead of 78.

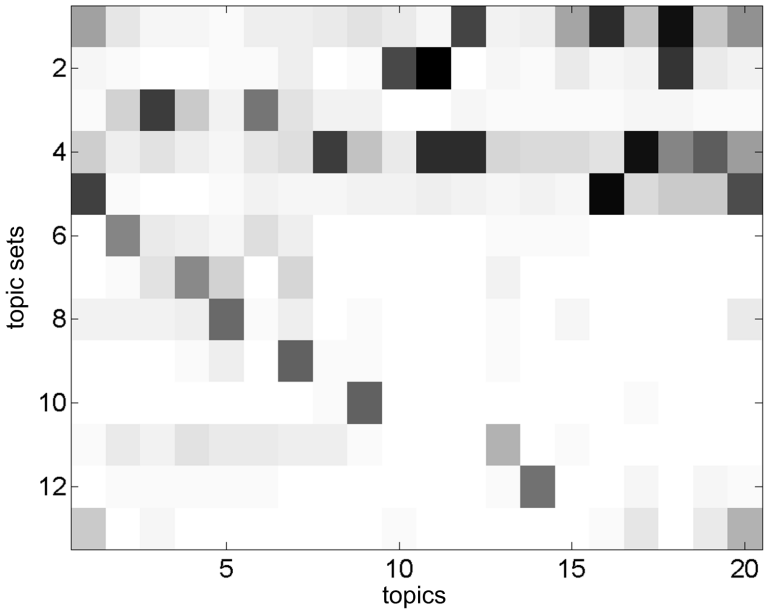


**Figure 5.6.** Triggering of topic sets. The more often a topic set is triggered by the documents of a topic, the darker is the rectangle corresponding to the (topic set;topic) pair. The measurement used SSD with $md = 2$.

Examples on the merged topics and keyword lists of the topic sets are presented in Tables 5.4 and 5.5. Due to the high number of topics, every topic set with all its keywords cannot be presented here. Incomplete lists are marked with "...". There are many words which are rare enough not to be discarded as stopwords but they imply topic sets containing lots of topics. This leads to some topic sets (ID 10, 11 and 12) which have too many topics and thus too many and very diverse keywords as well. Although they were triggered by over 80% of the documents, they do not contain more than 50% of the topics so they were not discarded. Due to space limitations, these 3 topic sets are not described in the table.

Based on Tables 5.4 and 5.5, the topic sets have clearly captured some similarities between the merged topics: sometimes it conforms the original hierarchy like topic sets 3 and 4, and sometimes it captures other similarities like topic set 5 containing marketing, strategy and performance measurement together, or topic set 2 merging sports with related markets.

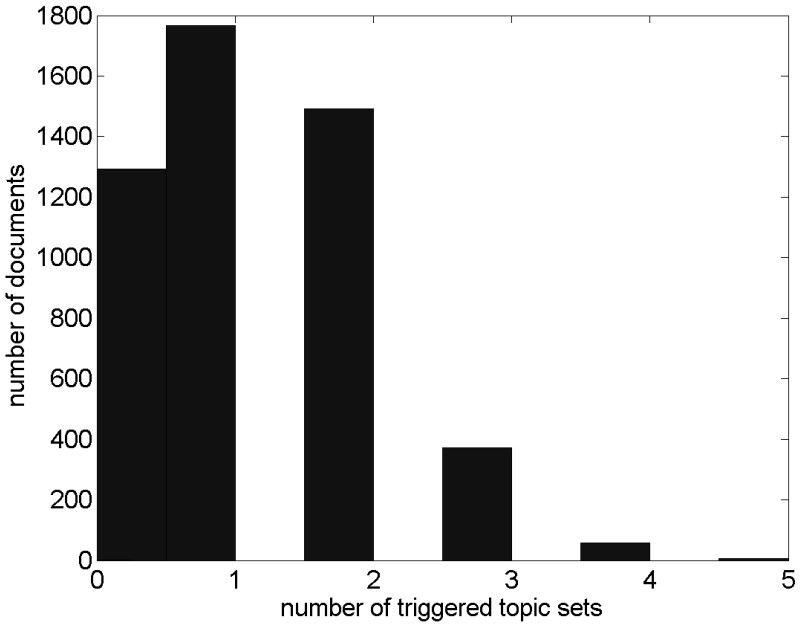


**Figure 5.7.** Histogram of the number of topic sets triggered by a document. Mean value is 1.23 topic sets.

**Table 5.4.** Topic sets in RCV1. The topics are represented by their code name in the RCV1 data set. The first letter identifies the four upper level topics *corporate/industrial*, *economics*, *government/social* and *markets*.

| ID | topics |
|---|---|
| 1 | M11, C15 |
| 2 | M14, GSPO |
| 3 | GPOL, GDIP |
| 4 | M14, M11 |
| 5 | M14, C15, M11, M13, M12, E12, C18, C11, C31 |
| 6 | GCRIM, C15, GPOL, GPRO |
| 7 | GCRIM, GPOL, C12 |
| 8 | M11 |
| 9 | GPOL, M14, GSPO |
| 10 | C15, GSPO, M14, GPOL, GDIS, GCRIM, M11, C21, GDIP, M13, E12, C11, GWEA, GVIO, E21, E11, C42 |
| 11 | M14, C15, M11, M13, GPOL, GCRIM, C18, C13, GDIP, C17, E21, C11, M12, GSPO, GVIO, C21, E12, C24, C12, E51, C42, C31, C41, GPRO, C33, GDEF, GDIS, E11, C22, G15, E13, E41, C14, GENV, C16, GHEA |
| 12 | C15, M14, M13, GPOL, C31, GCRIM, M11, C21, GDIP, E12, C13, GVIO, C11, M12, C18, E51, E11, E71 |

#### 5.4.1.3 Comparison of further topic sorting methods in FTSC

The F-measure based Topic Set Creation algorithm (Alg. 5.1 on page 76) orders the topics in descending individual F-measure order. The algorithm then searches for the topic set with the maximal F-measure in a greedy way taking the topics using this ordering after each other.

In the following, I will use the notations introduced in the proof of Prop. 5.2 on page 76. Beside the individual F-measure ($iF_T = iF(w,T)$), there are some further possibilities for the ordering of the topics: the number of covered documents of the topic ($c_T$), and the individual recall ($c_T/t_T$). Finally, although more time consuming, one can calculate the resulting F-measure every time after adding a topic as a function of the topic which will be added next. This leads to choosing the topic which mostly increases the F-measure, given the set of already added topics. (This is referred to as the $maxF$ method). The experiments for comparison

**Table 5.5.** Most important topic sets in RCV1: the automatic reconstruction of the topic hierarchy. Percentage under the topic identifier shows the ratio of documents triggering this topic set. Keywords without proper ending are a result of the stemming applied to the data set.

| ID | topics | example keywords |
|---|---|---|
| 1<br>23.6% | equity markets, performance | pretax, dax, pfennig, outperform, payout, canon, goldfield... |
| 2<br>13.6% | commodity markets, sports | cup, cricket, medal, coach, sheffield, yorkshir, wimbledon, athlet, mideast, lbs, intermonth, goalkeep, unbeat, semifin... |
| 4<br>13.6% | commodity markets, equity markets | unlead, gallon, composit, backward, meal, mideast, lbs, intermonth, overbought, telefon, sunseed, backfat, cottonseed... |
| 5<br>61.9% | commodity, equity, money and bond markets, performance, monetary/economic, ownership changes, strategy/plans, markets/marketing | volum, benchmark, stead, technic, buy, profit, actual, commod, mercantil, pork, factor, unlead, gallon, chip, unchang, liquid, yen, outweigh, pfennig, underperform, platin, bombay, payout, interbank, forint, overvalu, oversold, financier... |
| 6<br>29.4% | crime, law enforcement, performance, domestic politics, biographies... | widow, kidnap, jail, convict, extraordin, cocain, crim, murd, amnest, interpol, imprison, mafia, heroin, theft, horror, cardiac, bodyguard... |
| 10<br>81.4% | sports, commodity markets, disasters and accidents, crime, international relations... | rally, stead, earn, profit, pork, near, ghan, passport, favourit, captur, hero, storm, junior, wound, enemy, surrend, command, amsterdam, unfortun, vacuum, tea, checkpoint, cordon... |
| 11<br>99.3% | equity markets, money markets, domestic politics, crime, law enforcement, regulation/policy, legal/judicial, health... | decemb, detroit, microsoft, shift, personnel, speech, guidelin, concept, dive, simultan, consolid, geograph, omit, rotterdam, hydroelectr, portland, anchor, motorway, consul, denial, halfway... |
| 12<br>86.3% | markets/ marketing, international relations, war, economic performance... | fundament, volum, benchmark, export, buy, troop, versus, narrow, propagand, swissair, assassin, pistol, secretariat, oversold, hilton, hectic, rtl... |

**Table 5.6.** Comparison of results using different ordering methods in FTSC on the RCV1 data set.

| Measure | $c_T$ | $c_T/t_T$ | $iF_T$ | $maxF$ |
|---|---|---|---|---|
| precision | 0.6169 | **0.6307** | 0.6170 | 0.6430 |
| recall | **0.4658** | 0.4413 | **0.4674** | 0.4033 |
| F-measure | **0.5159** | 0.5029 | **0.5172** | 0.4780 |
| Mean number of triggered topic sets | 5.1181 | 7.9320 | **4.3766** | 5.0387 |
| Mean number of triggered topics | **38.3547** | 43.8467 | **39.7406** | 42.7464 |

were preformed on the RCV1 data set where the topics have different document numbers.

Table 5.6 presents a comparison of results using FTSC with different topic ordering methods. It should be noted that these are the final results of the two-level topic identification, so a theoretically optimal FTSC result does not necessarily lead to the best value. These results describe how far the FTSC algorithm supports the two-level topic identification method. The first three result columns correspond to methods which use values that can be calculated in advance. The $maxF$ method recalculates the F-measures for all remaining topics in every iteration, which means that method requires more calculation. Among the first three methods (the faster ones), $c_T$ and $iF_T$ seem to be slightly better than $c_T/t_T$, as they give slightly lower triggered topic number and higher F-measure, although with slightly lower precision. Although the $maxF$ method may seem to be better due to the recalculation of F-measures in every iteration, its results are not clearly better then the others. With the increased calculation requirements, it gives higher precision, but lower F-measure and a higher number of triggered topics, which leads to more comparisons altogether.

For these reasons, the $iF$ method was decided to be used in the final measurements, although the proof of its theoretical optimality required the condition of equal a-priori probability of the topics.

### 5.4.2 Experimental results on cascade structures

In this section, measurements evaluating the capabilities of the cascade structures are presented. The measurements were performed using the 20 Newsgroups data set.

Fig. 5.8 presents the similarity search results using a 5-level cascade. The results are presented after every cascade level in terms of precision, recall and F-measure. Narrow lines indicate results with threshold 1, and thick lines stand for results with threshold 2. The measurements used 20 base documents and exclusion threshold 1. Later cascade levels achieve lower precision as they use worse keywords, but the recall increases, as further levels select further documents. Increasing the threshold obviously increases the precision, but decreases the recall. After the fourth cascade level, there is a significant drop in precision as the system cannot choose acceptable keywords anymore. The reason for this is the lack of documents containing topic specific words, all such documents are already selected. This leads to very low precision, and a high recall of course. Based on these observations, the level of cascades is a suitable parameter for the user to set the desired balance between precision and recall.

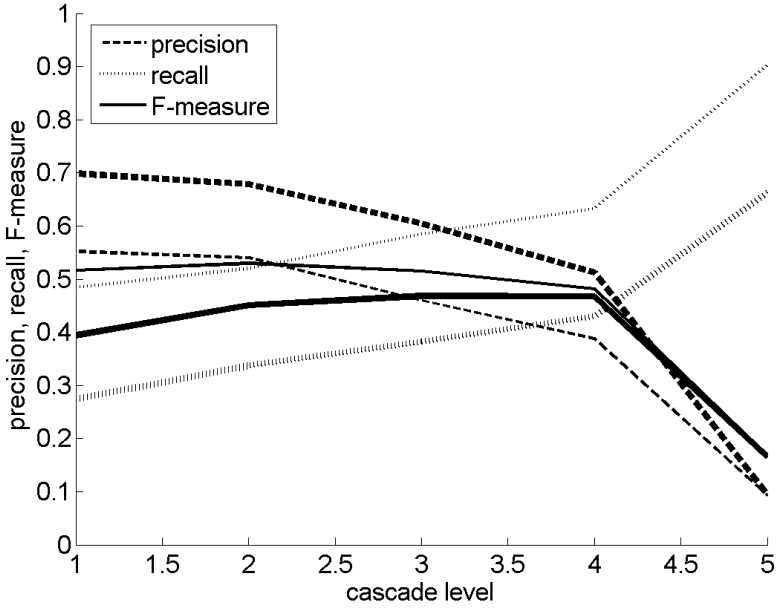


**Figure 5.8.** Performance of the cascaded similarity search on the 20 Newsgroups data set. The precision, recall and F-measure after the various cascade levels in presented. Narrow lines indicate results with threshold 1, thick lines stand for threshold 2. The number of base documents is 20.

Fig. 5.9 presents the results investigating the effects of the exclusion threshold. The threshold is 1 and the number of base documents is 20. Narrow lines stand

for exclusion threshold 1 and thick lines for exclusion threshold 2. Increasing the exclusion threshold is not recommended for the following reason: the exclusion selector is meant to remove as many as possible off-topics. This is the key of cascade structures as it makes the classification job of the next levels easier. But an increased exclusion threshold makes more off-topic documents remain in the document set, thus decreasing this effect.

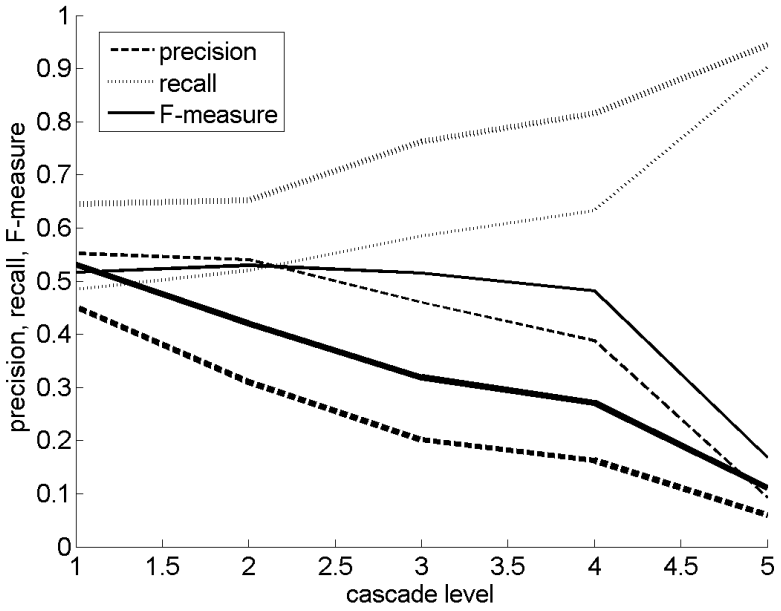


**Figure 5.9.** Effect of the exclusion threshold on cascade performance (20 Newsgroups). Narrow lines indicate exclusion threshold 1, thick lines stand for exclusion threshold 2. Number of base documents is 20.

Fig. 5.10 presents the number of keywords in the cascade structure. As mentioned before, after a certain point, the system is running out of documents containing topic specific words, so it cannot create good keyword lists either. This leads to keywords with low individual precision which cover many documents, so few keywords are enough to achieve acceptable recall. The detailed keyword lists for the topic *comp.graphics* are presented in Table 5.7.

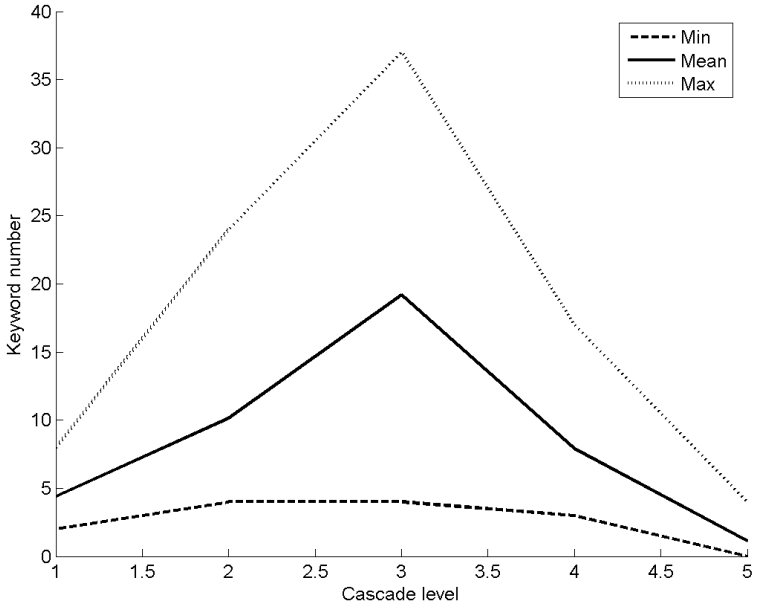


**Figure 5.10.** Keyword numbers in the 20 Newsgroups data set for 5 cascade levels. Minimal, maximal and mean keyword number of the topics are indicated.

**Table 5.7.** Keywords in the cascade levels for the topic *comp.graphics* in the 20 Newsgroups data set.

| Level | keywords |
|---|---|
| 1 | image images graphics |
| 2 | files points convert ftp format picture mode compatible algorithm colors |
| 3 | closed popular defined happy code useful takes latest lab site suggestions center directory six papers ps documentation greatly |
| 4 | inside best works note either anything luck already somebody ago file home department knows hello |
| 5 | mail please thanks |

Chapter 6

# Evaluation, Application and Conclusions

In this chapter, the results are evaluated briefly based on their applicability and novelty, possible application of the proposed methods is described, and the content of my dissertation is summarized.

Chapter 3 presented novel yields to feature selection for document topic identification and comparison. It is the basis for the further theses using it to create compact document representations for application in the search for similar documents and for the identification of a document's topic.

The most important new results in this chapter are the following:

- The main result is the keyword selection method called Precision based Keyword Selection (PKS).
- The suitability of PKS for creating 1-class classifiers for document topic identification was validated using both theoretical and experimental results.
- Further propositions on the linear execution time and keyword list optimality were discussed and proven formally.

Chapter 4 presented the key method of the search for similar documents (also called similarity search). It proposed the compact representation technique used for transferring the representation of a document from the remote mobile devices

for comparison. Using this representation, documents can be compared by transmitting only around 10-20 bytes of information, instead of the whole document. Synonym and hypernym words are handled using the proposed document extension technique adding further keywords to the documents.

The new results presented in this chapter are the following:

- The compact document representation was proposed.
- The similarity search technique was presented.
- The applicability of the similarity search was validated using both formal and experimental results.
- The document extension technique was proposed to handle synonym and hypernym keywords.
- For the learning of Related Generalizing Concepts (RGCs) of words, two methods were proposed: one based on co-occurrence statistics and one based on WordNet.
- The applicability of the document extension was validated using both formal and experimental results.

Chapter 5 proposed two classifier ensemble techniques aiming to improve the document topic identification and the similarity search. The first one is based on easy-to-identify topic sets and it creates a decision tree-like classifier ensemble leading to a lower number of checked keyword lists during the classification. The second technique trains successive 1-class classifiers specializing on the cases not recognized by the previous ones in order to improve the recall of the similarity search.

The new results are the following:

- The topic set creation algorithm was proposed, which consists of an initial topic set creation using the F-measure based Topic Set Creation (FTSC) algorithm, an evaluation phase, and finally the creating of additional topic sets for previously uncovered topics.
- The optimality of the greedy FTSC algorithm was proven formally.

- The applicability of the topic sets based topic identification was validated using experimental measurements.
- The cascaded similarity search structure was proposed and its applicability was validated using a formal proposition and experimental measurements.

## 6.1 Application of the results

The aim of this dissertation is a system for searching and comparing remote documents between mobile devices. The implementation requires the following components:

- Background process in mobile devices performing the search and the comparison of remote documents with the local ones.
- User interface on mobile devices notifying the user about detected similar documents and performing the download of the documents if requested by the user. The user interface can also be used to allow personalization of the similarity search: the system can allow the user to disable specific keywords of the local (base) documents, which will not be used in that case. It is also possible to enable adding further keywords to the base documents representation. In this case, care must be taken to avoid adding non-representable words (not keywords).
- Local document management for mobile devices creating the compact document representation of the documents and keeping the merged base document vector up-to-date. This component has to be only activated if the set of base documents changes upon moving a document to or from the mobile device.
- Central keyword list repository allowing access to the keyword lists every time.
- Central keyword list creation and topic maintenance system. Its goal is to create the keyword lists for all the topics and update these if there is a need to separate a topic into multiple ones. This maintenance has to be performed periodically to adapt the system to the changing social trends and to reflect these trends in the set of available topics.

Additionally, further optional services can be added to the system to decrease the functionality of the components running on mobile devices. These include the following:

- Optional topic identification server. The two-level topic identification can be performed with the help of a server too. To achieve this, the server has to send the keyword lists of the topic sets to the mobile device which chooses the triggered ones. After this, the server is asked for the keyword lists of the triggered topics. This extension becomes significant if the topic identification has many topics and many classification levels, where not having to store all keyword lists is significant advantage. (Of course, the keyword lists of already known topics or topics sets do not have to be retrieved again.)

- Optionally, the document extension can be performed on a central server as well which requires a globally accessible service performing the extension of arriving compact document representations.

The processing power requirement is only critical in the background searching process of mobile devices, as that is running in the low-resource environment continuously and it cannot deplete the batteries in hours due to data transmission and processing.

This dissertation contains the theoretical basics for the operation of the whole system. The tasks of the central keyword repository and maintenance system consists mainly of the training phases of the 1-class classifiers, the RGCF learning for document extension, and the topic set creation for the two-level topic identification. These functions can be implemented based on the MatLAB code created for the environment used for this research.

MatLAB code related to the application of the 1-class classifiers, document extension, two-level topic identification, and serial similarity search cascades can be used as a starting point of the implementation of the corresponding components. The components including the data transmissions, communication protocols, and user interfaces were not implemented, as they are not the subject of this dissertation.

The created MatLAB environment is also suitable to perform the evaluation measurements presented in this dissertation. The system is mainly implemented in MatLAB, but there are also components regarding document corpus pre-processing written in C#. Some result evaluation parts are implemented in terms of SQL queries and Excel sheets using MSSQL database. The database was used to allow

an easy exploration of the many possible parameter settings of the similarity search and the serial similarity search cascades.

## 6.2 Summary and future work

The new results proposed in this dissertation are organized in three theses and were proven with mathematical and engineering methods. The proposed techniques provide theoretical basis for searching for similar documents in a low-resource environment like a network of mobile devices. The summary of the theses are presented in Appendix A.

There are many directions for possible future work. This section summarizes the most important ones:

- Making the keyword selection distributed. The current form of the proposed system uses a central repository of keyword lists. Making the topic discovery distributed by using only the documents on the mobile devices as training set is the most important plan for future work. Achieving this requires not only a distributed keyword list creation method, but also a version control and maintenance system for synchronizing the keyword lists created in the decentralized cloud of mobile devices.
- Implementation of the system and testing in real-world environment. The implementation is planned to contain the decentralized keyword list creation mentioned previously. The real-world application is meant to provide further evaluation to all presented and planned techniques.
- Investigation of the classification capabilities if the $mp$ values of the keyword lists are transmitted as well, so that a maximum likelihood decision can be performed.
- Using $mp$ to measure the separability of topics and using its estimation to create classifier ensembles.
- Probabilistic estimation of the keyword lists sizes. The sizes of the keyword lists are now estimated using measurement results. In order to support further formal propositions, a probabilistic description of the F-measure optimizing process in PKS has to be developed.
- Checking the performance of serial cascaded similarity search and document extension together. The possible interferences between the cascaded similarity search and the document extension method still has to be evaluated in details.

- RGCF learning methods using ontologies. Beside WordNet, the Related Generalizing Concept Function can be learned from various sources of semantic information. Using ontologies is the most important possibility which remained unexplored.
- Keyword selection using semi-structured data. The PKS algorithm is purely a statistical method. Taking advantage of information in semi-structured corpora like web pages or XML documents is subject of further research.
- Monitoring topic changes. As the topics have to follow the social trends, these changes have to be detected and the corresponding changes have to be applied to the topics. For example if a music band gets very popular, it can get an own topic.
- The topic separability estimation capability of the *mp* value can be used to construct classifier ensembles by optimizing the first stages of decisions on easy-to-separate topic sets.

Appendix A

# Summary of the theses

## A.1 Summary of thesis I.

Publications related to this thesis are [Csorba and Vajk, 2006f]
[Csorba and Vajk, 2008c] [Csorba and Vajk, 2009b] [Csorba and Vajk, 2006e]
[Csorba and Vajk, 2006g] [Csorba, 2007b] [Csorba and Vajk, 2009e]
[Csorba, 2007a] [Csorba and Vajk, 2009f] [Csorba and Vajk, 2006a]
[Csorba and Vajk, 2006b] [Csorba and Vajk, 2006c] [Csorba and Vajk, 2006d]
[Csorba and Vajk, 2007b].

In my first thesis I propose a keyword selection algorithm called Precision based Keyword Selection (PKS, subthesis I.1.). It creates a keyword list for a given document topic. The selected keywords are topic specific ones which means that they rarely appear in off-topic documents. This makes them suitable for topic identification. Using the created keyword lists, I propose a simple, though sufficiently effective, classification method which selects the topic having the most keywords in the document (subthesis I.3.). I show that the PKS algorithm has linear execution time (subthesis I.2). I have shown that a parameter returned by the PKS algorithm allows the easy separability estimation of a given target- and a given set of off-topics (subthesis I.4.).

### Subthesis I.1.: Precision based Keyword Selection

*I have defined the individual precision and the minimal individual precision limit, the key concepts of the keyword selection algorithm. I have defined the Precision based Keyword Selection (PKS) algorithm which creates a keyword list for a given target topic in a parameterless way. I have shown with experimental results that the proposed algorithm allows higher precision in classification, than the baseline method. Formal results allow the estimation of the precision of document selections that use the created keyword lists. I have proven that a keyword list created by PKS allows the highest lower bound for expected precision (mp) among the possible keyword lists with the same size.*

In the following discussions, topics are handled as sets of documents and documents are handled as sets of words. Words selected for the document representations are the keywords.

**Definition A.1** (Document set selected by a keyword or keyword list)**.** The document set $S(w)$ selected by a keyword $w$ is the set of documents containing the word $w$:
$S(w) = \{d \in D : w \in d\}$ where $D$ is the set of all documents. Similarly, for a $K$ keyword list, $S(K) = \{d \in D : K \cap d \neq \emptyset\}$.

**Definition A.2** (Individual precision, recall and F-measure)**.** Given a target topic $T$, the precision, recall and F-measure of the set of selected documents can be calculated using the conventional definitions. Individual precision $iprec(w, T)$, recall $irecall(w, T)$ and F-measure $iF(w, T)$ of a word $w$ are the precision, recall and F-measure of $S(w)$ with respect to the target topic $T$.

**Proposed definition A.3** (Minimal individual precision limit $mp_T$ of topic $T$)**.** The minimal individual precision limit $mp_T$ of topic $T$ is the lower limit for individual precision of the keywords of topic $T$. Formally,

$$w \in K_T \leftrightarrow iprec(w, T) \geq mp_T \tag{A.1}$$

The lower bound of individual precisions in the keyword list, expressed by $mp_T$, is an important property of the keyword lists created by the PKS algorithm. PKS

optimizes $mp_T$ to maximize the F-measure of the selection using the resulting keyword list.

**Proposed definition A.4** (Precision based Keyword Selection (PKS))**.** The PKS algorithm is defined as presented in Algorithm 3.1 (on page 26). Given a $T$ target topic and a set of $\mathbb{U}$ off-topics, it returns a keyword list containing all words above the $mp_T$ minimal individual precision limit. $mp_T$ is optimized to achieve maximal F-measure with the keyword list.

The minimal individual precision limit $mp$ represents a balance between high precision and high recall, but high precision is maintained while optimizing F-measure. This gives high precision a priority over the high recall. Fig. 3.3 (on page 27) presents the curves of precision, recall and F-measure as a function of $x$.

Experimental results show that the $mp_T$ and $ep_T$ values, as returned by PKS, are suitable for estimations about the precision of the document selection.

### Subthesis I.2.: Linear execution time

*I have proven that the execution time of PKS is asymptotically linear with respect to the product of training document number and word number which is the size of the original document representations in the space of all possible words.*

### Subthesis I.3.: Most keywords (MKw) classification method

*I have shown that the most keywords (MKw) classification method – choosing the topic having the most keywords in the document – can take advantage of the properties of the keyword lists created by PKS, and achieve better classification quality than the baseline classifier.*

From the classifications point of view, a keyword list created by the PKS algorithm is a trained 1-class classifier: it is capable to select documents of its target topic. If there is a need to transfer a classifier selecting documents of a given topic, the transmission of the keyword list of the topic is sufficient.

The most important reason of choosing this classification method beside its simplicity is that the compact document representation - proposed in my second thesis - should contain as many keywords as possible. Using a classification method choosing the topic having the most keywords in the documents is a straightforward

decision, as only the keywords of the document's topic can be indicated in the compact document representation. (See thesis II. for further details.)

**Subthesis I.4.: Separability estimation**

*I have shown that the mp values are suitable for the measurement of topic separability, and the separability of a given topic from an arbitrary set of off-topics can be estimated using only the pairwise separabilities of the topics (Eqn. 3.4).*

The *mp* optimized by PKS is suitable to measure the separability of the target topic from a given set of off-topics: if there are few topic specific keywords (keywords with high individual precision), as two topics are very similar, *mp* has to be lower to achieve a keyword list having acceptable recall. Using the notation $mp_T(\{A;B\})$ for the minimal individual precision limit optimized by PKS for topic $T$ when the off-topics are $A$ and $B$, the proposed estimation method can estimate $mp_T(\{A;B\})$ using only $mp_T(\{A\})$ and $mp_T(\{B\})$.

Given a target topic $T$ and a set of off-topics $\mathbb{U}$, $mp_T(\mathbb{U})$ can be approximated with

$$\hat{mp}_T(\mathbb{U}) = \frac{1}{1+\sum_{V\in\mathbb{U}}(\frac{1}{mp_T(\{V\})}-1)} \tag{A.2}$$

## A.2 Summary of thesis II.

Publications related to this thesis are [Csorba and Vajk, 2008g]
[Csorba and Vajk, 2008i] [Csorba and Vajk, 2008b]
[Csorba and Vajk, 2008d] [Csorba and Vajk, 2008f] [Csorba, 2008]
[Csorba and Vajk, 2008h] [Csorba and Vajk, 2008j] [Csorba and Vajk, 2009a].

My second thesis proposes a searching method for mobile devices for finding remote documents with topics similar to the local ones. This is mainly a 1-class classification task. In order to maintain low communication traffic, documents are compared using only a compact document representation. The compact representations of the remote documents are downloaded, and used for the measurement of similarity between the remote and the locally stored documents (also called base documents).

I have proposed a compact document representation (subthesis II.1.). As the similarity measure is based on common keywords, documents not having common

keywords will be completely different. The problem of synonyms and other related words is handled by document extension (subthesis II.2.) which is based on a function returning the generalizations of keywords, called the Related Generalizing Concept Function (RGCF). The generalizing words are retrieved using an unsupervised, co-occurrance based method (subthesis II.3) or using WordNet (subthesis II.5.). If the unsupervised RGCF learning is used, a lower limit for the probability of document relatedness can be given, if the document similarity increases due to document extension (subthesis II.4.).

### Subthesis II.1.: Compact document representation, similarity search

*I have proposed a compact document representation technique allowing document topic comparison without the transmission of the whole document. I have shown the transformation of compact document representations into the original feature space and the similarity measure used to compare the remote documents to the base documents. Theoretical results provide a lower limit for the expected precision of the similarity search, and experimental results show that the representation is suitable for the proposed similarity search method.*

**Proposed definition A.5** (Compact document representation)**.** The compact document representation of a document $d$ is the pair $(\hat{T}(d), \mathbf{p}(\hat{T}(d), d))$ where $\hat{T}(d)$ is the estimated topic of the document $d$ and $\mathbf{p}(\hat{T}(d), d)$ is a binary vector indicating the presence or absence of the keywords of topic $\hat{T}(d)$ in the document $d$.

**Proposed definition A.6** (Similarity search)**.** The similarity search is the process of downloading the compact document representations of remote documents and calculating the number of their common keywords with the base documents. If this number exceeds a user-defined threshold, the user is notified.

### Subthesis II.2.: Document extension and Related Generalizing Concept Function

*I have proposed the document extension method, based on the Related Generalizing Concept Function (RGCF). RGCF provides the generalizing keywords of a given keyword. I proposed two methods for learning the RGCF: one based on unsupervised, co-occurrance statistics based method, and one based on the hypernym graph*

*of WordNet. Experimental results show that the document extension successfully increases the similarity measure of related documents which share few or no keywords.*

**Proposed definition A.7** (Related General Concepts Function (RGCF))**.** *RGCF* is the function returning the set of related general concepts (also keywords) $v_i$ for the keyword $w$:
$RGCF(w) = \{v_1, v_2, ..., v_n\}$.

**Proposed definition A.8** (Document extension)**.** The document extension adds all the generalizations of the keywords of a document, to the document:

$$d^{ext} = d \cup \bigcup_{w \in d} RGCF(w) \tag{A.3}$$

where $d^{ext}$ is the extended document.

### Subthesis II.3.: Keyword co-occurrence (KCo) based RGCF learning

*I proposed an unsupervised, keyword co-occurrance based method for RGCF learning. I have presented experimental results showing that the proposed method successfully discovers generalizations of the keywords. This makes it suitable for application for document extension.*

If a hierarchy of the document topics is available, keywords for topics on all hierarchy levels can be created. For the sake of simplicity, I will consider a two-level hierarchy with upper and lower level topics, but the methods can be easily generalized to more levels. The method is based on the assumption that keywords of upper level topics are more general than the keywords of lower levels, and that frequently co-occurring keywords are related to each other. Based on these observations, the proposed method collects generalizing keywords as follows:

**Proposed definition A.9** (Keyword co-occurrence based RGCF learning)**.**

$$v \in RGCF(w) \leftrightarrow w \in K_G, v \in K_H : G \subseteq H, \frac{S(w) \cap S(v)}{S(w)} \geq mcr$$

where $K_G$ and $K_H$ are the set of keywords for topics $G$ and $H$ respectively, $G \subseteq H$ indicates that $G$ is a subtopic of $H$, $S(w)$ is the set of documents containing the word $w$, and $mcr$ is the minimal co-occurrence rate.

The first condition ensures the generalization and the second ensures the frequent co-occurrence of the keywords $v$ and $w$.

### Subthesis II.4.: Probability of relatedness of documents with increasing similarity measure due to document extension

*I have given a lower limit for the probability, that if the similarity of two documents is increased by the document extension, the two documents belong to related topics. Experimental results show that document extension increases the similarity of related documents, and does not increase the similarity of unrelated documents significantly.*

**Definition A.10** (Related topics)**.** Two topics are related in a topic hierarchy, if they have a common parent topic.

Documents of related topics (for example subtopics of *animals* like *hawks* and *dolphins*) are considered to be a suitable test environment for the document extension, as the common parent topic ensures loose relatedness but the documents are different enough to share few or no keywords.

### Subthesis II.5.: Creating RGCF using WordNet

*I have proposed an RGCF learning method using the hypernym graph of WordNet. I have presented measurement results comparing this RGCF learning to the keyword co-occurrence based one. Measurement results show that this RGCF learning also allows document extension to improve the recall of the similarity search.*

**Proposed definition A.11** (Hypernym distance of words in WordNet)**.** The $h(w, v)$ hypernym distance of words $w$ and $v$ is the length of the route along the directed hypernym edges from $w$ to $v$. If $w$ and $v$ are synonyms (they belong to the same synset in WordNet), $h(w, v) = 0$.

For example if *animal* is a hypernym of *mammal*, and *mammal* is a hypernym of *elephant*, then $h(elephant, animal) = 2$.

**Proposed definition A.12** (WordNet based RGCF learning)**.** The RGCF learned using WordNet is defined as

$$v \in RGCF(w) \leftrightarrow h(w, v) \leq dl \tag{A.4}$$

where $dl$ is the distance limit, a parameter of the learning method.

## A.3 Summary of thesis III.

Publications related to this thesis are [Csorba and Vajk, 2008e]
[Csorba and Vajk, 2009c] [Csorba and Vajk, 2006h] [Csorba, 2006]
[Csorba and Vajk, 2006i] [Csorba and Vajk, 2007a] [Csorba and Vajk, 2008a].

In my third thesis I propose two classifier ensemble techniques aiming to decrease the number of used keyword lists during topic identification, and to improve the recall of the similarity search.

The first part of the theses is a two-level topic identification method. Its key idea is to create a decision tree-like classifier by merging some similar topics into topic sets. If a topic set is very different from a document, the topics inside this topic set are not checked for similarity with the document. This way, the mean number of checked topics per document can be decreased. The topic sets are created using the F-measure based Topic Set Creation (FTSC) algorithm (subthesis III.1.), and its optimality in terms of the easy-to-identify property of the created topic sets is also discussed (subthesis III.2.). The Small Sets on Demand extension (subthesis III.3.) is proposed for further decreasing the mean number of checked topics.

In the second part of the thesis, I propose a cascaded similarity search method to improve the recall of the search for similar documents. Its key idea is the training of further levels of 1-class classifiers specialized on the documents not recognized by previous levels. I show that there are words which can be better keywords in the later levels, as in the first one (subthesis III.4.).

## Two-level topic identification using topic sets

Fig. 5.1 (on page 74) illustrates the topic identification using topic sets: only the topics of triggered topic sets, ones having common keyword with the document, will be checked.

### Subthesis III.1.: F-measure based Topic Set Creation algorithm (FTSC)

*I have proposed the greedy F-measure based Topic Set Creation (FTSC) algorithm for creating topic sets which can be easily identified, and the method for creating the topic sets used in the document topic identification using FTSC. Experimental results show that the proposed method successfully decreases the mean number of checked keyword lists during the identification of a document's topic.*

The key idea of the F-measure based Topic Set Creation algorithm (FTSC) is to retrieve for every word the set of topics which that word can select with the highest F-measure. With other words, if one would select all documents the given word appears in, which target topic set would get the highest F-measure.

**Proposed definition A.13** (Easy-to-identify topic set of a word)**.** The easy-to-identify topic set of the word $w$ is

$$\mathbb{TS}^{opt}(w) = \arg\max_{\mathbb{TS}}\{iF(w, \mathbb{TS})\} \tag{A.5}$$

where $iF(w, \mathbb{TS})$ is the individual F-measure of the word $w$ with respect to the topic set $\mathbb{TS}$ as target topic.

Finding the optimal topic set for every word by calculating the individual F-measure for all possible topic sets would be very resource consuming. The FTSC algorithm, illustrated in Alg. 5.1 (on page 76), is searching for the topic set $\mathbb{TS}^{opt}(w)$ for a given $w$ word in a greedy way: it adds the $T$ topics to the topic set in descending $iF(w, T)$ order until the individual F-measure is maximized.

Using FTSC, the topic sets are created as illustrated in Alg. 5.2 (on page 79). Topic set evaluation in the function *GetSuitableTopicSets* is creating keyword lists for all topic sets (using PKS) and removes the ones covering too many topics or achieving too low precision or recall. The detailed evaluation conditions are presented in the dissertation. In the last step, topics not covered by any remaining

topic sets are moved into a separate topic set created for each of them individually. These additional topic sets contain only one topic, and are called *small topic sets*.

**Proposed definition A.14** (small and big topic sets)**.** A topic set is a small topic set, if it contains exactly one topic. Otherwise, it is a big topic set.

After the topic sets have been created, a keyword list is created for each of them using PKS. The second level of the ensemble is trained just as there would be no topic sets. A keyword list is created for every topic set and topic independently.

### Subthesis III.2.: Optimality of FTSC

*I have proven that the greedy FTSC algorithm achieves optimal results, if the a-priori probabilities of the topics are equal.*

### Subthesis III.3.: Using the classifier ensemble, Small Sets on Demand extension

*I have proposed the small sets on demand (SSD) extension for the topic sets based topic identification which avoids the check of small topic sets in many cases. I have shown measurements supporting that this extension successfully decreases the mean number of checked keyword lists during the identification of a document's topic, and it does not significantly decrease the precision.*

Using the topic sets for topic identification is illustrated in Alg. 5.3 (on page 80). If the topic of a new document has to be identified, it is first compared to the keyword lists of the topic sets, and then, the keyword lists of the triggered topics are checked. The triggered topic having the most keywords in the document is selected.

**Proposed definition A.15** (Set of triggered topic sets)**.** Given a $d$ document, the $Trig(d, \mathbb{TS})$ set of triggered topic sets contains the topic sets having common keywords with the document.

$$Trig(d) = \{\mathbb{TS} : K_{\mathbb{TS}} \cap d \neq \emptyset\} \tag{A.6}$$

The Small Sets on Demand (SSD) extension means that if sufficient (a given *mb* number of) big topics are triggered, then small topic sets are not checked. In this case, the algorithm is illustrated in Alg. 5.4 (on page 81).

## Cascade structures

Cascade structures are designed for the similarity search to increase the recall by training further selectors (creating further keyword lists) specialized on the documents missed by the previous levels.

### Subthesis III.4.: The similar document search cascade structure

*I have proposed a cascade structure for the search for similar documents. The key idea of the cascades is the training of multiple levels of 1-class classifiers. Each level is trained only on the documents not selected by previous levels, not as similar document, neither as off-topic document. Experimental results show that this structure successfully increases the recall of the similarity search.*

The elements of a cascade structure consists of a keyword list used to recognize similar documents, and an exclude keyword list used to recognize off-topic documents. Only documents not found to be either similar or off-topic, are forwarded to the next cascade level which is specialized on these remaining cases.

**Proposed definition A.16** (Allowed and Excluded topic set)**.** $\mathbb{A}$ is the set of topics which have base documents: $T \in \mathbb{A} \leftrightarrow \exists d \in B \cap T$ where $B$ is the set of base documents. The set of excluded topics contains topics which do not have base documents: $\mathbb{E} = \overline{\mathbb{A}}$.

The training of the cascades is presented in Alg. 5.5 (on page 82), and its application in Alg. 5.6 (on page 82).

*I have shown that there are words which can have individual precision higher in later cascade levels than in the previous levels. This supports the theory that successive selectors can successfully cover more documents without too strong precision decrease, as there are words which get better keywords after removing several off-topic and target topic documents.*

# Bibliography

[Amitay et al., 2005] Amitay, E., Darlow, A., Konopnicki, D., and Weiss, U. (2005). Queries as anchors: selection by association. In *HYPERTEXT '05: Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, pages 193–201, New York, NY, USA. ACM.

[Ando, 2001] Ando, R. (2001). *The Document Representation Problem: An Analysis of LSI and Iterative Residual Rescaling*. PhD thesis, Cornell University.

[Androutsopoulos et al., 2000] Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., and Spyropoulos, C. D. (2000). An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–167, New York, NY, USA. ACM.

[Anh and Moffat, 2006] Anh, V. N. and Moffat, A. (2006). Pruned query evaluation using pre-computed impacts. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 372–379, New York, NY, USA. ACM.

[Aseervatham, 2008] Aseervatham, S. (2008). A local latent semantic analysis-based kernel for document similarities. In *International Joint Conference on Neural Networks, IJCNN 2008*, pages 214–219. IEEE.

[Aslam et al., 2005] Aslam, J. A., Yilmaz, E., and Pavlu, V. (2005). A geometric interpretation of r-precision and its correlation with average precision. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 573–574, New York, NY, USA. ACM.

[Azcarraga et al., 2004] Azcarraga, A., Yap TN, J., Tan, J., and Chua, T. (2004). Evaluating keyword selection methods for websom text archives. *IEEE Transactions on Knowledge and Data Engineering*, 16(3):380–383.

[Baeza-Yates, 1999] Baeza-Yates, R.; Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley.

[Bai et al., 2005] Bai, J., Song, D., Bruza, P., Nie, J.-Y., and Cao, G. (2005). Query expansion using term relationships in language models for information retrieval. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 688–695, New York, NY, USA. ACM.

[Banerjee, 2005] Banerjee, A. (2005). *Scaleable Clustering Algorithms*. PhD thesis, The University of Texas at Austin.

[Banerjee and Langford, 2004] Banerjee, A. and Langford, J. (2004). An objective evaluation criterion for clustering. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 515–520, New York, NY, USA. ACM Press.

[Batu et al., 2003] Batu, T., Ergün, F., Kilian, J., Magen, A., Raskhodnikova, S., Rubinfeld, R., and Sami, R. (2003). A sublinear algorithm for weakly approximating edit distance. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 316–324, New York, NY, USA. ACM.

[Baumann et al., 2002] Baumann, S., Dengel, A., Junker, M., and Kieninger, T. (2002). Combining ontologies and document retrieval techniques: a case study for an e-learning scenario. *13th International Workshop on Database and Expert Systems Applications, 2002.*, pages 133–137.

[Belew, 2000] Belew, R. (2000). *Finding Out About. A Cognitive Perspective on Search Engine Technology and the WWW.* Cambridge University Press.

[Benczúr et al., 2009] Benczúr, A. A., Erdélyi, M., Masanes, J., and Siklósi, D. (2009). Web spam challenge proposal for filtering in archives. In *5th International Workshop on Adversarial Information Retrieval on the Web, AIRWeb'09*, pages 61–62.

[Benczur et al., 2006] Benczur, A., Hernáth, Z., and Porkoláb, Z. (2006). Lord - lay-out relationship and domain definition language. In *Advances of Database and Information Systems*, pages 251–230.

[Bhowmik, 2008] Bhowmik, R. (2008). Keyword extraction from abstracts and titles. *Southeastcon, 2008. IEEE*, pages 610–617.

[Billerbeck et al., 2003] Billerbeck, B., Scholer, F., Williams, H. E., and Zobel, J. (2003). Query expansion using associated queries. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 2–9, New York, NY, USA. ACM.

[Blair, 2002] Blair, D. C. (2002). The challenge of commercial document retrieval, part II: a strategy for document searching based on identifiable document partitions. *Information Processing and Management*, 38(2):293–304.

[Bloehdorn et al., 2006] Bloehdorn, S., Cimiano, P., and Hotho, A. (2006). Learning ontologies to improve text clustering and classification. In Spiliopoulou, M., Kruse, R., Nürnberger, A., Borgelt, C., and Gaul, W., editors, *From Data and Information Analysis to Knowledge Engineering: Proceedings of the 29th Annual Conference of the German Classification Society (GfKl 2005), Magdeburg, Germany, March 9-11, 2005*, volume 30 of *Studies in Classification, Data Analysis, and Knowledge Organization*, pages 334–341. Springer.

[Bloehdorn and Hotho, 2004] Bloehdorn, S. and Hotho, A. (2004). Boosting for text classification with semantic features. In Mobasher, B., Nasraoui, O., Liu, B., and Masand, B. M., editors, *Advances in Web Mining and Web Usage Analysis*, volume 3932 of *Lecture Notes in Computer Science*, pages 149–166. Springer.

[Bodon, 2009] Bodon, F. (2009). Adatbányászati algoritmusok. http://www.cs.bme.hu/~bodon/.

[Brand and Huang, 2003] Brand, M. and Huang, K. (2003). A unifying theorem for spectral embedding and clustering. In *International Workshop On Artificial Intelligence and Statistics*.

[Bratko et al., 2006] Bratko, A., Filipič, B., Cormack, G. V., Lynam, T. R., and Zupan, B. (2006). Spam filtering using statistical data compression models. *J. Mach. Learn. Res.*, 7:2673–2698.

[Brill, 1992] Brill, E. (1992). A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, Trento. URL: citeseer.nj.nec.com/brill92simple.html.

[Büttcher and Clarke, 2006] Büttcher, S. and Clarke, C. L. A. (2006). A document-centric approach to static index pruning in text retrieval systems. In Yu, P. S., Tsotras, V. J., Fox, E. A., and Liu, B., editors, *Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management, Arlington, Virginia, USA, November 6-11, 2006*, pages 182–189. ACM.

[Carmel et al., 2001] Carmel, D., Cohen, D., Fagin, R., Farchi, E., Herscovici, M., Maarek, S., Y., and Soffer, A. (2001). Static index pruning for information retrieval systems. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–50.

[Chen et al., 2005] Chen, C.-M., Lee, H.-M., and Hwang, C.-W. (2005). A hierarchical neural network document classifier with linguistic feature selection. *Applied Intelligence*, 23(3):277–294.

[Chih-Ping Wei and Hsiao, 2008] Chih-Ping Wei, C.-S. Y. and Hsiao, H.-W. (2008). A collaborative filtering-based approach to personalized document clustering. *Decision Support Systems, Special Issue Clusters*, 45(3):413–428.

[Chim and Deng, 2008] Chim, H. and Deng, X. (2008). Efficient phrase-based document similarity for clustering. *IEEE Trans. Knowl. Data Eng*, 20(9):1217–1229.

[Chirita et al., 2006] Chirita, P.-A., Firan, C. S., and Nejdl, W. (2006). Summarizing local context to personalize global web search. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 287–296, New York, NY, USA. ACM.

[Choi et al., 2006] Choi, N., Song, I.-Y., and Han, H. (2006). A survey on ontology mapping. *ACM SIGMOD Record*, 35(3):34–41.

[Chowdhury and McCabe, 1998] Chowdhury, A. and McCabe, M. C. (1998). Improving information retrieval systems using part of speech tagging. Technical report.

[Combarro et al., 2006] Combarro, E. F., Montañés, E., Ranilla, J., and Díaz, I. (2006). Angular measures for feature selection in text categorization. In Haddad, H., editor, *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC), Dijon, France, April 23-27, 2006*, pages 826–830. ACM.

[Cook and Das, 2007] Cook, D. J. and Das, S. K. (2007). How smart are our environments? an updated look at the state of the art. *Pervasive and Mobile Computing*, 3(2):53–73.

[Cooper et al., 2002] Cooper, J. W., Coden, A., and Brown, E. W. (2002). A novel method for detecting similar documents. In *35th Annual Hawaii International Conference on System Sciences (HICSS'02)*, volume 4, page 101.

[Cormode and Muthukrishnan, 2007] Cormode, G. and Muthukrishnan, S. (2007). The string edit distance matching problem with moves. *ACM Trans. Algorithms*, 3(1):2.

[Cristianini et al., 2002] Cristianini, N., Shawe-Taylor, J., and Lodhi, H. (2002). Latent semantic kernels. *Journal of Intelligent Information Systems*, 18(2/3):127–152. Special Issue on Automated Text Categorization.

[Csorba, 2006] Csorba, K. (2006). Term cluster creation and confidence measurement for document clustering. In *AACS'06 Automation and Applied Computer Science Workshop 2006*, pages 73–84.

[Csorba, 2007a] Csorba, K. (2007a). Dokumentumok meta-adatainak kinyerése mobil peer-to-peer hálózatok fejlesztésére. Technical report, Department of Automation and Applied Informatics, Budapest University of Technology and Economics.

[Csorba, 2007b] Csorba, K. (2007b). Precision optimized term clustering. In *AACS'07 Automation and Applied Computer Science Workshop 2007*, pages 13–22.

[Csorba, 2008] Csorba, K. (2008.). Searching for similar documents with mobile devices using taxonomies. In *AACS'08 Automation and Applied Computer Science Workshop 2008*, pages 139–149.

[Csorba and Vajk, 2006a] Csorba, K. and Vajk, I. (2006a). Composition of methods in document clustering. In *ISIM'06 Information Systems Implementation and Modelling*, pages 149–156.

[Csorba and Vajk, 2006b] Csorba, K. and Vajk, I. (2006b). Document clustering using singular value decomposition and double clustering. In *microCAD 2006 International Scientific Conference*, pages 49–54.

[Csorba and Vajk, 2006c] Csorba, K. and Vajk, I. (2006c). Double clustering in latent semantic indexing. In *SAMI'06 The 4th Slovakian - Hungarian Joint Symposium on applied Machine Intelligence*, pages 319–329.

[Csorba and Vajk, 2006d] Csorba, K. and Vajk, I. (2006d). Feature space transformations in document clustering. In *INES'06, 10th International Conference on Intelligent Engineering Systems*, pages 175–179.

[Csorba and Vajk, 2006e] Csorba, K. and Vajk, I. (2006e). Greedy term selection for document classification with given minimal precision. In *HUCI'06 7th International Symposium of Hungarian Researchers on Computational Intelligence*, pages 247–254.

[Csorba and Vajk, 2006f] Csorba, K. and Vajk, I. (2006f). Supervised term cluster creation for document clustering. *Scientific Bulletin of Politehnica University of Timisoara, Romania, Transactions on Automatic Control and Computer Science*, 3:p.49.

[Csorba and Vajk, 2006g] Csorba, K. and Vajk, I. (2006g). Term clustering and confidence measurement in document clustering. In *CONTI'06 The 7th International Conference on Technical Informatics*, volume 2., pages 183–188.

[Csorba and Vajk, 2006h] Csorba, K. and Vajk, I. (2006h). Term clustering and confidence measurement in document clustering. In *Advances in Information Systems Development, New Methods and Practice for the Networked Society*, volume 1., pages 481–491. Springer Verlag.

[Csorba and Vajk, 2006i] Csorba, K. and Vajk, I. (2006i). Term clustering and confidence measurement in document clustering. In *ICCC 2006. IEEE International Conference on Computational Cybernetics*, pages 1–6.

[Csorba and Vajk, 2007a] Csorba, K. and Vajk, I. (2007a). Cascaded classifiers in document clustering. In *microCAD 2007 International Scientific Conference*, volume N, pages 19–23.

[Csorba and Vajk, 2007b] Csorba, K. and Vajk, I. (2007b). Comparison of feature space covering methods in document clustering. In *microCAD 2007 International Scientific Conference*, volume N, pages 13–18.

[Csorba and Vajk, 2008a] Csorba, K. and Vajk, I. (2008a). Cascaded search for similar documents between mobile devices. In *The 12th WSEAS International Conference on COMPUTERS*, pages 122–127.

[Csorba and Vajk, 2008b] Csorba, K. and Vajk, I. (2008b). Improving document similarity measurement for mobile environment with document extension. In *ECML PKDD 2008, Ubiquitous Knowledge Discovery Workshop.*

[Csorba and Vajk, 2008c] Csorba, K. and Vajk, I. (2008.c). Iterative search for similar documents on mobile devices. *Lecture Notes in Artificial Intelligence, LNCS*, 5243.:38–45.

[Csorba and Vajk, 2008d] Csorba, K. and Vajk, I. (2008.d). Searching for similar documents in a mobile device environment. In *microCAD 2008 International Scientific Conference*, volume O, pages 151–156.

[Csorba and Vajk, 2008e] Csorba, K. and Vajk, I. (2008e). Searching for similar documents on mobile devices using classifier cascades. *International Journal of Computers*, 2.:pp.126–133.

[Csorba and Vajk, 2008f] Csorba, K. and Vajk, I. (2008f). Searching for similar documents on mobile devices using taxonomy. In *CONTI'08 The 7th International Conference on Technical Informatics*, volume 2., pages 103–108.

[Csorba and Vajk, 2008g] Csorba, K. and Vajk, I. (2008g). Searching for similar documents using keywords and taxonomies in mobile device environments. *Scientific Bulletin of Politehnica University of Timisoara, Romania, Transactions on Automatic Control and Computer Science*, 53.:p.233.

[Csorba and Vajk, 2008h] Csorba, K. and Vajk, I. (2008.h). Taxonomic support for document classication in mobile device environment. In *HSI'08 Human System Interaction*.

[Csorba and Vajk, 2008i] Csorba, K. and Vajk, I. (2008i). Topic comparison of remote documents using small communication traffic. *Periodica Polytechnica*. Under review.

[Csorba and Vajk, 2008j] Csorba, K. and Vajk, I. (2008.j). Unsupervised taxonomy creation for mobile device applications. In *microCAD 2008 International Scientific Conference*, volume O, pages 7–11.

[Csorba and Vajk, 2009a] Csorba, K. and Vajk, I. (2009a). Comparison of two generalizing keyword selection algorithms. In *MicroCAD 2009.*, pages pp.13–18.

[Csorba and Vajk, 2009b] Csorba, K. and Vajk, I. (2009b). Estimating keyword based separability of document topics. *Informatica*. Under review.

[Csorba and Vajk, 2009c] Csorba, K. and Vajk, I. (2009c). Improved topic identification for similar document search on mobile devices. *Acta Cybernetica*, 19., pp.17–40.

[Csorba and Vajk, 2009d] Csorba, K. and Vajk, I. (2009d). Keyword selection for very compact document topic representations. *Information Processing Letters*. Under review.

[Csorba and Vajk, 2009e] Csorba, K. and Vajk, I. (2009e). Measuring topic separability using topic specific keywords. In *MicroCAD 2009.*, pages pp.7–12.

[Csorba and Vajk, 2009f] Csorba, K. and Vajk, I. (2009f). Transformations and selection methods in document clustering. In *Intelligent Engineering Systems and Computational Cybernetics*. Springer Verlag.

[Deerwester et al., 1990] Deerwester, S., Dumais, S.-T., Furnas, G.-W., Landauer, T.-K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal American Society Information Science*, 41(6):391–407.

[Delmastro et al., 2008] Delmastro, F., Passarella, A., and Conti, M. (2008). P2P multicast for pervasive ad hoc networks. *Pervasive and Mobile Computing*, 4(1):62–91.

[Dobrokhotov et al., 2003] Dobrokhotov, P. B., Goutte, C., Veuthey, A.-L., and Gaussier, É. (2003). Combining NLP and probabilistic categorisation for document and term selection for swiss-prot medical annotation. In *Eleventh International Conference on Intelligent Systems for Molecular Biology (Supplement of Bioinformatics)*, pages 91–94.

[Dong and Han, 2005] Dong, Y.-S. and Han, K.-S. (2005). Boosting svm classifiers by ensemble. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1072–1073, New York, NY, USA. ACM.

[Du et al., 2007] Du, N., Wu, B., and Wang, B. (2007). Concept forest: A new ontology-assisted text document similarity measurement method. *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 395–401.

[Dubin, 2004] Dubin, D. (2004). The most influential paper gerard salton never wrote. *Library Trends*, 3:748–764.

[Dudás, 2006] Dudás, L. (2006). Morfémák megtanulása szövegből. In *MicroCAD 2006 International Scientific Conference*, pages 61–66.

[Dupret, 2003] Dupret, G. (2003). Latent concepts and the number orthogonal factors in latent semantic analysis. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 221–226, New York, NY, USA. ACM Press.

[Edmonds, 2007] Edmonds, A. (2007). Using concept structures for efficient document comparison and location. *CIDM 2007. IEEE Symposium on Computational Intelligence and Data Mining*, pages 238–242.

[Efron et al., 2002] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2002). Least angle regression.

[Efron, 2008] Efron, M. (2008). Query expansion and dimensionality reduction: Notions of optimality in rocchio relevance feedback and latent semantic indexing. *Information Processing & Management*, 44(1):163–180.

[Ekler et al., 2008] Ekler, P., Nurminen, J. K., and Kiss, A. J. (2008). Experiences of implementing bittorrent on java me platform. In *1st IEEE International Peer-to-Peer for Handheld Devices Workshop, CCNC'08*.

[El-Yaniv and Souroujon, 2001] El-Yaniv, R. and Souroujon, O. (2001). Iterative double clustering for unsupervised and semi-supervised learning. In *Machine Learning: ECML 2001, 12th European Conference on Machine Learning, Freiburg, Germany, September 5-7, 2001, Proceedings*, volume 2167 of *Lecture Notes in Artificial Intelligence*, pages 121–132. Springer.

[Ercan and Cicekli, 2007] Ercan, G. and Cicekli, I. (2007). Using lexical chains for keyword extraction. *Inf. Process. Manage*, 43(6):1705–1714.

[Farial Shahnaz and Plemmons, 2006] Farial Shahnaz, Michael W. Berry, V. P. and Plemmons, R. J. (2006). Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386.

[Ferber, 2003] Ferber, R. (2003). *Information Retrieval. Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web.* dpunkt Verlag, Heidelberg.

[Fleischman and Hovy, 2003] Fleischman, M. and Hovy, E. H. (2003). Recommendations without user preferences: a natural language processing approach. In *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces*, pages 242–244. ACM.

[Flesca et al., 2007] Flesca, S., Manco, G., Masciari, E., Pontieri, L., and Pugliese, A. (2007). Exploiting structural similarity for effective web information extraction. *Data Knowl. Eng*, 60(1):222–234.

[Forstner and Charaf., 2005] Forstner, B. and Charaf., H. (2005). Neighbor selection in peer-to-peer networks using semantic relations. *WSEAS Transactions on Information Science and Applications*, 2(2):239–244.

[Fortuna et al., 2006a] Fortuna, B., Grobelnik, M., and Mladenić, D. (2006a). Semi-automatic data-driven ontology construction system. In *Conference on Data Mining and Data Warehouses (SiKDD 2006).*

[Fortuna et al., 2006b] Fortuna, B., Grobelnik, M., and Mladenić, D. (2006b). System for semi-automatic ontology construction. In *3rd Annual European Semantic Web Conference.*

[Freund and Schapire, 1995] Freund, Y. and Schapire, R. E. (1995). A decision theoretic generalization of on-line learning and an application to boosting. In *Second European Conference on Computational Learning Theory (EuroCOLT-95)*, pages 23–37.

[Fuhr, 2004] Fuhr, N. (2004). *Information Retrieval Methods for Literary Texts.* PhD thesis, Universitaet Duisburg-Essen, Campus Duisburg.

[Furnas et al., 1988] Furnas, G., Deerwester, S., Dumais, S. T., Landauer, T. K., Harshman, R., Streeter, L. A., and Lochbaum, K. E. (1988). Information retrieval using a singular value decomposition model of latent semantic structure. In Chiaramella, Y., editor, *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 465–480, Grenoble, France. ACM.

[Garner and Hemsworth, 1997] Garner, P. N. and Hemsworth, A. (1997). A keyword selection strategy for dialogue move recognition and multi-class topic identification. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, pages 1823–1826, Munich, Germany.

[Ghanem et al., 2002] Ghanem, M. M., Guo, Y., Lodhi, H., and Zhang, Y. (2002). Automatic scientific text classification using local patterns: Kdd cup 2002 (task 1). *SIGKDD Explor. Newsl.*, 4(2):95–96.

[Göker and Myrhaug, 2008] Göker, A. and Myrhaug, H. I. (2008). Evaluation of a mobile information system in context. *Inf. Process. Manage*, 44(1):39–65.

[Goutte and Gaussier, 2005] Goutte, C. and Gaussier, E. (2005). A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. *Lecture Notes in Computer Science*, 3408:345–359.

[Grenager et al., 2005] Grenager, T., Klein, D., and Manning, C. (2005). Unsupervised learning of field segmentation models for information extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 371–378, Ann Arbor, Michigan. Association for Computational Linguistics.

[Guo, 2008] Guo, Q. (2008). The similarity computing of documents based on VSM. In Takizawa, M., Barolli, L., and Enokido, T., editors, *Network-Based Information Systems, 2nd International Conference, NBiS 2008*, volume 5186 of *Lecture Notes in Computer Science*, pages 142–148, Turin, Italy. Springer.

[H. Gregory Silber, 2002] H. Gregory Silber, K. F. M. (2002). Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, 28(4):487–496.

[H. Lieberman and Vivacqua, 1999] H. Lieberman, N. v. D. and Vivacqua, A. (1999). Let's browse: a collaborative browsing agent. *Knowledge-Based Systems*, 12(8):427–431.

[Halácsy et al., 2004] Halácsy, P., Kornai, A., Németh, L., Rung, A., Szakadát, I., and Trón, V. (2004). Creating open language resources for hungarian. In *4th Int. Conf. on Language Resources and Evaluation (LREC-04)*, pages 203–210.

[Hammouda and Kamel, 2002] Hammouda, K. M. and Kamel, M. S. (2002). Phrase-based document similarity based on an index graph model. In *ICDM*, pages 203–210. IEEE Computer Society.

[Hammouda and Kamel, 2004] Hammouda, K. M. and Kamel, M. S. (2004). Document similarity using a phrase indexing graph model. *Knowl. Inf. Syst*, 6(6):710–727.

[Hamon et al., 1998] Hamon, T., Nazarenko, A., and Gros, C. (1998). A step towards the detection of semantic variants of terms in technical documents. In Boitet, C. and Whitelock, P., editors, *Proceedings of the Thirty-Sixth Annual*

*Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, pages 498–504, San Francisco, California. Association for Computational Linguistics, Morgan Kaufmann Publishers.

[He et al., 2008] He, D., Brusilovsky, P., wook Ahn, J., Grady, J., Farzan, R., Peng, Y., Yang, Y., and Rogati, M. (2008). An evaluation of adaptive filtering in the context of realistic task-based information exploration. *Inf. Process. Manage*, 44(2):511–533.

[Hempstalk et al., 2008] Hempstalk, K., Frank, E., and Witten, I. H. (2008). One-class classification by combining density and class probability estimation. In *Machine Learning and Knowledge Discovery in Databases*, volume 5211 of *Lecture Notes in Computer Science*, pages 505–519. Springer Verlag.

[Hersh, 1994] Hersh, W. (1994). OHSUMED: An interactive retrieval evaluation and new large test collection for research. In Croft, W. B. and van Rijsbergen, C. J., editors, *Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval*, pages 192–201, London, UK. Springer Verlag.

[Hofmann, 2001] Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196.

[Hsi-Ching Lin and Chen, 2006] Hsi-Ching Lin, L.-H. W. and Chen, S.-M. (2006). Query expansion for document retrieval based on fuzzy rules and user relevance feedback techniques. *Expert Systems with Applications*, 31(2):397–405.

[Hurson et al., 2006] Hurson, A. R., Muñoz-Avila, A. M., Orchowski, N., Shirazi, B., and Jiao, Y. (2006). Power-aware data retrieval protocols for indexed broadcast parallel channels. *Pervasive and Mobile Computing*, 2(1):85–107.

[Isbell and Viola, 1999] Isbell, C.-L. and Viola, P. (1999). Restructuring sparse high dimensional data for effective retrieval. *Adv. in Neural Information Proc. Systems*, 11:480–486.

[Iván and Ormándi, 2007] Iván, S. and Ormándi, R. (2007). Magyar mondatok svm alapú szintaxis elemzése. In *V. Magyar Számítógépes Nyelvészeti Konferencia*.

[Jain et al., 1999] Jain, A., Murty, M., and Flynn, P. (1999). Data clustering: a survey. *ACM Computing Surveys*, 31(3):264–323.

[Jamali et al., 2006] Jamali, M., Sayyadi, H., Hariri, B. B., and Abolhassani, H. (2006). A method for focused crawling using combination of link structure and content similarity. In *Web Intelligence*, pages 753–756. IEEE Computer Society.

[Jensen et al., 2008] Jensen, D., Giraud-Carrier, C. G., and Davis, N. (2008). A method for computing lexical semantic distance using linear functionals. *J. Web Sem*, 6(2):99–108.

[Jia and Peng, 2007] Jia, X. and Peng, H. (2007). Probabilistic document correlation model. *CISW 2007. International Conference on Computational Intelligence and Security Workshops*, pages 433–436.

[J.K. Nurminen, 2008] J.K. Nurminen, J. N. (2008). Energy-consumption in mobile peer-to-peer - quantitative results from file sharing. In *CCNC 2008: Consumer Communications and Networking Conference*, pages 729–733, Las Vegas, NV.

[Jones et al., 2006] Jones, R., Rey, B., Madani, O., and Greiner, W. (2006). Generating query substitutions. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 387–396, New York, NY, USA. ACM.

[Kazem Taghva and Condit, 2004] Kazem Taghva, Julie Borsack, T. N. and Condit, A. (2004). The role of manually-assigned keywords in query expansion. *Information Processing & Management*, 40(3):441–458.

[Keerthi, 2005] Keerthi, S. S. (2005). Generalized LARS as an effective feature selection tool for text classification with SVMs. In *Proceedings of the Twenty-Second International Conference on Machine Learning*, Bonn, Germany.

[Klimt and Yang, 2004] Klimt, B. and Yang, Y. (2004). The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning 2004*, pages 217–226.

[Kokiopoulou and Saad, 2004] Kokiopoulou, E. and Saad, Y. (2004). Polynomial filtering in latent semantic indexing for information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 104–111, New York, NY, USA. ACM.

[Kotsiantis, 2007] Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. In Maglogiannis, I., Karpouzis, K., Wallace, M., and Soldatos, J., editors, *Emerging Artificial Intelligence Applications in Computer Engineering*, volume 160 of *Frontiers in Artificial Intelligence and Applications*, pages 3–24. IOS Press.

[Kozima and Ito, 1996] Kozima, H. and Ito, A. (1996). Context-sensitive measurement of word distance by adaptive scaling of a semantic space. Technical report, Communications Research Laboratory, Japan.

[Kozlova, 2005] Kozlova, N. (2005). Automatic ontology extraction for document classification. Master's thesis, Computer Science Department, Saarland University.

[Kwok, 1996] Kwok, K. L. (1996). A new method of weighting query terms for ad-hoc retrieval. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 187–195, New York, NY, USA. ACM.

[Lacroix et al., 1998] Lacroix, Z., Sahuguet, A., and Chandrasekar, R. (1998). Information extraction and database techniques: A user-oriented approach to querying the Web. *Lecture Notes in Computer Science*, 1413:289.

[Lagus and Kaski, 1999] Lagus, K. and Kaski, S. (1999). Keyword selection method for characterizing text document maps. In *ICANN99. Ninth International Conference on Artificial Neural Networks (IEE Conf. Publ. No.470)*, volume 1, pages 371–6, London, UK. IEE.

[Lagus et al., 2004] Lagus, K., Kaski, S., and Kohonen, T. (2004). Mining massive document collections by the websom method. *Information Sciences*, 163(1-3):135–156.

[Lam-Adesina and Jones, 2001] Lam-Adesina, A. M. and Jones, G. J. F. (2001). Applying summarization techniques for term selection in relevance feedback. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–9, New York, NY, USA. ACM.

[Lang, 1995] Lang, K. (1995). Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339.

[Larsen, 2005] Larsen, K. (2005). Generalized naive bayes classifiers. *SIGKDD Explorations*, 7(1):76–81.

[Lee and Seung, 1999] Lee, D.-D. and Seung, H.-S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791.

[Lee and Ng, 2007] Lee, J. W. and Ng, Y.-K. (2007). Using fuzzy-word correlation factors to compute document similarity based on phrase matching. *Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on*, 2:186–192.

[Lehmann and Shawe-Taylor, 2006] Lehmann, A. and Shawe-Taylor, J. (2006). A probabilistic model for text kernels. In *ICML '06: Proceedings of the 23rd*

*international conference on Machine learning*, pages 537–544, New York, NY, USA. ACM Press.

[Lewis et al., 2005] Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2005). Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(1):361–398.

[Li and Chou, 2002] Li, L. and Chou, W. (2002). Improving latent semantic indexing based classifier with information gain.

[Liaw and Huang, 2003] Liaw, S.-S. and Huang, H.-M. (2003). An investigation of user attitudes toward search engines as an information retrieval tool. *Computers in Human Behavior*, 19(6):751–765.

[Liu et al., 2003] Liu, B., Chin, C. W., and Ng, H. T. (2003). Mining topic-specific concepts and definitions on the web. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 251–260.

[Liu et al., 2004] Liu, F., Yu, C. T., and Meng, W. (2004). Personalized web search for improving retrieval effectiveness. *IEEE Trans. Knowl. Data Eng*, 16(1):28–40.

[Lodhi et al., 2002] Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002). Text classification using string kernels. *J. Mach. Learn. Res.*, 2:419–444.

[Losada and Barreiro, 2006] Losada, D. E. and Barreiro, A. (2006). Negations and document length in logical retrieval. *Information Systems*, 31(7):610–620.

[Luo et al., 2007] Luo, P., Xiong, H., Lü, K., and Shi, Z. (2007). Distributed classification in peer-to-peer networks. In Berkhin, P., Caruana, R., and Wu, X., editors, *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*, pages 968–976. ACM.

[Maedche and Staab, 2001] Maedche, A. and Staab, S. (2001). Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2):72–79.

[Magyar et al., 2007] Magyar, G., Knapp, G., Wojtkowski, W., and Wojtkowski, W. G., editors (2007). *Advances in Information Systems Development. New Methods and Practice for the Networked Society.* Springer Science, New York, USA. ISBN 978-0-387-70760-0.

[Marchionini, 2006] Marchionini, G. (2006). Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46.

[Miller et al., 1990] Miller, G. A., Fellbaum, C., Gross, D., , and Miller, K. J. (1990). Introduction to wordnet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

[Mladenić and Grobelnik, 2003] Mladenić, D. and Grobelnik, M. (2003). Feature selection on hierarchy of web documents. *Decision Support Systems*, 35(1):45–87.

[Mocan et al., 2006] Mocan, A., Cimpian, E., and Kerrigan, M. (2006). Formal model for ontology mapping creation. In Cruz, I. F., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., and Aroyo, L., editors, *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 459–472, Athens. Springer.

[Moffat et al., 2005] Moffat, A., Zobel, J., and Hawking, D. (2005). Recommended reading for ir research students. *SIGIR Forum*, 39(2):3–14.

[Moreale and Watt, 2003] Moreale, E. and Watt, S. (2003). An agent-based approach to mailing list knowledge management. In van Elst, L., Dignum, V., and Abecker, A., editors, *AMKM*, volume 2926 of *Lecture Notes in Computer Science*, pages 118–129. Springer.

[Mészáros and Dobrowiecki, 2009] Mészáros, T. and Dobrowiecki, T. (2009). Controlled natural languages for interface agents. In *AAMAS 2009: 8th international conference on Autonomous agents and multiagent systems*, pages 1173–1174.

[M.T. Martín-Valdivia and Ureña-López, 2008] M.T. Martín-Valdivia, M.C. Díaz-Galiano, A. M.-R. and Ureña-López, L. (2008). Using information gain to improve multi-modal information retrieval systems. *Information Processing & Management*, 44(3):1146–1158.

[Na et al., 2007] Na, S.-H., Kang, I.-S., and Lee, J.-H. (2007). Adaptive document clustering based on query-based similarity. *Inf. Process. Manage*, 43(4):887–901.

[Niall Rooney and Dobrynin, 2006] Niall Rooney, David Patterson, M. G. and Dobrynin, V. (2006). A scaleable document clustering approach for large document corpora. *Information Processing & Management*, 42(5):1163–1175.

[Novácek et al., 2007] Novácek, V., Dabrowski, M., Kruk, S. R., and Handschuh, S. (2007). Extending community ontology using automatically generated suggestions. In Wilson, D. and Sutcliffe, G., editors, *The Twentieth International Florida Artificial Intelligence Research Society Conference, May 7-9, 2007, Key West, Florida, USA*, page 290. AAAI Press.

[Oded Maimon, 2005] Oded Maimon, L. R. (2005). *The Data Mining and Knowledge Discovery Handbook*. Springer Verlag.

[Otterbacher et al., 2008] Otterbacher, J., Radev, D. R., and Kareem, O. (2008). Hierarchical summarization for delivering information to mobile devices. *Inf. Process. Manage*, 44(2):931–947.

[Peshkin and Pfeffer, 2003] Peshkin, L. and Pfeffer, A. (2003). Bayesian information extraction network. In *Intl. Joint Conference on Artificial Intelligence, 2003.*

[Pohárnok et al., 2007] Pohárnok, M., Naszódi, M., Kis, B., Nagy, L., Bóna, A., and László, J. (2007). Exploring the spatial organization of interpersonal relations by means of computational linguistic analysis. *Empirical Culture and Text Research*, 3:39–49.

[Pons-Porrata et al., 2007] Pons-Porrata, A., Llavori, R. B., and Ruiz-Shulcloper, J. (2007). Topic discovery based on text mining techniques. *Inf. Process. Manage*, 43(3):752–768.

[Porter, 2006] Porter, M. (2006). The porter stemming algorithm, official website: http://tartarus.org/ martin/porterstemmer/.

[Pradhan et al., 2004] Pradhan, S., Ward, W., Hacioglu, K., Martin, J. H., and Jurafsky, D. (2004). Shallow semantic parsing using support vector machines. In *Human Language Technologies annual meeting 2004.*, pages 233–240.

[Prószéky and Miháltz, 2008] Prószéky, G. and Miháltz, M. (2008). Magyar wordnet: az első magyar lexikális szemantikai adatbázis. *Magyar Terminológia*, 1(1):43–57.

[Qi and Davison, 2009] Qi, X. and Davison, B. D. (2009). Web page classification: Features and algorithms. *ACM Comput. Surv.*, 41(2):1–31.

[Qiu and Frei, 1993] Qiu, Y. and Frei, H.-P. (1993). Concept based query expansion. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–169, New York, NY, USA. ACM.

[Roberto J. Bayardo, 2007] Roberto J. Bayardo, Yiming Ma, R. S. (2007). Scaling up all pairs similarity search. In *16th Int. Conf. on World Wide Web*, pages 131–140, Banff, Alberta, Canada.

[Sabou et al., 2006] Sabou, M., Lopez, V., Motta, E., and Uren, V. (2006). Ontology selection: Ontology evaluation on the real semantic web. In *WWW2006*, Edinburgh, UK.

[Sahami and Heilman, 2006] Sahami, M. and Heilman, T. D. (2006). A web-based kernel function for measuring the similarity of short text snippets. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 377–386, New York, NY, USA. ACM.

[Sahlgren and Karlgren, 2002] Sahlgren, M. and Karlgren, J. (2002). SICS at CLEF 2002: Automatic query expansion using random indexing. In *Proceedings of CLEF 2002 Workshop*.

[Sakkis et al., 2003] Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D., and Stamatopoulos, P. (2003). A memory-based approach to anti-spam filtering for mailing lists. *Information Retrieval*, 6(1):49–73.

[Salton, 1987] Salton, G. (1987). *A Theory of Indexing (CBMS-NSF Regional Conference Series in Applied Mathematics No. 18)*. Philadelphia, Society for Industrial and Applied Mathematics.

[Salton et al., 1975] Salton, G., Wong, A., and Yao, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11).

[Schönhofen, 2008] Schönhofen, P. (2008). *Extracting document features to improve classification and clustering*. PhD thesis, Budapest University of Technology and Economics.

[Schönhofen and Charaf, 2004] Schönhofen, P. and Charaf, H. (2004). Using concept relationships to improve document categorization. *Periodica Polytechnica Ser. El. Eng.*, 48(3-4):165–182.

[Schone and Jurafsky, 2000] Schone, P. and Jurafsky, D. (2000). Knowledge-free induction of morphology using latent semantic analysis. In Cardie, C., Daelemans, W., Nédellec, C., and Sang, E. T. K., editors, *Proceedings of the Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic Workshop, Lisbon, 2000*, pages 67–72. Association for Computational Linguistics, Somerset, New Jersey.

[Sevillano et al., 2006] Sevillano, X., Cobo, G., Alías, F., and Socoró, J. C. (2006). Feature diversity in cluster ensembles for robust document clustering. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 697–698, New York, NY, USA. ACM Press.

[Silva and Martins, 2003] Silva, M. J. and Martins, B. (2003). Web information retrieval with result set clustering. In Moura-Pires, F. and Abreu, S., editors, *Progress in Artificial Intelligence, 11th Protuguese Conference on Artificial Intelligence, EPIA 2003, Beja, Portugal, December 4-7, 2003, Proceedings*, volume 2902 of *Lecture Notes in Computer Science*, pages 450–454. Springer.

[Singhal, 2001] Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Engineering Bulletin*, 24(4):35–43.

[Sinka and Corne, 2003] Sinka, M. P. and Corne, D. (2003). Evolving better stoplists for document clustering and web intelligence. In Abraham, A., Köppen, M., and Franke, K., editors, *Design and Application of Hybrid Intelligent Systems, HIS03, the Third International Conference on Hybrid Intelligent Systems, Melbourne, Australia, December 14-17, 2003*, volume 105 of *Frontiers in Artificial Intelligence and Applications*, pages 1015–1023. IOS Press.

[Slonim and Tishby, 2000] Slonim, N. and Tishby, N. (2000). Document clustering using word clusters via the information bottleneck method. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Clustering, pages 208–215.

[Smith and Humphreys, 2006] Smith, A. E. and Humphreys, M. S. (2006). Evaluation of unsupervised semantic mapping of natural language with leximancer concept mapping. *Behavior Research Methods*, 38(2):262–279.

[Snow et al., 2004] Snow, R., Jurafsky, D., and Ng, A. Y. (2004). Learning syntactic patterns for automatic hypernym discovery. In *Annual Conference on Neural Information Processing Systems 2004.*

[Strehl, 2002] Strehl, A. (2002). *Relationship-based Clustering and Cluster Ensembles for High-Dimensional Data Mining.* PhD thesis, The University of Texas at Austin.

[Szarvas et al., 2006] Szarvas, G., Farkas, R., Felföldi, L., Kocsor, A., and Csirik, J. (2006). A highly accurate named entity corpus for hungarian. In *5th Int. Conf. on Language Resources and Evaluation (LREC-06).*

[Tang et al., 2006] Tang, J., Li, J.-Z., Liang, B., Huang, X., Li, Y., and Wang, K. (2006). Using bayesian decision for ontology mapping. *J. Web Sem*, 4(4):243–262.

[Tao et al., 2005] Tao, T., Wang, X., Mei, Q., and Zhai, C. (2005). Accurate language model estimation with document expansion. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 273–274, New York, NY, USA. ACM Press.

[Tao Li and Ogihara, 2007] Tao Li, S. Z. and Ogihara, M. (2007). Hierarchical document classification using automatically generated hierarchy. *Journal of Intelligent Information Systems*, 29(2).

[Taskar et al., 2004] Taskar, B., Klein, D., Collins, M., Koller, D., and Manning, C. (2004). Max-margin parsing. In *Empirical Methods in Natural Language Processing (EMNLP04)*, Barcelona, Spain.

[Termier et al., 2001] Termier, A., Sebag, M., and Rousset, M.-C. (2001). Combining statistics and semantics for word and document clustering. In Maedche, A., Staab, S., Nedellec, C., and Hovy, E. H., editors, *IJCAI'2001 Workshop on Ontology Learning*, volume 38 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Thompson, 2008] Thompson, P. (2008). Looking back: On relevance, probabilistic indexing and information retrieval. *Information Processing & Management*, 44(2):963–970.

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selecion via the lasso. In *Journal of the Royal Statistical Society*, Methodological, pages 267–288.

[Tikk, 2007] Tikk, D., editor (2007). *Szövegbányászat*. TypoTEX, Budapest. 294 pages.

[Tikk et al., 2005] Tikk, D., Kardkovács, Z., and Magyar, G. (2005). A szavak hálójában: szabadszavas mélyháló-kereső program. *Híradástechnika*, 60(5):2–8. ISSN 0018–2028, (In Hungarian).

[Toutanova et al., 2004] Toutanova, K., Manning, C. D., and Ng, A. Y. (2004). Learning random walk models for inducing word dependency distributions. In *International Conference on Machine Learning 2004*.

[Tseng and Juang, 2003] Tseng, Y.-H. and Juang, D.-W. (2003). Document-self expansion for text categorization. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 399–400, New York, NY, USA. ACM Press.

[Tun, 2006] Tun, N. N. (2006). Semantic enrichment in ontologies for matching. In *AOW '06: Proceedings of the second Australasian workshop on Advances in ontologies*, pages 91–100, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

[Turenne, 2003] Turenne, N. (2003). Learning semantic classes for improving email classification. In *Eighteenth International Joint Conference on Artificial Intelligence, Text-Mining & Link-Analysis Workshop*.

[Vailaya et al., 2005] Vailaya, A., Bluvas, P., Kincaid, R., Kuchinsky, A., Creech, M. L., and Adler, A. (2005). An architecture for biological information extraction and representation. *Bioinformatics*, 21(4):430–438.

[Vechtomova et al., 2003] Vechtomova, O., Robertson, S., and Jones, S. (2003). Query expansion with long-span collocates. *Inf. Retr.*, 6(2):251–273.

[Wagstaff et al., 2001] Wagstaff, K., Cardie, C., Rogers, S., and Schroedl, S. (2001). Constrained K-means clustering with background knowledge. In *Proc. 18th International Conf. on Machine Learning*, pages 577–584. Morgan Kaufmann, San Francisco, CA.

[Wan and McKeown, 2004] Wan, S. and McKeown, K. (2004). Generating overview summaries of ongoing email thread discussions. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 549, Morristown, NJ, USA. Association for Computational Linguistics.

[Wan, 2007] Wan, X. (2007). A novel document similarity measure based on earth mover's distance. *Inf. Sci*, 177(18):3718–3730.

[Wang and Yang, 2006] Wang, J. and Yang, S. (2006). Content-based clustered p2p search model depending on set distance. In *WI-IATW '06: Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, pages 471–476, Washington, DC, USA. IEEE Computer Society.

[Wang et al., 2006] Wang, X., Sun, J.-T., Chen, Z., and Zhai, C. (2006). Latent semantic analysis for multiple-type interrelated data objects. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 236–243, New York, NY, USA. ACM Press.

[Wedemeyer and Srinivasan, 2003] Wedemeyer, M. and Srinivasan, P. (2003). Mining concept profiles with the vector model or where on earth are diseases being studied? In *Proceedings of Text Mining Workshop. Third SIAM International Conference on Data Mining*.

[Weiss et al., 2005] Weiss, S. M., Indurkhya, N., Zhang, T., and Damerau, F. J. (2005). *Text Mining - Predictive Methods for Analyzing Unstructured Information*. Springer Verlag.

[Xiao, 2005] Xiao, J. (2005). Agent-based similarity-aware web document prefetching. In *International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, pages 928–933. IEEE Computer Society.

[Yan et al., 2005] Yan, J., Liu, N., Zhang, B., Yan, S., Chen, Z., Cheng, Q., Fan, W., and Ma, W.-Y. (2005). Ocfs: optimal orthogonal centroid feature selection for text categorization. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 122–129, New York, NY, USA. ACM Press.

[Yu, 2004] Yu, K. (2004). *Statistical Learning Approaches to Information Filtering*. PhD thesis, Ludwig-Maximilians-Universit¨at München, Fakultät für Mathematik, Informatik und Statistik.

[Zhang et al., 2004] Zhang, L., Zhu, J., and Yao, T. (2004). An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(4):243–269.

[Zhu and Mutka, 2008] Zhu, D. and Mutka, M. W. (2008). Cooperation among peers in an ad hoc network to support an energy efficient IM service. *Pervasive and Mobile Computing*, 4(3):335–359.

[Zú 2001] Zú niga, G. L. (2001). Ontology: its transformation from philosophy to information systems. In *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, pages 187–197, New York, NY, USA. ACM.

[Zvi Boger and Shapira, 2001] Zvi Boger, Tsvi Kuflik, P. S. and Shapira, B. (2001). Automatic keyword identification by artificial neural networks compared to manual identification by users of filtering systems. *Information Processing & Management*, 37(2):187–198.