Text categorization on a multi-lingual corpus

Domonkos Tikk[1] and György Biró[2]

[1] Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
H-1117 Budapest, Magyar Tudósok körútja 2., Hungary
Email: tikk@tmit.bme.hu

[2] Department of Informatics, Eötvös Loránd Science University
H-1117 Budapest, Pázmány sétány 1/c, Hungary
Email: gbiro@tmit.bme.hu

Abstract

This paper presents experiments with a hierarchical text categorizer on a multi-lingual (English, French) corpus. The results obtained are very similar for both languages. The results allow us to apply in the near future cross-language text categorization that can be used to support automatic translation to create multi-lingual topic glossary.

1 Introduction

Traditionally, document categorization has been performed manually. However, as the number of documents explosively increases, the task becomes no longer amenable to the manual categorization, requiring a vast amount of time and cost. This has lead to numerous researches for automatic document classification. A text classifier assign a document to appropriate category/ies, also called *topic*, in a predefined set of categories.

Originally, research in text categorization addressed the *binary problem*, where a document is either relevant or not w.r.t. a given category. In real-world situation, however, the great variety of different sources and hence categories usually poses *multi-class* classification problem, where a document belongs to exactly one category selected from a predefined set [1, 2, 3]. Even more general is the case of *multi-label* problem, where a document can be classified into more than one category. While binary and multi-class problems were investigated extensively [4], multi-label problems have received very little attention [5].

As the number of topics becomes larger, multi-class categorizers face the problem of complexity that may incur rapid increase of time and storage, and compromise the perspicuity of categorized subject domain. A common way to

manage complexity is using a hierarchy[1], and text is no exception [6]. Internet directories and large databases are often organized as hierarchies; see e.g. Yahoo and WIPO's patent database[2].

Recently, we proposed a hierarchical text categorization approach and showed its effectiveness on two document collection [7, 8]. In this paper we apply it for a bi-lingual corpus of ILO (International Labour Organization), i.e. where all documents are given in English and in French language as well. Different languages require different kind of text procession; the indexing method needs to be customized for the specialities of various languages which necessitates first different stemmers and the application of other language-dependent heuristics.

The final goal of these experiments is to perform cross-language text categorization that can be used to support automatic translation to create multilingual topic glossary. Cross-language text categorization is a brand new branch of text mining and information retrieval [9].

The paper is organized as follows. Section 2 presents the approach applied in details. Section 3 report on our experience. The conclusion is drawn in Section 5.

2 The proposed method

The core idea of the categorization method is the training algorithm that sets and maintains category descriptors in a way that allows the classifier to be able to correctly categorize the most training, and consequently, test documents. We start the categorization with empty descriptors.

We now briefly describe the training procedure. First, when classifying a training document we compare it with category descriptors and assign the document to the category of the most similar descriptor. When this procedure fails finding correct category we raise the weight of such terms in the category descriptors that appear also in the given document. If a document is assigned to a category incorrectly, we lower the weight of such terms in the descriptors that appear in the document. We tune category descriptors by finding the optimal weights for each terms in each category descriptor by this awarding–penalizing method. The training algorithm is executed iteratively and ends when the performance of the classifier cannot be further improved significantly. See the block diagram of Figure 1 for an overview and details in Subsection 2.2 about the training algorithm. For test documents the classifier works in one pass by omitting the feedback cycle.

The rest of this section is organized as follows. Subsection 2.1 describes the topic hierarchy, vector space model and descriptors. Subsection 2.2 presents classification and the training method.

[1] In general hierarchy is considered to be an acyclic digraph; in this paper we restrict our investigation to tree structured hierarchies.

[2] `http://www.yahoo.com`, `http://www.wipo.int/ibis/`

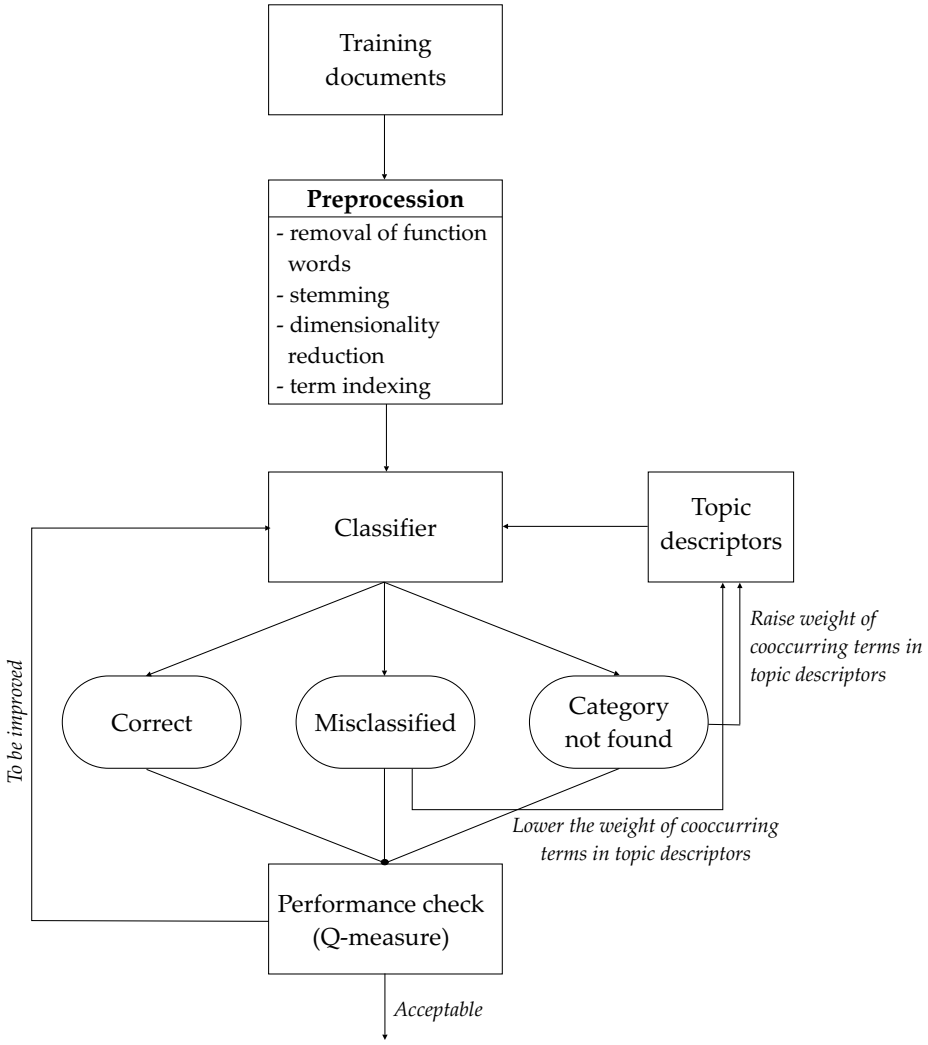

Figure 1: The flowchart of the training algorithm

2.1 Definitions

Let $\mathcal{C}$ be the fixed finite set of categories organized in a *topic hierarchy*. In this paper, we deal with tree structured topic hierarchies.

Let $\mathcal{D}$ be a set of text documents and $d \in \mathcal{D}$ an arbitrary element of $\mathcal{D}$. In general, documents are pre-classified under the categories of $\mathcal{C}$, in our case into leaf categories. We differentiate training, $d \in \mathcal{D}_{\text{Train}}$, and test documents, $d \in \mathcal{D}_{\text{Test}}$, where $\mathcal{D}_{\text{Train}} \cap \mathcal{D}_{\text{Test}} = \emptyset$, and $\mathcal{D}_{\text{Train}} \cup \mathcal{D}_{\text{Test}} = \mathcal{D}$. Training documents are used to inductively construct the classifier. Test documents are used to test the performance of the classifier. Test documents do not participate in the

construction of the classifier in any way.

Each document $d_j \in \mathcal{D}$ is classified into a leaf category of the hierarchy. No document belongs to non-leaf categories. We assume that a parent category owns the documents if its child categories, i.e., each document belongs to a *topic path* containing the nodes (representing categories) from the root to a leaf. Formally,

$$\text{topic}(d_j) = \{c_1, \ldots, c_q \in \mathcal{C}\} \tag{1}$$

determines the set of topics document d_j belongs to along the topic path (c_1 is the highest). Note that the root is not administrated in the topic set, as it owns all documents. As a consequence of our previous condition, c_q is a leaf-category. The index refers to the depth of the category.

Texts cannot be directly interpreted by a classifier. Because of this, an indexing procedure that maps a text d into a compact representation of its content needs to be uniformly applied to all documents (training and test). We choose to use only words as meaningful units of representing text, because the use of n-grams (word sequences of length n) increases dramatically the storage requirement of the model, and, as it was reported in [10, 11], the use of more sophisticated representation than simple words do not increase effectiveness significantly.

As most research works, we also use the *vector space* model, where a document d_j is represented by a vector of term *weights*

$$d_j = \langle w_{1j}, \ldots, w_{|\mathcal{T}|j} \rangle, \tag{2}$$

where $\mathcal{T}$ is the set of *terms* that occurs at least ones in the training documents $\mathcal{D}_{\text{Train}}$, and $0 \leq w_{kj} \leq 1$ represents the relevance of kth term to the characterization of the document d. Before indexing the documents *function words* (i.e. articles, prepositions, conjunctions, etc.) are removed, and stemming (grouping words that share the same morphological root) is performed on $\mathcal{T}$. We utilize the well-known tf×idf weighting [12], which defines w_{kj} in proportion to the number of occurrence of the kth term in the document, o_{kj}, and in inverse proportion to the number of documents in the collection for which the terms occurs at least once, n_k:

$$w_{kj} = o_{kj} \cdot \log\left(\frac{N}{n_k}\right), \tag{3}$$

Term vectors (2) are normalized before training.

We characterize categories analogously as documents. To each category is assigned a vector of *descriptor term weights*

$$\text{descr}(c_i) = \langle v_{1i}, \ldots, v_{|\mathcal{T}|i} \rangle, \quad c_i \in \mathcal{C} \tag{4}$$

where weights $0 \leq v_{1i} \leq 1$ are set during training. All weights are set initially to 0. The descriptor of a category can be interpreted as the prototype of a document belonging to it.

2.2 Classification and training

2.2.1 Classification

When classifying a document $d \in \mathcal{D}$ its term vector (2) is compared to topic descriptors (4) and based on the result the classifier selects (normally) a unique category. At a given stage of the classification only a subset of topic are considered as potential category; this is based on the following greedy selection process. The classification method works downward in the topic hierarchy level by level. First, it determines the best among the top level categories. Then its children categories are considered and the most likely one is selected. Considered categories are always siblings linked under the *winner category* of the previous level. Classification ends when a leaf category is found.

Let us assume that at an arbitrary stage of the classification of document d_j we have to select from k categories: $c_1, \ldots, c_k \in \mathcal{C}$. Then we calculate the conformity of term vector of d_j and each topic descriptors $\mathrm{descr}(c_1), \ldots, \mathrm{descr}(c_k)$, and select that category that gives the highest conformity measure. We applied the unnormalized cosine measure that calculates this value as a function f of the sum of products of document and descriptor term weights:

$$\mathrm{conf}(d_j, \mathrm{descr}(c_i)) = f\left(\sum_{k=1}^{|\mathcal{T}|} w_{kj} \cdot v_{ki}\right), \tag{5}$$

where $f : \mathbb{R} \to [0, 1]$ is an arbitrary smoothing function with $\lim_{x\to 0} f(x) = 0$ and $\lim_{x\to\infty} f(x) = 1$. The smoothing function is applied (analogously as in control theory) to alleviate the oscillating behavior of training.

McCallum [13] criticized the greedy topic selection method because it requires high accuracy at internal (non-leaf) nodes. In order to alleviate partly the risk of a high level misclassification, we control the selection of the best category by a minimum conformity parameter $\mathrm{conf}_{\min} \in [0, 1]$, i.e. the greedy selection algorithm continues when

$$\mathrm{conf}(d_j, \mathrm{descr}(c_{\mathrm{best}})) \geq \mathrm{conf}_{\min}$$

satisfied, where c_{best} is the best category at the given level.

2.2.2 Training

In order to improve the effectiveness of classification, we apply supervised iterative learning, i.e. we check the correctness of the selected categories for training documents and if necessary (a document is classified incorrectly), we modify term weights in category descriptors.

The classifier can commit two kinds of error: it can misclassify a document d_j into c_i, and usually simultaneously, it cannot determine the correct category of d_j. Our weight modifier method is able to cope with both types of error. We scan all the decisions made by the classifier and process as follows.

For each considered category c_i at a given level we accumulate a vector $\delta(c_i) = \langle \delta(v_{1i}), \ldots, \delta(v_{\mathcal{T}i}) \rangle$ where

$$\delta(v_{ki}) = \alpha \cdot (\mathrm{conf}_{\mathrm{req}} - \mathrm{conf}(d_j, \mathrm{descr}(c_i)))) \cdot w_{ij}, \quad 1 \leq k \leq \mathcal{T} \tag{6}$$

where $\mathrm{conf}_{\mathrm{req}} = 1$ when $c_i \in \mathrm{topic}(d_j)$, 0 otherwise. Here $\alpha \geq 0 \in \mathbb{R}$ is the learning rate. The category descriptor weight v_{ki} is updated as $v_{ki} + \delta(v_{ki})$, $1 \leq k \leq \mathcal{T}$, whenever category c_i takes part in an erroneous classification. If d_j is misclassified into c_i then $(\mathrm{conf}_{\mathrm{req}} - \mathrm{conf}(d_j, \mathrm{descr}(c_i)))$ is negative, hence the weight of co-occurring terms in d_j and c_i are reduced in $\mathrm{descr}(c_i)$. In the other case, if c_i is the correct but unselected category of d_j, then $(\mathrm{conf}_{\mathrm{req}} - \mathrm{conf}(d_j, \mathrm{descr}(c_i)))$ is positive, thus the weight of co-occurring terms in d_j and c_i are increased in $\mathrm{descr}(c_i)$

We also experimented with a more sophisticated weight setting method where the previous momentum of the weight modifier is also taken into account in the determination of the current weight modifier. Let $\delta^{(n)}(v_{ki})$ be the weight modifier in the nth training cycle, and $\delta^{(0)}(v_{ki}) = 0$ for all $1 \leq k \leq \mathcal{T}$. Then the weight modifier of the next training cycle is $\delta^{(n+1)}(c_i) = \langle \delta^{(n+1)}(v_{1i}), \ldots, \delta^{(n+1)}(v_{\mathcal{T}i}) \rangle$, and its elements are calculated as

$$\begin{aligned} \delta^{(n+1)}(v_{ki}) = \alpha \cdot (\mathrm{conf}_{\mathrm{req}} - \mathrm{conf}(d_j, \mathrm{descr}(c_i)))) \cdot w_{ij} \\ + \delta^{(n)}(v_{ki}) \cdot \beta \end{aligned} \tag{7}$$

where $\beta \in [0, 1]$ is the momentum coefficient. The value of α and β can be uniform for all categories, or can depend on the level of the category. We experienced that at a lower value, typically 0.05..0.2 is better if the number of training documents is plentiful, i.e. higher in the hierarchy, and a higher value is favorable when only a few training documents are available for the given category, i.e. at leaf categories.

The number of nonzero weights in category descriptors increases as the training algorithm operates. In order to avoid their proliferation, we propose to set descriptor term weights to zero under a certain threshold.

The training cycle is repeated until the given maximal iteration has not been finished or the performance of the classifier reaches a quasi-maximal value. We use the following optimization (or quality) function to measure inter-training effectiveness of the classifier for a document d:

$$\begin{aligned} Q(d) = & \frac{\#(\text{correctly found topics of } d)}{\#(\text{total topics of } d)} \\ & \cdot \frac{1}{1 + \#(\text{incorrectly found topics of } d)}. \end{aligned}$$

The overall Q is calculated as average of $Q(d)$ values:

$$\overline{Q} = \frac{\sum_{d \in \mathcal{D}_{\mathrm{Train}}} Q(d)}{|\mathcal{D}_{\mathrm{Train}}|} \tag{8}$$

The quality measure $\overline{Q}$ is more sensible to small changes in the effectiveness of the classifier than, e.g., F-measure [14] that we use to qualify the final performance of the classifier (see Section 3). Therefore, it is more suitable for inter-training utilization. By setting a maximum variance value $\max_{\text{var}}$ (typically 0.9..1.0) we stop training when actual $\overline{Q}$ drops below the $\max_{\text{var}} \cdot \overline{Q}^{\text{best}}$, where $\overline{Q}^{\text{best}}$ is the best $\overline{Q}$ achieved so far during training.

3 Experimental results

3.1 Dimensionality reduction

When dealing with large document collection, the large number of terms, $|\mathcal{T}|$, can cause problem in document processing, indexing, and also in category induction. Therefore, before indexing and category induction many authors apply a pass of *dimensionality reduction* (DR) to reduce the size of $|\mathcal{T}|$ to $|\mathcal{T}'| \ll |\mathcal{T}|$ [4]. Beside that it can speed up the categorization, papers also reported that it can increase the performance of the classifier with a few percent, if only a certain subset of terms are used to represent documents (see e.g. [15, 16]). In our previous experiments [17] we also found that performance can be increased slightly (less than 1%) if rare terms are disregarded, but the effect of DR on time efficiency is more significant. We applied DR by removing the least frequent terms in the overall collection. In general, we reduced $|\mathcal{T}|$ by disregarding terms that either occur less than $\min_{\text{occur}}$ times in the entire train document set, or occur more often than a certain threshold in the training set, i.e. if $n_k/|\mathcal{D}_{\text{Train}}| \geq \max_{\text{freq}}$. By the former process we disregard words that are not significant in the classification, while by the later process we ignore words that are not discriminative enough between categories. The typical values are $\min_{\text{occur}} \in [1\,..\,10]$ and $\max_{\text{freq}} \in [0.05\,..\,1.0]$.

3.2 Performance measures

Beside the usual recall, precision, and F-measure, we have adopted two heuristic evaluation measures that were proposed in [18] and we are proposing another one which focuses on recall. Let us suppose that algorithm returns an ordered list of guesses, where the order is determined by the confidence level used as weight. Then we can define the following measures (see Figure 2):

1. **Top prediction** (briefly: Top) The top category predicted by the classifier is compared to the main category of the document, shown as [mc] in Figure 2.

2. **Three guesses** (Top 3) The top three categories predicted by the classifier are compared to main category of the document. If a single match is found, the categorization is deemed successful. This measure is adapted to evaluating categorization assistance, where a user ultimately makes the

decision. In this case, it is tolerable that the correct guess appears second or third in the list of suggestions.

3. **Five Guesses** (5G) We compare the top 5 prediction of the classifier with all categories associated with the document. The number of correct guesses are counted and normalized for each documents and the final results are cumulated over the whole collection.

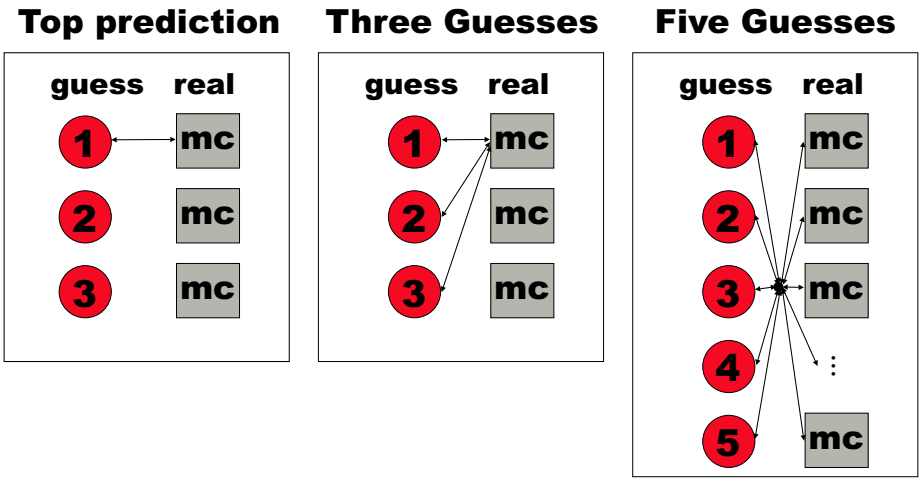

Figure 2: Explanation to the three evaluation measures Top, Top 3 [18], 5G

We use these measures at each level of the taxonomy.

3.3 Document collection

We obtained the bi-lingual document corpus used for our experiments as a courtesy of ILO, however it is unfortunately not a public collection at the moment. It contains about 36000 documents in a category tree of 18000 elements in 10 levels. Documents are short texts in a very wide range of topics. The original taxonomy is available in three languages: English, French, Spanish, but the collection contains documents in only the first two. The category tree is not balanced, i.e. some part of the taxonomy is deeper than others. The number of document belonging to categories at various levels are (top–down): 0, 717, 46869, 92072, 96249, 45554, 12548, 1742, 71, 31. (Each document is counted as many times as many categories it is assigned to). Most documents are attached to several categories, usually to 5 or 6. The experienced results for the English and French corpora are displayed in Tables 1 and 2 for two cases, when 16% and 33% of the total collection is used for test, respectively. We differentiated results based on the confidence level. Here 0.0 means that all guesses are considered, while 0.8 means that only those decision are considered where the confidence

level is not less than 0.8. Obviously, the higher is the confidence level, the lower is the number of considered documents.

We selected the 0.2 confidence level to show the results, because, on the one hand, at lower levels the number of predicted categories are much higher than the original number of categories, hence it is unrealistic to compare them at these levels; on the other hand, at higher levels the number of predicted category is much less than the number of original categories, so we face again with the same problem.

Table 1: Summary of results on the English ILO corpus

Depth/	test	perf. measure		
conf.level	set	Top	Top3	5G
1/0.2	16%	95.46	99.56	84.97
	33%	95.06	95.54	84.96
3/0.2	16%	83.20	94.85	65.53
	33%	82.99	94.88	65.71
5/0.2	16%	58.93	67.14	47.37
	33%	57.08	65.39	46.37
7/0.2	16%	46.61	50.60	44.26
	33%	48.55	51.04	45.86

Table 2: Summary of results on the French ILO corpus

Depth/	test	perf. measure		
conf.level	set	Top	Top3	Any
1/0.2	16%	95.42	99.55	85.46
	33%	95.09	99.51	85.42
3/0.2	16%	83.89	95.26	66.28
	33%	83.39	94.88	65.97
5/0.2	16%	58.23	66.86	47.58
	33%	56.85	65.54	46.85
7/0.2	16%	46.27	48.23	42.20
	33%	42.86	44.95	38.95

One can observe that at most selected taxonomy level the results are very similar for both test settings and languages. Surprisingly, the quality of categorization does not decrease significantly when the test collection was doubled on the account of the training collection, moreover, in some case, see e.g. the English corpus at 7/0.2 level the results are actually improved. At higher taxonomy there are no significant differences between the results for the two languages, but the quality of categorization decreases slower for the English corpus than for the French with going down in the taxonomy.

As hinted in [9] there are two typical practical setups for cross-lingual text categorization:

1. poly-lingual training: simultaneous training on labelled documents in two languages (A and B) allows to classify both A and B documents with the same classifier.

2. cross-lingual training: a monolingual trained classifier for language A plus a translation of the most important terms from language A to B allows to classify documents written in B.

The results just presented will be a good baseline for testing these hypotheses with our text classifier algorithm.

4 Acknowledgement

This work was funded by FKFP 0180/2001.

5 Conclusion

In this paper we showed the effectiveness of our text categorization algorithm on a bi-lingual document corpus. The accuracy of the results allows us to apply cross-lingual text categorization on this basis, which task is to be completed in the near future.

References

[1] K. D. Baker and A. K. McCallum, "Distributional clustering of words for text classification," in *Proc. of the 21th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, Melbourne, Australia, 1998, pp. 96–103.

[2] S. M. Weiss, C. Apte, F. J. Damerau, D. E. Johnson, F. J. Oles, T. Goetz, and T. Hampp, "Maximizing text-mining performance," *IEEE Intelligent Systems*, vol. 14, no. 4, pp. 2–8, July/August 1999.

[3] E. Wiener, J. O. Pedersen, and A. S. Weigend, "A neural network approach to topic spotting," in *Proc. of the 4th Annual Symposium on Document Analysis and Information Retrieval*, 1993, pp. 22–34.

[4] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, March 2002.

[5] L. Aas and L. Eikvil, "Text categorisation: A survey," Norwegian Computing Center, Raport NR 941, 1999.

[6] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan, "Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies," *The VLDB Journal*, vol. 7, no. 3, pp. 163–178, 1998.

[7] D. Tikk, G. Biró, L. Kovács, and J. D. Yang, "Hierarchical text categorization on the Reuters collection," in *Proc. of 1st Serbian–Hungarian Joint Symposium on Intelligent Systems (SISY03)*, Subotica, Serbia-Montenegro, September 19–20, 2003, pp. 81–85.

[8] D. Tikk and G. Biró, "Experiment with a hierarchical text categorization method on the WIPO patent collection," in *Proc. of the 4th International Symposium on Uncertainty Modeling and Analysis (ISUMA 2003)*, University of Maryland, USA, 2003.

[9] N. Bel, C. H. A. Koster, and M. Villegas, "Cross-lingual text categorization," in *Proceedings ECDL 2003*, Trondheim, August 2003, pp. 126–139, http://www.cs.kun.nl/peking/ecdl03.ps.gz.

[10] C. Apte, F. J. Damerau, and S. M. Weiss, "Automated learning of decision rules for text categorization," *ACM Trans. Information Systems*, vol. 12, no. 3, pp. 233–251, July 1994.

[11] S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in *Proc. of 7th ACM Int. Conf. on Information and Knowledge Management (CIKM-98)*, Bethesda, MD, 1998, pp. 148–155.

[12] G. Salton and M. J. McGill, *An Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[13] A. McCallum, R. Rosenfeld, T. Mitchell, and A. Ng, "Improving text classification by shrinkage in a hierarchy of classes," in *Proc. of ICML-98*, 1998, `http://www-2.cs.cmu.edu/~mccallum/papers/hier-icml98.ps.gz`.

[14] C. J. van Rijsbergen, *Information Retrieval*, 2nd ed. London: Butterworths, 1979, `http://www.dcs.gla.ac.uk/Keith`.

[15] D. Koller and M. Sahami, "Hierarchically classifying documents using a very few words," in *International Conference on Machine Learning*, vol. 14. San Mateo, CA: Morgan-Kaufmann, 1997.

[16] W. Wibovo and H. E. Williams, "Simple and accurate feature selection for hierarchical categorisation," in *Proc. of the 2002 ACM symposium on Document engineering*, McLean, Virginia, USA, 2002, pp. 111–118.

[17] D. Tikk, J. D. Yang, and S. L. Bang, "Hierarchical text categorization using fuzzy relational thesaurus," To appear in *Kybernetika*.

[18] C. J. Fall, A. Törcsvári, and G. Karetka, "Readme information for WIPO-alpha autocategorization training set," December 2002, `http://www.wipo.int/ibis/datasets/wipo-alpha-readme.html`.