# Improving Document Similarity Measurement for Mobile Environment with Document Extension

Kristóf Csorba and István Vajk

Budapest University of Technology and Economics
Department of Automation and Applied Informatics
{kristof|vajk}@aut.bme.hu

**Abstract**
This paper presents a new method for searching for documents which have similar topics to a given set of documents. It is designed to help mobile device users to search for documents in a peer-to-peer environment which have similar topic to the ones on the users own device. The algorithms are designed for slower processors, smaller memory and small data traffic between the devices. These features allow the application in an environment of mobile devices like phones or PDA-s. The keyword list based topic comparison is enhanced with a document extension to allow comparison of documents with similar topics but few or no common keywords. The architecture, the employed algorithms and the experimental results are presented in this paper.

## 1 Introduction

Document classification is usually a resource consuming task. The most important source of the complexity is the dimension number of the feature space. In feature space models, documents are represented initially by $n$ dimensional vectors, where there is a dimension for every word (or term in general) in the document set. This can easily lead to dimension numbers higher then 10,000. As these vectors and the term document matrix built from them is very sparse, many dimensionality reduction techniques have been proposed including feature selection techniques like information gain-based feature selections [9], double clustering [11], least angle regression [4] and various feature extraction techniques applied to the term document matrix such as singular value decomposition [7] or orthogonal locality preserving indexing [1]. [14] proposes a method for topic specific keyword selection, which is based on word weights in document vector centroids. These methods may select keywords with negative weight as well, but as we will see, our similarity measure cannot take negative weight into account making such keywords unacceptable and thus these methods not applicable.

The technique we contribute in this paper was designed for mobile phone environment. We assume that the user interests can be characterized by the set of documents stored locally. If a mobile device was able to search for similar documents to the ones stored on it, it would be able to notify the user about documents which might be of interest. Using these techniques the users would not have to define search phrases to discover interesting information in a peer-to-peer network. Similar, ”topic by example” search approaches are investigated in [15], but in the current scenario, this process is done by mobile devices which is why the algorithms have to satisfy two important conditions: on one hand the algorithms are not allowed to consume too many resources (processor time, memory and power) and on the other hand a low communication traffic has to be maintained to reduce financial communication costs and energy consumption of wireless transmissions.

These conditions significantly reduce the applicability of many common document classification techniques: complicated arithmetic, huge matrices and weight vectors for the terms, all these can easily lead to serious performance problems on a mobile phone. To avoid these problems we designed an algorithm package which enables mobile devices to describe the topic of their documents and compare them with documents on other devices. They use small amount of communication, fast and simple algorithms to accomplish this.

In order to keep computational complexity minimal, a similarity measure based on the number of common keywords between documents was selected. A keyword list is created for every possible document topic (Subsection 2.2). Documents are represented by their topic and a binary mask indicating the presence of absence of the keywords of the topic. As the keyword lists are globally accessible, transferring the topic identifier and a bitmask is sufficient to calculate the number of common keywords between two documents (Subsection 2.3). As this technique makes documents without common keywords have zero similarity, a document extension is applied (Section 3) which is similar to query expansion techniques [10]. Generalizing keywords related to the originally contained keywords are added to increase the number of possible common keywords between related documents.

## 2 Document topic representation

The procedure described in this section is used to create topic specific keyword lists for very compact document representation.

### 2.1 Notations

The following notations are used in the description of the algorithms:

- $D$ is the set of documents in the training data set
- $d$ is an arbitrary document, a set of words
- $\mathbf{d}$ binary document vector of the document $d$. Every word is assigned a unique dimension.
- $T$ is an arbitrary document topic, a set of documents
- $T(d)$ is the topic of document $d$.
- A *keyword* is a word which is present in at least one keyword list. This means that they are words used in the topic representations.

- $K_T = \{w_1, w_2, ..., w_n\}$ is the keyword list for the topic $T$.
- $\mathbf{t}_i$ is a virtual document vector of a document containing exactly the keywords of the topic $T_i$.
- Given the topics $T$, $G$ and $H$, $T \supseteq G$ means that $G$ is subtopic of $T$. This relationship is transitive: $T \supseteq G \wedge G \supseteq H \rightarrow T \supseteq H$.

**Definition 1 (topic hierarchy).** *Topic hierarchy is a set of trees where every node corresponds to a document topic. A $T$ node is child of $G$ exactly if $T \subseteq G$.*

**Definition 2 (related topics).** *Two topics $G$ and $H$ are related exactly if $\exists T : T \supseteq G \wedge T \supseteq H$.*

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, $f$ is 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 $F = 2PR/(P + R)$.

### 2.2 Creating topic specific keyword lists

The compact document representation uses one bit for every keyword of the topic of the document to indicate its presence or absence. As the keywords cannot be weighted this way, keywords can have a weight of one (present, thus relevant for the document) or zero (absent, not relevant). This approach does not allow employing keywords with negative weight: common keywords can only increase the similarity. This constraint makes many feature selection techniques not applicable in this case.

On the other hand, the topic most suitable for the representation of a document has to be selected in order to assign the best suitable keyword list which is used later to create the binary mask for the document representation:

$$\widehat{T}(d) = \underset{i}{\operatorname{argmax}}\{\mathbf{d}^T \mathbf{t}_i\} \tag{1}$$

that is, the topic with the most keywords occurring in the document is selected.

Now the document representation is equivalent to

$$\mathbf{d}_r = \mathbf{d}. * \mathbf{t}_{\widehat{T}(d)} \tag{2}$$

where $.*$ indicates dimensionwise multiplication. This operation removes every word from the represented document which is not keyword of the estimated topic $\widehat{T}(d)$.

The precision of the topic identification is of key importance. Low precision can increase similarity to off-topic documents due to a document representation based on an off-topic keyword list. This means that the probability of a topic has to be high if one of its keywords is present in a document. This probability is called the (topic dependent) individual precision *iprec* of a word:

$$\text{iprec}(w, T_i) = P(T(d) = T_i | w \in d) \tag{3}$$

Our keyword selection algorithm PKS (Precision based Keyword Selection) selects keywords having $iperc(w)$ over a $minprec$ limit:

$$K_{T_i}(minprec) = \{w : \text{iprec}(w, T_i) \geq minprec\} \tag{4}$$

This ensures that all keywords satisfy a minimum limit for individual precision which can be considered a measure of topic specificity. $minprec$ is set to maximize the achievable F-measure using the resulting keyword list.

$$\text{select}(d, T_i, minprec) \leftrightarrow \mathbf{d}^T \mathbf{t}_i(minprec) > 0 \tag{5}$$

is a selector collecting documents containing at least one keyword of the topic $T_i$, where $\mathbf{t}_i$ is calculated from the keyword list $K_{T_i}$ using its definition and it depends on $minprec$ through Eqn. 4.

$$minprec_{final}(T_i) = \underset{minprec}{\text{argmax}}\{\text{F}(\text{select}(d, T_i, minprec))\} \tag{6}$$

where $F$ returns the F-measure of the given document selection.

Briefly, PKS selects all words with individual precision over a minimal limit where the limit maximizes the F-measure of the classifier using the resulting keyword list.

Very high $minprec$ leads to very good results in terms of precision but few keywords cover few documents and lead to low recall. The optimized keyword list is a trade-off between precision and recall.

It is important to mention that if we have a hierarchy of topics, we can create keyword lists for every topic on every hierarchy level. For the sake of simplicity, we will consider a two level hierarchy in this paper and thus call the topics on the higher hierarchy level *upper level topics* and the ones on the lower level *lower level topics.*

Due to the supervised nature of the PKS algorithm, it requires a labeled training document set for the calculation of the individual precision of all words. In a mobile device environment, this has to be done prior to the semantic searches, using a central server containing a training set. Possibilities for a distributed PKS running on the mobile devices and using fewer available documents (similar goals are mentioned in [16]), and the conditions when the keyword lists have to be regenerated due to the changes in the topic structure of the document set are subject of further research.

### 2.3 Searching for similar documents

Similarity of two documents $d$ and $f$ is defined as the number of common keywords which can be easily calculated from the compact document representations.

$$\text{similarity}(\mathbf{d}_r, \mathbf{f}_r) = \mathbf{d}_r^T \mathbf{f}_r \tag{7}$$

Defining a virtual document $l$ containing every keyword of every local document, the similarity of all local and one remote document can be calculated which is the importance of the remote document:

$$l = \text{sign} \sum_{d \in L} \mathbf{d}_r \tag{8}$$

$$\text{importance}(d) = \text{similarity}(\mathbf{l}, \mathbf{d}_r) \tag{9}$$

where $L$ is the set of local documents.

The user is notified about every remote document having an importance above a predefined limit. This limit can be set by the user in the form of some easy-to-understand settings like "retrieve many", "loosely related only", "strongly related only".

Using this technique, only a few bytes of data has to be retrieved about every remote document to calculate its importance. The whole document is downloaded only if it qualifies as an important enough document and the user asks for its retrieval after the notification.

An important proposition about the precision of the document search is the following:

**Proposition 1 (Precision of document search).** *If searching for similar documents to the local ones represented by l, the minprec minimal precision limit used during the keyword list creation is valid for the expected precision of the selection of similar documents if the* $\widehat{T}(d)$ *topic of the documents was identified correctly.*

It can be interpreted as a lower limit for the expected precision of similar document search which is derived from the quality of keyword lists used for the document representations. The formal proof is not presented due to space limitations.

## 3 Document extension

The method presented in the previous section has already the ability to search for similar documents. Unfortunately if two documents have related topic but they do not share any common keywords, their similarity measure will be zero. Document extension presented in the following slightly increases the similarity measure of documents which have related topics. Two documents, one about *hawk* and one about *dolphins* without any common keyword would have zero similarity. If the system recognizes *animal* to be a related generalizing concept to both hawks and dolphins, it can be added to both document representations which would increase the similarity. This way, the two documents will have a similarity higher then the similarity of two totally unrelated documents.

### 3.1 Collecting related generalizing concepts

Assuming a topic hierarchy, a related generalizing concept (RGC) for a given $w \in K_T$ keyword is another keyword $g \in K_G, T \subseteq G$ which appears often together with $w$. The first condition ensures that $g$ belongs to a parent topic of $T$ and this way it is a more general concept. The second condition ensures semantic relationship between the two keywords.

**Definition 3 (Related General Concepts Function (RGCF)).** *RGCF is the function assigning related general concepts (keyword) $v_i$ to the keyword $w$: $RGCF(w) = \{v_1, v_2, ..., v_n\}$.*

$$g \in \text{RGCF}(w) \leftrightarrow \tag{10}$$

$$g \in K_G \wedge w \in K_T \wedge T \subseteq G \tag{11}$$

$$\frac{\{d \in D : w \in d \wedge g \in d\}}{|D|} \geq mincorate \tag{12}$$

where *mincorate* is the minimal rate of co-occurrence.

It should be noted, that there are many similar related works aiming to support query expansion or document classification starting from a word ontology such as WordNet [5]. Due to the concept of our system using very topic specific keyword lists, even if we used an ontology to discover related words, we would still have to check keyword condition (4). Keyword relations not confirmed by the training set cannot be used because it could destroy the semantic properties of the document extension described in the following subsection.

### 3.2 Extension of documents

Extension of the documents with the related general concepts of their originally contained keywords can be done easily using RGCF:

$$\mathbf{d}_r^{ext} = (\mathbf{d}_r^T \mathbf{A})^T \tag{13}$$

$$\mathbf{A}_{w,v} = 1 \leftrightarrow w = v \vee v \in RGCF(w) \tag{14}$$

The multiplication of the document vector with the matrix **A** adds the RGC-s of the originally contained keywords to the document vector.

The most important feature of the document extension is summarized in the following proposition:

**Proposition 2 (Probability of extension).** *The chance that a keyword $v \in K_A$ is added to two documents $d$ and $f$ during the document extension which increases the similarity measure of the two documents, is higher if the two documents belong to related topics.*

This means that the document extension increases the similarity measure mainly of documents with related topics. This makes the document extension applicable to enhance topic comparisons of related documents. The formal proof is not presented due to space limitations.

## 4 Measurements

In this section, we present measurements which evaluate the capabilities of the techniques presented previously. The measurements were performed using the commonly used data set 20 Newsgroups [8]. 20 Newsgroups is a collection of 20 Usenet topics, each containing 1000 documents. From this data set, we used a subset containing 3 upper level and 6 lower level topics: *hardware.PC*, *hardware.MAC*, *sport.baseball*, *sport.hockey*, *science.electronics* and *science.space*. As the RGCF creation requires a hierarchic topic structure, many other standard data sets, like Reuters-21578 cannot be used. The measurements employed 4-fold-crossvalidation.

### 4.1 Baseline measurements

To have an overview on the complexity of the classification problem, multiple baseline measurements were performed with a linear (LDA) classifier were employed with 500, 1000, 2000, 3000, 4000 and 5000 keywords. The keyword selection applied a stopword removal and TFIDF based feature selection. Based on our observations, using more then 5000 keywords does not make significant improvement. The measurements were created with the software TMSK available with [12] using the default settings according to the parameters not mentioned here.

The results of these classifications are presented in Fig.1. The linear classifier (together with the TFIDF based feature selection) achieves a precision of about 85%, but requires lots of features to do this.

### 4.2 Keyword list creation and RGCF learning

The keyword list creation is performed with the PKS algorithm. This algorithm is evaluated in [3] in details. The effectiveness of PKS is indicated by the results of the other methods presented here because all of them operate on topic representations based on keyword lists. The keyword list sizes for the upper level topics were 32, 21 and 63 keywords. The keyword number of the lower level topics were always below 20. As the document topic comparison requires the transmission of the keyword list identifier and a binary mask of the size 1 bit for each keyword, the topic representations will not be longer than 18 bytes assuming 16 bit keyword list identifier and at most 128 keywords. Communication traffic of this size for each document is acceptable in most scenarios.

Examples on keywords are presented in Table 1 where the keywords representing the topic of a document about space shuttles is shown.

Examples for RGCF values are presented in Table 2. The document extension aims to support the topic comparisons and these RGC keywords are definitely related to the keywords which they are assigned to. (It should be emphasized that they are to *support comparison* of loosely related documents, not to increase similarity of already similar documents.)

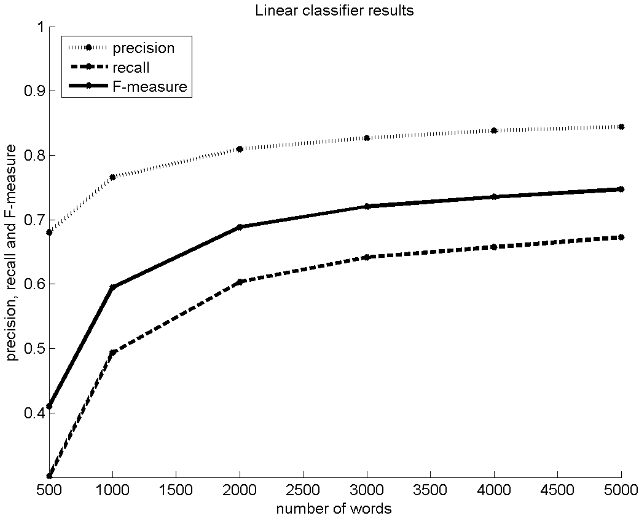

**Fig. 1.** Results with Linear classifier on 20 Newsgroups

**Table 1.** Example for keywords representing the topic of a 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...

### 4.3 Effects of the document extension

In order to have a visual on the similarity structure of the documents in the topic hierarchy, we measure the following similarities:

- Average lower level intra-topic similarity measure (L intra): this is the average similarity measure of documents in the same upper level and lower level topic.
- Average lower level inter-topic similarity measure (L inter): this is the average similarity measure of documents in the same upper level, but different lower level topic.
- Average upper level inter-topic similarity measure (U inter): this is the average similarity measure of documents in different upper level topics.

In order to check the effectiveness of the document extension, we compare the original similarity structure of the documents in the testing set to the results

**Table 2.** Related General Concepts Function (*RGCF*)

| keyword | related general concepts |
|---|---|
| baseball | season league |
| hockey | season teams sharks |
| defenseman | scoring league teams score defensive |
| penalties | season Pittsburgh scoring league shots |
| satellites | project probe radar planet antenna |
| shuttle | vehicle |
| astronomy | project science vehicle gravity probe engineering planet solar probes |
| lunar | project vehicle elements gravity objects probe radar planet cloud solar |

of the document extension. This is presented in Table 3. As it is clearly visible, the document extension strongly increases the similarity measure of documents on higher topic hierarchy levels if the documents belong to the same subtree in the topic hierarchy.

**Table 3.** Effects of document extension on document similarity measure

| | original | | | extended | | |
|---|---|---|---|---|---|---|
| | L intra | L inter | U inter | L intra | L inter | U inter |
| topic 1 | 0.3075 | **0.1232** | 0.0139 | 0.4165 | **0.1746** | 0.0178 |
| topic 2 | 0.3831 | **0.1544** | 0.0035 | 0.7647 | **0.3621** | 0.0044 |
| topic 3 | 0.3678 | **0.1192** | 0.0159 | 0.6522 | **0.2606** | 0.0202 |

As an example, Table 4 presents the keywords added to the document about space shuttles. It is clear that documents containing the additional keywords might have a loose relationship to space shuttles. Although the presence of these keywords alone implies non-zero similarity measure, this should still not be considered to be a clear indication of topic similarity.

**Table 4.** Keywords added to the representation of the document on space shuttles during document extension.

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

### 4.4 Searching for similar documents

Fig. 2 shows the results of a search for similar documents using the original document vectors and Fig. 3 the same with extended ones. The evaluation is performed on upper level topics because document extension increases the similarity measure mainly of documents inside the same upper level topic but in different lower level topics. Such document pairs have usually less or no common keywords but still related topic. The support of discovery of such document relationships is the aim of the document extension.

Precision is of key importance in our application, because low precision means many false notifications to the user. The increasing threshold obviously increases the precision and lowers the recall. The increasing number of local documents increases the set of used keywords, thus it increases the recall but makes more chances for misclassification which lowers the precision. It is clear that the document extension increases the recall significantly (more documents get non-zero similarity measure which was the aim of document extension). The degradation of precision is not significant for small local document numbers. For more local documents, a precision decrement is observable which is caused by the added parent keywords and their additional chance to cause false selection. Altogether, the results confirm that the document extension is suitable for the goal it was designed for.

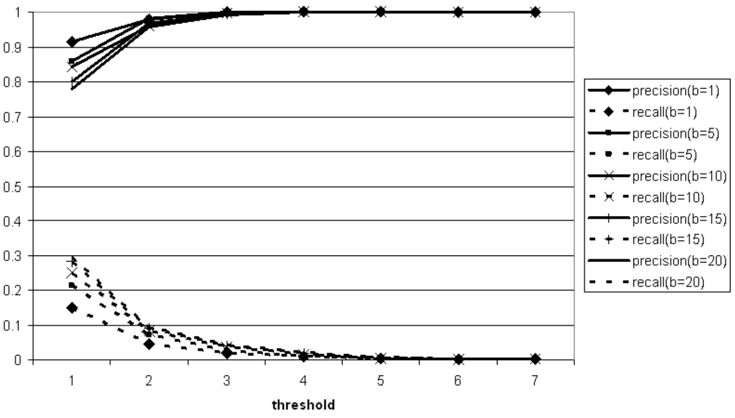

**Fig. 2.** Evaluation of the search for similar documents in the data set *without* document extension. For local document numbers $b$ 1, 5, 10, 15 and 20, we present the precision and recall of the selection, as a function of the threshold.

Compared to the baseline measurements, the precision of our algorithm is significantly higher, which is achieved with much less keywords (dimensions).

## 5 Conclusions

This paper presented a new topic identification and representation technique for mobile device environments. The various devices can search for documents

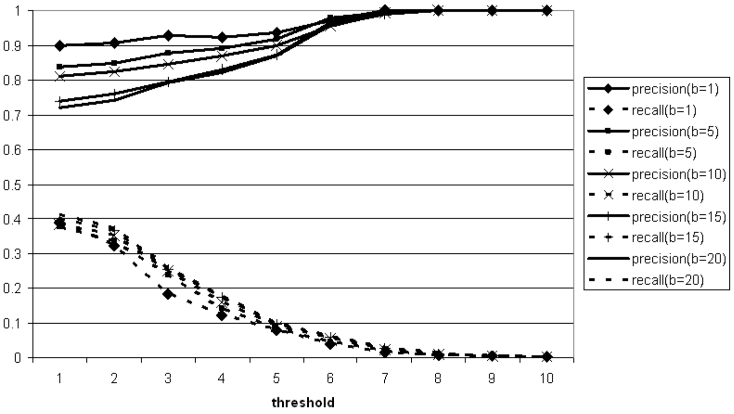

**Fig. 3.** Evaluation of the search for similar documents in the data set *with* document extension.

similar to the topics of the documents stored on them to find information that might be of interest for their users. Our system employs simple and fast algorithms and keeps the communication traffic low by employing very compact topic representations. As the retrieval of less documents is much less annoying then misclassifications, our system concentrates on high precision document selection even with lower recall. This property is ensured by the keyword selection algorithm which, in contrast with related works, is not allowed to select negative weighted features, but ensures high precision topic identification with the selection of only very topic specific keywords.

Documents with similar topics but few common keywords can be found with the help of document extension which employs a new technique for learning the semantic relationship of keywords while preserving the suitability for high precision topic identification.

In this paper, we presented the algorithms for the proposed system, and experimental results for the evaluation of multiple aspects of the techniques.

### 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. D. Cai and X. He. Orthogonal locality preserving indexing. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, New York, NY, USA, 2005. ACM Press.

2. P. Cimiano, A. Hotho, and S. Staab. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research*, 24:305–339, 2005.
3. 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(65)(No 3./2006.):49, 2006.
4. B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression, 2002.
5. C. Fellbaum, editor. *WordNet: An Electronic Lexical Database.* The MIT Press, Cambridge, Massachusetts, 1989.
6. B. Forstner, I. Kelenyi, and G. Csucs. *Towards Cognitive and Cooperative Wireless Networking: Techniques, Methodologies and Prospects*, chapter Peer-to-Peer Information Retrieval Based on Fields of Interest, pages 311–325. Springer Verlag, 2007.
7. G. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. Harshman, L. A. Streeter, and K. E. Lochbaum. Information retrieval using a singular value decomposition model of latent semantic structure. In Y. Chiaramella, editor, *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 465–480, Grenoble, France, 1988. ACM.
8. 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.
9. L. R. Oded Maimon, editor. *The Data Mining and Knowledge Discovery Handbook.* Springer, 2005.
10. Qiu, Yonggang, Frei, and H. P. Concept based query expansion. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Query Expansion, pages 160–169, 1993.
11. N. Slonim and N. Tishby. 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, 2000.
12. S.M. Weiss, N. Indurkhya, T. Zhang, F.J. Damerau, editor. *Text Mining, Predictive Methods for Analysing Unstuctured Information.* Springer, 2005.
13. 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 on Research and development in information retrieval*, pages 122–129, New York, NY, USA, 2005. ACM Press.
14. 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.
15. 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.
16. H. F. Witschel Estimation of global term weights for distributed and ubiquitous IR In *Proceedings of ECML 2006 Workshop 4: Ubiquitous Knowledge Discovery for users*, 18.-22. Sepember 2006., Berlin, Germany, 2006.