Acta Cybernetica **19** (2009) 17–40.

Improved Topic Identification for Similar Document Search on Mobile Devices*

Kristóf Csorba† and István Vajk†

Abstract

This paper presents a novel, two level classifier ensemble designed to support document topic identification in mobile device environments. The proposed system aims at supporting mobile device users who search for documents located in other mobile devices which have similar topic to the documents on the users own device. Conforming to the environment of mobile devices, the algorithms are designed for slower processor, smaller memory capacity and they maintain small data traffic between the devices in order to keep low the cost of communication. We propose a keyword list based topic comparison, enhanced with a two level classifier ensemble to accelerate the topic identification process. The new technique enables document topic comparison using few communication traffic and it requires few calculations.

Keywords: document classification, topic hierarchization, keyword selection

1 Introduction

Due to the rapid improvement of mobile devices (like mobile phones and PDAs) in storage capacity and processing capabilities, more and more information can be stored on them. People reading e-books on PDA are not unusual. There is an increasing interest in searching techniques designed for such ubiquitous environments, and most of the search engines are still based on keywords specified by the user. The techniques presented in this paper are designed for an automatic searching for documents which might be of the user's interest. The proposed system is analyzing the local documents and notifies the user if there is a document with similar topics available for retrieval. It is running as a background process and requires user interaction only if it finds something. Our work is part of a project aiming at supporting semantic search in mobile devices connected to a peer-to-peer network [1].

*This work has been fund of the Hungarian Academy of Sciences for control research and the Hungarian National Research Fund (grant number T68370).

†Budapest University of Technology and Economics, Department of Automation and Applied Informatics, Goldmann Gy. tér 3., 1111 Budapest, HUNGARY.
E-mail: {kristof,vajk}@aut.bme.hu

DOI: 10.14232/actacyb.19.1.2009.3

Searching for documents with a given topic is a frequent task today. The most common approaches are based on keywords given by the user as search criteria. Another possibility is the "topic by example" [2] approach where the user presents documents for which similar ones should be retrieved. For this purpose document topics have to be compared. This is the case in our system as well, extended with the assumption that documents stored on the users own mobile device represent the interest of the user [3].

The most common solutions for topic comparison use the bag-of-words representation of documents and calculate the similarity measure of them using the document feature vectors. These feature vectors may indicate relevance of some words to the document's topic or some extracted features. Due to the huge number of the possible words, the feature space methods usually reduce the number of features before comparing the documents. The two main approaches that use this method are feature selection and feature extraction [4][5]. Feature selection means the selection of some already available features and discarding the remaining ones, feature extraction on the other hand aims at deriving new features based on the original ones. Some feature selection methods are based on information gain or mutual information [6], least angle regression [7] or optimal orthogonal centroid feature selection [8].

If there are many document topics, using a classifier ensemble [6] may avoid the need to compare a document to every possible topic during the classification. If the number of the possible topics of a document can be limited, a classifier can be created with significantly better classification capabilities. The paper [9] proposes a technique which creates a topic hierarchy using linear discriminant projection and trains multiple classifiers, like nodes of a decision tree, to improve the classification.

The system proposed in this paper allows document classification and topic comparison of documents in mobile device environments. The main contribution consists of a feature selection algorithm and a topic hierarchy creation method. The most important property the proposed system requires is the applicability in mobile devices with limited processor and memory capacity. The created classifiers and topic comparison method have to be easy-to-calculate, and the topic comparison of documents stored on different devices has to be performed using limited communication traffic because communication between mobile devices is usually not free of charge. To conform these requirements the size and processing complexity of the document topic representations have to be in balance between very small size and good comparability. This requirement makes the frequently used huge, weighted document vectors not applicable.

The organization of the paper is as follows: section 2 presents an overview of the contribution which consists of a keyword selection and document search method presented in section 3, a two level topic identification technique presented in section 4 and experimental results presented in section 5. Finally, conclusions are summarized in section 6.

2 Overview of the contribution

Our proposed document topic identification and comparison system is based on the following key ideas:

A list of topic-specific keywords is assigned to every possible topic using a labeled training set and a supervised learning method. A document is represented by the identifier of the topic which has the most common keywords with the document, and a simple binary vector indicating the presence or absence of the given keywords in the document. This representation allows comparing the topics more detailed than just comparing the assigned topic labels of the documents.

This representation is easy to generate: the document has to be parsed and every word has to be compared to the keywords in the keyword lists. The topic with the most common keywords is selected and the binary indicator vector is created.

To create the classifier first the topic specific keyword lists have to be created: a set of the most topic-specific keywords for all possible document topics is created. Weighting of the keywords is not possible due to the binary vector in the representation which is essential for size limitations. Another important constraint implied by the proposed similarity measure (number of common keywords) is the following: keywords which would have to get negative weight cannot be used at all, because the similarity measure, the number of common keywords cannot decrease due to the presence of an additional keyword. This makes most feature selection approaches like mutual-information and information-gain based ones not applicable.

After the keyword lists have been created, the mobile devices have to identify the topic best matching their documents. They could download every keyword list and compare all of them to the documents but this would reduce scalability of the system if there were many topics. To waive this limitation, a classifier ensemble is created which is similar to a decision tree: topics are ordered into topic sets which have their keyword lists as well. This enables the mobile device to omit the check of some keyword lists and limit the search space for the best matching topic to 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. The extension using topic sets is presented in section 4.

The search for similar documents is a very simple procedure: the mobile device downloads the compact topic representation of remote documents and calculates

the similarity, the number of common keywords, using only the representations. If it exceeds a user defined threshold, the user is notified. The user can decide if the document itself should be downloaded or not. The similarity search is presented in subsection 3.2 in details.

3 Document topic representation and comparison using keyword lists

In this section, the keyword selection method, the document representation, and the process of searching similar documents are described. As the document search is executed in the background and the user is notified when a document similar to the local ones is found, the rate of misclassifications is of key importance in this application. It is much more important than finding all similar documents.

For performance measurements we will use the common measures precision, recall and F-measure: if there is a specific target topic from which we want to select as many documents as possible, c is the number of correctly selected documents, f the number of false selections, and t is the number of documents in the target topic then precision is defined as $P = c/(c + f)$, recall is $R = c/t$, and F-measure is $F = 2PR/(P + R)$. Our aim to avoid misclassifications means the requirement of a high precision even at the price of a lower recall. But we cannot entirely omit the recall either, so we optimize F-measure while maintaining a high precision. Intuitively described a classifier with high precision is selected which is low enough to allow an acceptable recall as well.

In the description of the algorithms the following notations are used:

- $d \in T$ indicates that the document d belongs to the topic T.
- $w \in d$ indicates that the word w is present in the document d. Documents are handled as sets of words.
- $K_T = \{w_1, w_2, ..., w_n\}$ is the keyword list (set of keywords w_i) for the topic T. T might refer to a set of topics as well, in which case K_T contains all keywords of the topics in the topic set.
- A *keyword* is a word which appears in at least one keyword list. This means that keywords are words used in the topic representations.
- S_T is the selector which aims at selecting documents from the topic T. We use the selector expression instead of classifier because it selects documents for one topic, and leaves the remaining ones to be selected by other selectors. Documents not selected by any selectors will have unidentified topic. S_T is interpreted as the set of documents selected by the selector as well.
- $d \in S_T$ means that the document d is selected by the selector S_T.
- S_d is a selector based on a document d which selects documents which have common keywords with the document d.

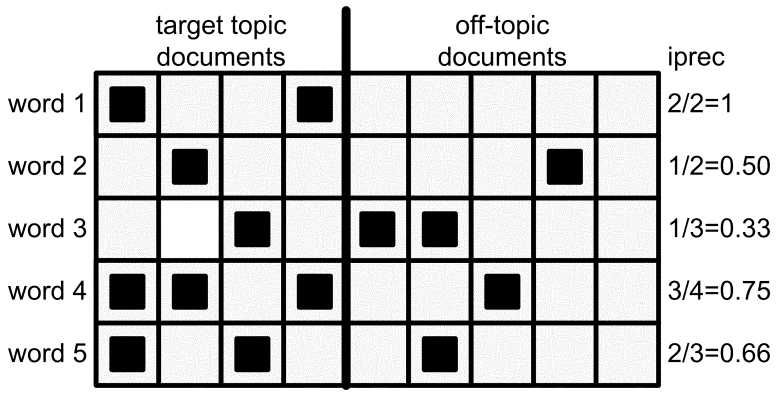

Figure 1: Example for individual precision. Rows stand for words and columns stand for documents.

3.1 Creating topic specific keyword lists

The techniques proposed in the following will require topic-specific keyword lists. These are created using the Precision-based Keyword Selection (PKS) algorithm described and evaluated in [10] in details. The PKS algorithm is based on the *individual precision* of words which is defined as follows:

Definition 1 (Individual precision)**.** *Individual precision of a keyword w is the precision of a selector for which $d \in S_w \Leftrightarrow w \in d$, that is, it selects documents containing w.*

Fig. 1 shows an example for the individual precision. Using this quality measure of the words the PKS algorithm collects words for a K_T keyword list for a given T target topic using the following definition:

Definition 2 (Topic specific keyword)**.** *$w \in K_T$ if and only if $P(T|w) \geq minprec$, that is, a word w is keyword of the topic T if and only if the conditional probability of the topic given w is present in a document is higher than a predefined minimal limit $minprec$.*

As the probability $P(T|w)$ is the expected individual precision of w, the minimal limit in the definition is called minimal precision limit *minprec*. The PKS algorithm selects topic specific keywords and selects documents containing at least one keyword. The *minprec* limit is set to allow the maximal F-measure for the selection of documents in the given topic as shown in Fig. 2.

It should be noted that there are many words in the documents which are very rare (spelling errors belong to this category too) and thus might have very high precision if their few documents belong to the same topic. To avoid such words the system does not consider words appearing in less than 0.5% of the documents in the training set. Too frequent words (stopwords) usually cannot be topic specific keywords because their expected precision is low: it is near the a-priori probability

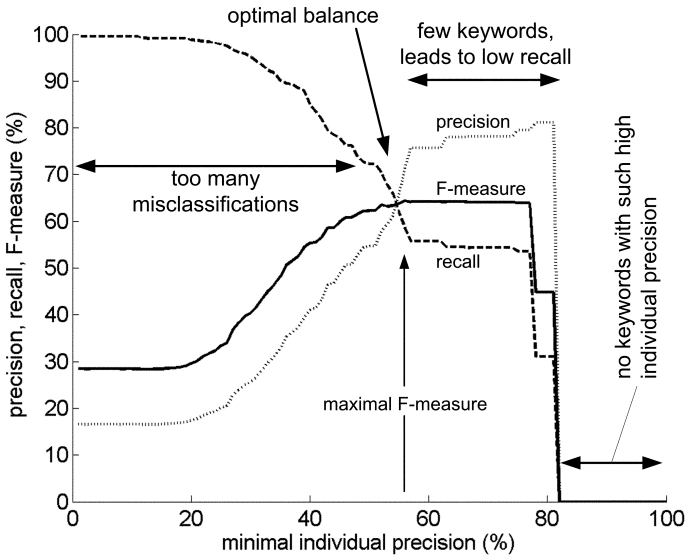

Figure 2: The PKS algorithm collects keywords starting from 100% minimal precision and decreasing it until the F-measure reaches its maximum.

of the target topic which is far lower than the *minprec* limit. In spite of this, if the number of topics is too high and it leads to a too low minimal precision, the minimal precision limit has to be limited to a predefined minimal value in order to avoid stopwords being keywords.

3.2 Searching for similar documents

Using the keyword lists created with the PKS algorithm a compact representation for every document can be created:

Definition 3 (Compact document representation)**.** *The compact document representation of a document d is the pair $(l, \mathbf{p})$ where l is the unique identifier of the keyword list used for the representation, and $\mathbf{p}$ is the presence vector containing the binary presence indication of the keywords in the keyword list l.*

The comparison of two documents means counting the common keywords. If the two documents are represented using the same keyword list, this is a simple inner product. If the keyword lists differ, the two vectors have to be mapped into the *global keyword space* where every keyword has a unique dimension. If all the keyword lists contain the unique dimension index of the contained keywords, it can be done easily.

For the sake of simplicity, document vectors are considered in the global keyword space in the following if not stated otherwise, as these vectors are equivalent to the document representations.

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

$$\text{similarity}(\underline{t}, \underline{d}) = \underline{t}^T \cdot \underline{d} \tag{1}$$

Searching for documents similar to a set of local documents is performed with the merged *base document vector*:

Definition 4 (Merged base document vector $\underline{b}$)**.** *Merged base document vector* $\underline{b}$ *representing all local (base) documents for comparisons with remote documents is defined as*

$$\underline{b} := \text{sign}\Big(\sum_{d \in B} \underline{d}\Big) \tag{2}$$

where B *is the set of base documents.*

Given a remote compact document representation it is transformed into the global keyword space. The result is the $\underline{r}$ remote document vector. The user is notified if

$$\text{similarity}(\underline{b}, \underline{r}) \geq th \tag{3}$$

where th is the minimal similarity measure threshold which a remote document must have to the base documents in order to notify the user about its availability. The threshold is defined by the user and allows controlling the balance between precision and recall: lower threshold notifies about more documents while higher threshold notifies only if a document is really very similar to the base documents.

In order to transform a document representation to the global keyword space, the index of the keywords (their associated dimensions) are required. This information is supplied with the keyword lists themselves. If a document is represented with a keyword list not known by the mobile device, the keyword list is simply downloaded from a central repository using the keyword list identifier in the document representation. If the new keyword list has common keywords with that of the base documents, it should be stored for later use. If it is not the case, only the identifier of the keyword list should be stored in order to remember that documents using this keyword list for the document representation cannot have any common keywords with the base documents. We believe that after some initial time, unknown keyword lists will be rare. (As a possible enhancement, the central repository could tell the mobile devices which keyword lists have common keywords with a given set of keyword lists (the ones of the base documents)).

It should be noted that this type of document representation can be further improved by word stemming, part-of-speech tagging, and by adding synonyms of the keywords to the representations. Applying stemming of part-of-speech tagging would require significantly more resources. According to the extension of the documents with synonyms of the keywords, the previous paper of the authors [11] is referred to which describes a method aiming to handle synonyms and hypernyms of the keywords.

Using the methods described here, a mobile device can represent its documents for others and it can search for remote documents that are similar to the base

documents. In the following section we describe an extension which can reduce the number of keyword lists required to be checked during the topic identification process. It allows the mobile device to retrieve and store fewer keyword lists and to complete the topic identification by using fewer keyword lists.

4 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. 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. In the current description we employ a two level classifier ensemble: the first level is using the topic sets and the second is using the keyword list of the triggered topics. 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 us to remove 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, final keyword lists are created for the remaining topic sets using the PKS algorithm described earlier.

During the classification, 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 (we call these topic sets the *triggered topic sets*), and finally only the keyword lists of topics in triggered topic sets (*triggered topics*) are compared to the document (Fig. 3).

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

4.1 Creating easy-to-identify topic patterns

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. The topics not covered by topic sets are covered with separate topic sets containing only one topic.

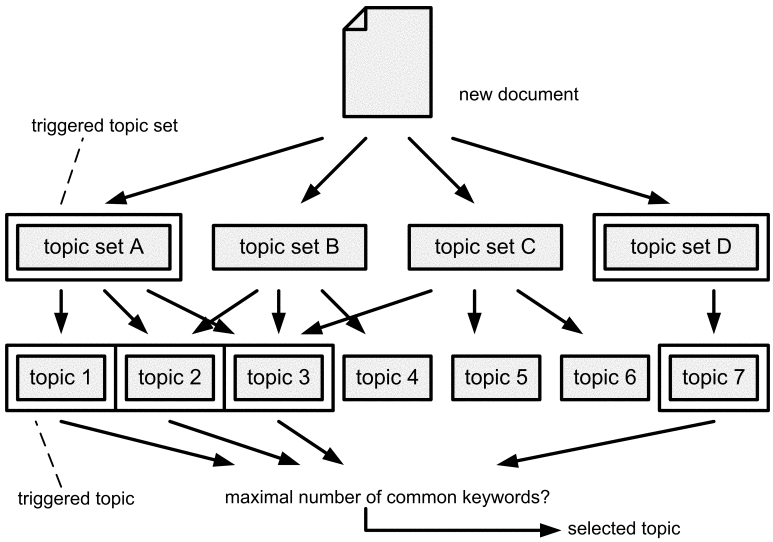

Figure 3: Topic sets. Only triggered topics (topics of triggered topic sets) are checked during topic classification.

4.1.1 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. Similarly to the individual precision, we define individual F-measure and assign every w word that set of topics for which w achieves the highest iF individual F-measure (Fig. 4).

Definition 5 (Individual F-measure)**.** *Individual F-measure $iF(w, \mathcal{T})$ of a w word regarding a $\mathcal{T}$ set of topics is the F-measure of a classifier selecting exactly the documents containing w.*

$$\mathcal{T}^{opt}(w) = \arg\max_{\mathcal{T}}\{iF(w, \mathcal{T})\} \tag{4}$$

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. A similarly defined individual recall is unsuitable as well because it does not take the precision into consideration which is still very important as we are going to create keyword lists for the topic sets using PKS.

The FTSC algorithm is searching for the topic set $\mathcal{T}^{opt}(w)$ for every w word in a greedy way: it adds the T topics to the topic set in descending $iF(w, T)$ order

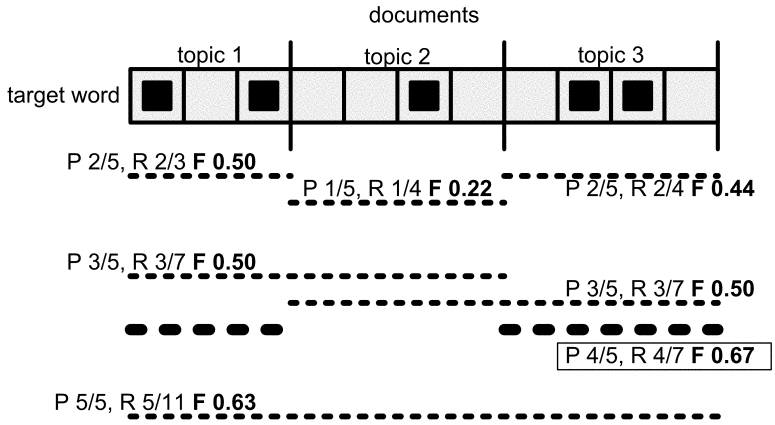

Figure 4: Example for the topic optimization for a given word. P, R and F stand for precision, recall and F-measure respectively. In this example, the individual F-measure is maximized by a topic set containing topics 1 and 3.

until the individual F-measure is maximized (Fig. 5).

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:

Proposition 1. *The FTSC algorithm selects the optimal topic set* $\mathcal{T}(w) = \mathcal{T}^{opt}(w)$ *for every word if the a-priori topic probabilities are equal for all topics.*

4.1.2 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. For example, if a topic set covers all topics, we cannot take advantage of it. 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 classification of every document 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. But topic sets may be overlapping and keywords may cause misclassifications as well. The precision and recall of the resulting classification is calculated and topic sets fulfilling the following conditions are preserved:

- Sufficiently high precision and recall. If a topic set has too low precision or recall, it is discarded. In our experiments, the minimal limit was set to 0.5

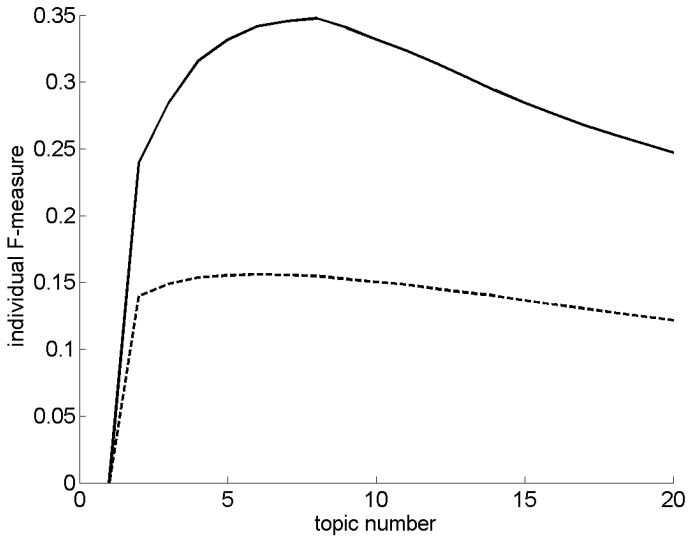

Figure 5: Evolution of individual F-measure in the 20 Newsgroups data set while increasing the number of target topics, presented for two example words. The size of the optimal topics sets are 8 and 6 in this case.

for both precision and recall.

- It 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 our experiments every topic had to have a 0.6 recall inside the topic set.

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

- 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. We believe that such topic sets are 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.

4.1.3 Creating additional topic sets

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

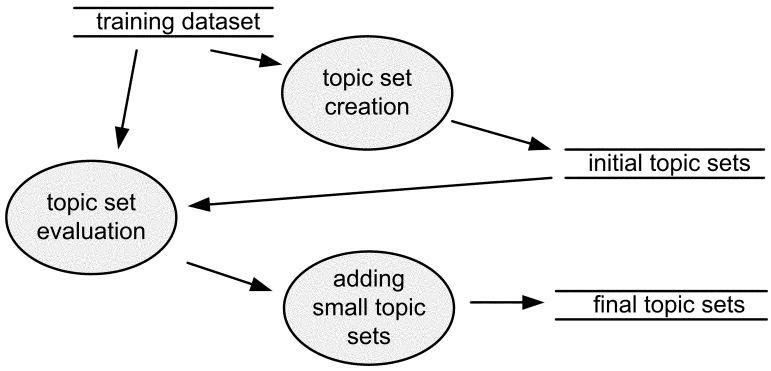

Figure 6: Data flow diagram of the FTSC algorithm. The training set is used to create the initial topic sets and the evaluation phase creates the final topic sets by modifying or removing worse topic sets and adding new ones if necessary.

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

The data flow diagram of the FTSC algorithm is presented in Fig. 6.

4.1.4 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 independently.

The trained first level of the classifier consists of a set of pairs

$$\{\mathcal{T} = \{T_1, T_2, ..., T_n\}; K_{\mathcal{T}}\} \tag{5}$$

where the first element is the set of topics contained in the current topic set and the second element is the keyword list for the topic set.

4.2 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.

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 just as in the case without the topic sets, however not triggered topics are not checked because they are considered to be "hopeless".

Unfortunately there are always topics which are not covered by the initial topic sets and they have to be placed in a topic set containing only one topic. These are the *small topic sets* mentioned in the previous section. 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* (containing more than one topic), the *small topic sets* are not checked because we assume that the real topic of the document is covered by the big topic sets. We call this extension *Small Sets on Demand* (SSD) because small topic sets are only checked if there seems to be a need for it.

Formally, given the d document, the $Trig(d)$ set of triggered topic sets contains the topics sets having common keywords with the document.

$$Trig(d) = \{\mathcal{T} : |K_{\mathcal{T}} \cap d| \geq 0\} \tag{6}$$

The Small Sets on Demand means

$$Trig'(d) = \{\mathcal{T} \in Trig(d) : |\mathcal{T}| \geq 1\} \tag{7}$$

and $Trig'(d)$ is used instead of $Trig(d)$ if $|Trig'(d)| \geq mb$ where mb is the minimal number of triggered big topic sets to skip the small topic sets entirely.

The $C(d)$ set of topics to check is the union of all topics in the triggered topic sets:

$$C(d) = \bigcup_{\mathcal{T} \in Trig(d)} \mathcal{T} \tag{8}$$

And finally the identified topic of the document d is the topic with the most common keywords with the document among the checked topics:

$$T(d) = \arg\max_{T \in C} \{|K_T \cup d|\} \tag{9}$$

5 Measurements

This section presents measurement results according to various aspects of the topic set based document topic identification and the search for similar documents. The measurements were performed using the commonly used data sets 20 Newsgroups [12] and the Reuters Corpus Volume 1 (RCV1, LYRL2004 split) [13].

First we present results according to the classifier ensemble used for topic identification in the 20 Newsgroups data set because the interpretation is easier with this data set. After that we present the evaluation on RCV1 and finally we present measurement results about the searching for similar documents.

5.1 Evaluation of the classifier ensemble

The most important condition the two level classifier has to satisfy is the minimal degradation in the classification performance. Table 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.

From Table 1 we can draw the following conclusions:

- 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. 7. This means that around 16-17 keyword lists 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 is classified without checking the small topic sets (thus checking only $5 + 3.12$ keyword lists in average).
- 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.

Table 1: Classification results, with the 20 Newsgroups data set, with and without topic sets (no Small Sets on Demand, every topic set is always checked) and with SSD using mb (minimal number of triggered big topic sets to skip checking of the small topic sets) 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

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 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.

Details about the topic sets are presented in Table 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 is clearly about

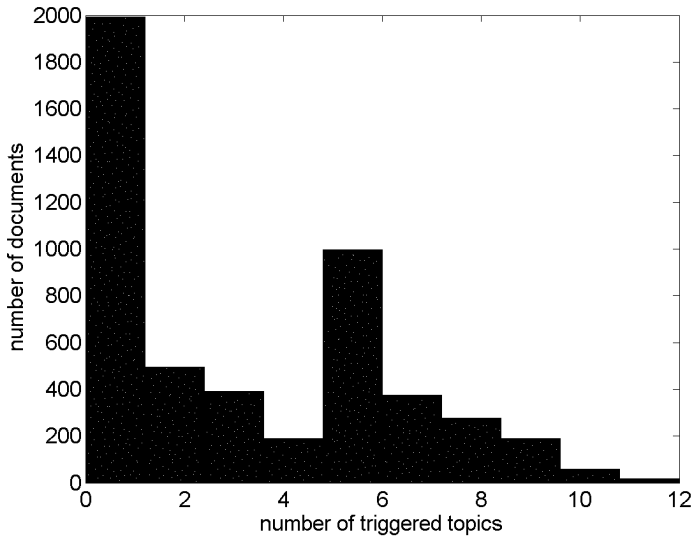

Figure 7: Histogram of the number of triggered topics in 20 Newsgroups. The mean value is 3.25 topics.

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.

If we evaluate the FTSC method as a method for ordering topics into hierarchy, we 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 caused 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. 8. During the application of the system, documents of a given topic may trigger multiple topic sets. The corresponding measurement results are presented in Fig. 9. 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. 10.

Table 2: Topic sets in 20 Newsgroups. 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	P 57	christians church pgp soviet
	talk.politics.mideast	R 70	muslim heaven spirit
	sci.crypt		secret israeli secure arab
	sci.space		security orbit encryption
2	rec.sport.baseball	P 71	teams team turkish baseball
	rec.sport.hockey	R 65	season hockey player fans nhl
	talk.politics.mideast		league players israeli arab
3	comp.os.ms-windows.misc	P 62	server microsoft window
	comp.windows.x	R 68	motif
4	talk.politics.guns	P 63	cars pgp citizens cup economic
	rec.sport.hockey	R 69	guns secure wings federal fbi
	sci.crypt		warrant security weapons keys
	rec.autos		enforcement hockey coverage
	talk.politics.misc		gun criminal crime secret nhl
			car police encryption agents
5	alt.atheism	P 73	atheist christians bible holy
	soc.religion.christian	R 70	belief morality sin heaven god
			church faith

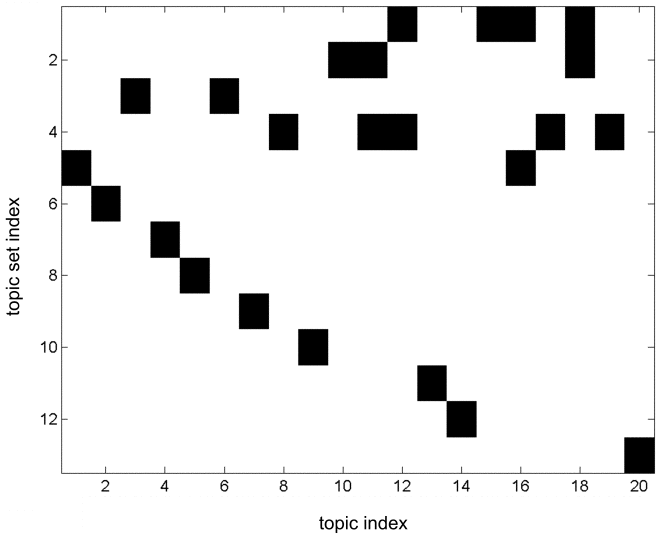

Figure 8: Topic sets retrieved for the 20 topics of the 20 Newsgroups data set.

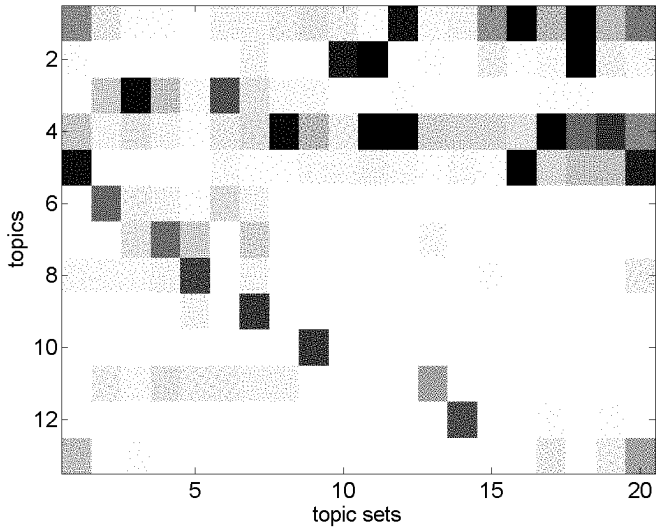

Figure 9: Triggering of topic sets in 20 Newsgroups. The more often a topic set is triggered by the documents of a topic the darker is the rectangle corresponding to the (topic set;topic) pair. The measurement used SSD with $md = 2$.

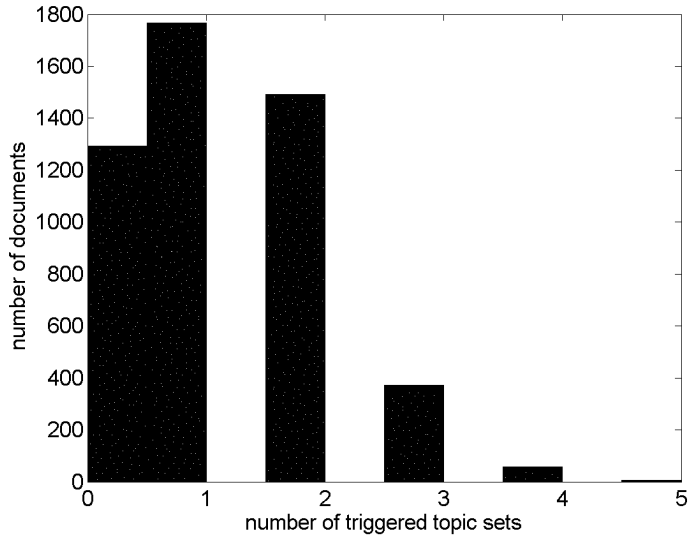

Figure 10: Histogram of the number of triggered topic sets in 20 Newsgroups. Mean value is 1.23 topic sets.

5.2 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 we 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 we removed the predefined upper level topics *CCAT*, *GCAT*, *MCAT* and *ECAT* from the data set, and some further topics which contained too few documents were removed too. 78 topics remained.

We applied exactly the same methods to the RCV1 data set as to the 20 Newsgroups previously. The classification results were obtained without topic sets and with topic sets using SSD with $mb = 2$. The results presented in Table 3 are similar to the ones for the 20 Newsgroups (Table. 1). We believe that 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 constant.

Table 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

As RCV1 has much more topics than 20 Newsgroups, the capability of the topic sets to decrease the number of keyword lists checked with a given document during topic identification is more significant: although there are 78 topics, the mean number of triggered topics is 37.8. If no small topics are required to check, only 12 big topic sets are checked and 4.7 of these big topic sets are triggered by a document in average. This means that if there is no need to check small topic sets, the classification of a document requires the check of 12 big topic sets and those trigger 37.8 topics in average, so $12 + 37.8 = 49.8$ keyword lists are checked in average, instead of 78.

Examples on the merged topics and keyword lists of the topic sets are presented in Tables 4 and 5. Due to the high number of topics we cannot present every topic set with all its keywords. 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 4 and 5 the topic sets have clearly captures 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.

Table 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

5.3 Evaluation of similar document search

In order to have an overview on the task of searching for similar documents first we identified the topic of documents in the 20 Newsgroups data set using the techniques discussed before. Using the document representations we calculated the document similarity matrix to have an overview on the similarity structure. The matrix is presented in Fig. 11. If we used a single base document which corresponds to a row (or column) vector in the matrix and the similarity threshold would be $th = 1$, the set of selected documents would be the set of documents indicated by dots in the figure.

In order to evaluate the search method we simulated it using various settings:

- The number of base documents were 1, 5, 10, 15 and 20.
- Threshold values between 1 and 10 were investigated.
- The base documents were always taken from one topic, but every 20 topics were investigated this way.

- For every setting of the previous parameters 100 measurements were performed by selecting random sets of base documents with the given size.

Table 5: 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 23.6%	equity markets, performance	pretax, dax, pfennig, outperform, payout, canon, goldfield...
2 13.6%	commodity markets, sports	cup, cricket, medal, coach, sheffield, yorkshir, wimbledon, athlet, mideast, lbs, intermonth, goalkeep, unbeat, semifin...
3 8.0%	domestic politics, international relations	podium, obstruc, diplom, palestin, gaza, elector, bosnian, sworn, boycot, disarm, peacemak, provoc, breakaway, proclaim...
4 13.6%	commodity markets, equity markets	unlead, gallon, composit, backward, meal, mideast, lbs, intermonth, overbought, telefon, sunseed, backfat, cottonseed...
5 61.9%	commodity markets, performance, equity markets, money markets, bond markets, 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, midpoint, interbank, forint, overvalu, oversold, cottonseed, financier, hectic, nationsbank...
6 29.4%	crime, law enforcement, performance, domestic politics, biographies, personalities, people...	widow, kidnap, jail, convict, extraordin, cocain, crim, murd, amnest, interpol, imprison, mafia, heroin, theft, horror, cardiac, bodyguard...
7 13.9%	crime, law enforcement, domestic politics, legal/judicial	courtroom, conspir, kidnap, jail, corrupt, truth, crimin, fbi, arrest, interpol, heroin, motorway, bodyguard...
8 6.0%	equity markets	chip, composit, fts, nikkei, dax, ibi, cac, seng, komercn, sse, nse, tisc, telefon...
9 14.6%	domestic politics, commodity markets, sports	cargoe, cricket, halftim, medal, coach, wimbledon, athlet, octan, goalkeep...

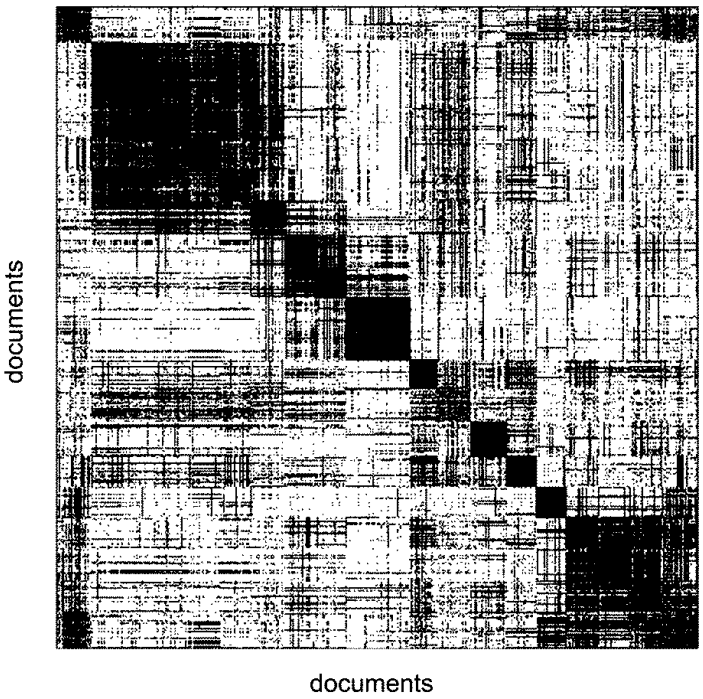

Figure 11: Document similarity matrix of 20 Newsgroups. Dots indicate non-zero similarity of the documents corresponding to the row and column. As the documents are ordered by topics along the axes topics are visible as rectangles along the main diagonal because documents inside the same topic tend to contain common keywords.

The results are presented in Fig. 12. It is clear that increasing the threshold increases precision and decreases recall as less documents have the chance to be selected and documents from the target topic (topic of the base documents) have usually more common keywords with the investigated remote document. Using one single base document provides very few keywords for the similarity search thus it leads to low recall values although with higher precision. More base documents provide more keywords, thus, it increases the recall but leads to more misclassifications as well, because more keywords introduce more chances for false classifications. As the threshold is defined by the user the balance between high precision and low recall (few false notifications but few retrieved documents), or lower precision and high recall (more mistakes but more found documents) can be set according to the preferences of the user.

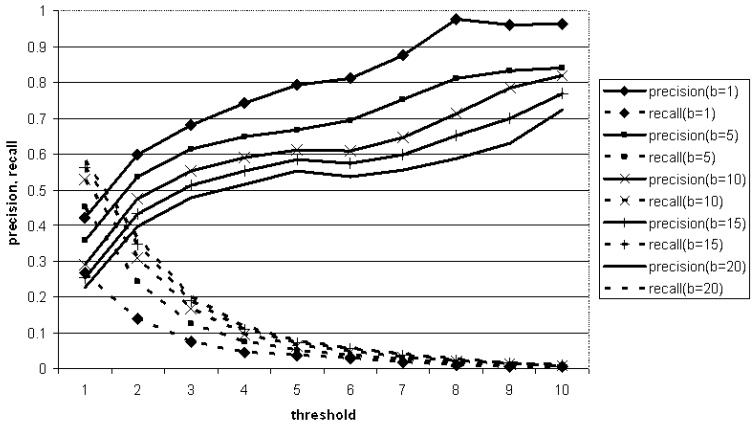

Figure 12: Results of the search for similar documents. Precision and recall is presented for various b number of baseline documents in the function of the th threshold.

6 Conclusions

We proposed a similar document retriever system which employs topic specific keywords to create compact topic representations allowing topic comparisons without downloading a whole document. The topic identification is an important step during the creation of the compact document representation. The improvement of this step presented in this paper allows the topic identification to download and process less possible topics. This accelerates the procedure and reduces communication traffic. The details of the applied algorithms and experimental results were presented in this paper.

Acknowledgements: This work has been fund of the Hungarian Academy of Sciences for control research and the Hungarian National Research Fund (grant number T68370).

References

[1] B. Forstner and H. Charaf. Neighbor selection in peer-to-peer networks using semantic relations. *WSEAS Transactions on Information Science and Applications*, Volume 2(Issue 2):239–244, February 2005. ISSN 1790-0832.

[2] W. Buntine. Topic-specific scoring of documents for relevant retrieval In *Proceedings of ICML 2005 Workshop 4: Learning in Web Search*, 7 August 2005, Bonn, Germany, 2005.

[3] Paul-Alexandru Chirita and Claudiu S. Firan and Wolfgang Nejdl. Summarizing local context to personalize global web search. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pp. 287–296, Arlington, Virginia, USA, 2006., ACM Press.

[4] Sholom M. Weiss, Nitin Indurkhya, Tong Zhang, Fred J. Damerau, editor. *Text Mining, Predictive Methods for Analysing Unstuctured Information*. Springer, 2005.

[5] B. Fortuna, D. Mladenić, and M. Grobelnik. Semi-automatic construction of topic ontology. In *Proceedings of SIKDD 2005 at multiconference IS 2005*, Ljubljana, Slovenia, 2005.

[6] L. R. Oded Maimon, editor. *The Data Mining and Knowledge Discovery Handbook*. Springer, 2005.

[7] S. S. Keerthi. Generalized lars as an effective feature selection tool for text classification with svms. In *Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany*, 2005.

[8] J. Yan, N. Liu, B. Zhang, S. Yan, Z. Chen, Q. Cheng, W. Fan, and W.-Y. Ma. Ocfs: optimal orthogonal centroid feature selection for text categorization. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference*, pp. 122–129, New York, NY, USA, 2005. ACM Press.

[9] Tao Li, Shenghuo Zhu, and Mitsunori Ogihara Hierarchical document classification using automatically generated hierarchy In *Journal of Intelligent Information Systems* Springer Netherlands, Volume 29, Number 2, 2007.

[10] K. Csorba and I. Vajk. Supervised term cluster creation for document clustering. *Scientific Bulletin of Politehnica University of Timisoara, Romania, Transactions on Automatic Control and Computer Science*, Vol. 51, 2006.

[11] K. Csorba, I. Vajk Improving Document Similarity Measurement for Mobile Environment with Document Extension In *ECML PKDD 2008, Ubiquitous Knowledge Discovery Workshop* Antwerp, Belgium, 2008. `http://www.ecmlpkdd2008.org/`

[12] K. Lang. NewsWeeder: learning to filter netnews. In A. Prieditis and S. J. Russell, editors, *Proceedings of ICML-95, 12th International Conference on Machine Learning*, pages 331–339, Lake Tahoe, US, 1995. Morgan Kaufmann Publishers, San Francisco, US.

[13] Lewis, D. D.; Yang, Y.; Rose, T.; and Li, F. RCV1: A New Benchmark Collection for Text Categorization Research. In *Journal of Machine Learning Research*, 5:361-397, 2004. `http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf`.

Appendix: formal proof of proposition on optimality of topic set selection in FTSC

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

Proof. We 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 $r = c/t$.

$$F = \frac{2 \cdot p \cdot r}{p + r} = \frac{2 \cdot c \cdot c}{s \cdot t(c/s + c/t)} = \frac{2 \cdot c}{t + s} \tag{10}$$

If we search for the optimal topic set for a given w word, the s number of selected documents is constant. We assume that the t target document number is the same for every topic (assuming equal a-priori topic probability). A topic set containing $|\mathcal{T}| = n$ topics and maximizing F-measure is maximizing

$$F = \frac{2 \cdot \sum_{T \in \mathcal{T}} c_T}{n \cdot t + s} \tag{11}$$

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

$$iF(w, T) = \frac{2 \cdot c_T}{t + s} \tag{12}$$

where the denominator is topic independent, $\mathcal{T}$ has to contain the n topics with the highest individual F-measure regarding w. If we add the topics to $\mathcal{T}$ in decreasing individual F-measure order, the $\mathcal{T}$ maximizing F-measure is a global optimum. □