KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 25–31

TEXT CATEGORIZATION EXPERIMENTS USING WIKIPEDIA

ZALÁN BODÓ AND ZSOLT MINIER AND LEHEL CSATÓ$^{(1)}$

Abstract. Over the years many models had been proposed for text categorization. One of the most widely applied is the vector space model, assuming independence between indexing terms. Since training corpora sizes are relatively small – compared to ∞ – the generalization power of the learning algorithms is relatively low. Using a bigger unannotated text corpus can boost the representation and hence the learning process. Based on the work of Gabrilovich and Markovitch we use Wikipedia articles to give word distributional representation for documents. Since this causes dimensionality increase, some feature clustering is needed. For this end we use LSA.

1. Introduction

Text categorization is one of the more profoundly examined task of information retrieval. The amount of textual information available these days makes more and more necessary the intelligent and *efficient* methods that help navigating in this virtual space.

It is a "classical" categorization or classification task: given a function $f : D \to 2^C$ (by training examples), where D is the set of documents and C is the set of categories, find $\widehat{f}$, which best approximates f. The training examples $(\mathbf{d}_i, y_i)$, $i = 1, \ldots, |D|$, compose the training corpus, where $\mathbf{d}_i$ and y_i denotes the document and the associated label respectively. The problem is evidently a multi-class and multi-label task, since usually there are more than two classes, and a document can belong to many classes.

The solution of the problem is usually separated into two phases: term selection and machine learning. Both of these two are important parts of a text categorization system. A good term selection is very useful, because usually the data contain noise, irrelevant features can lead the system into wrong direction, etc. Some machine learning techniques like SVMs and Rocchio's method can also filter out irrelevant terms.

2000 *Mathematics Subject Classification.* 68P20, 68T30.

Key words and phrases. text categorization, document representation, feature selection, latent semantic analysis.

The most widely used model in information retrieval and hence in text categorization is the vector space model (VSM) introduced in [9]. The index terms are words or stems taken from the training corpus, which constitute the basis of the vector space, therefore they are assumed to be independent – although words are evidently not semantically independent. Because of its simplicity and good performance it is the most popular one used in the IR community, since other, more sophisticated models do not provide significantly better performance. Meaningful, descriptive terms should get a higher weight in the representation, so term weighting is also an important factor of the system. The term frequency×inverse document frequency (tfidf) is defined as

$$w_{ij} = \mathrm{tr}(i,j) \cdot \log \frac{|D|}{n_j}$$

which gives higher weights for more descriptive (frequent) terms in a document (tf), and also higher weights for terms which have more discriminative power (idf). Here w_{ij} denotes the weight of word j in document i, $freq(i,j)$ is the frequency of word j in the ith document, $|D|$ is the total number of documents and n_j is the number of documents in which word j appears through the corpus.

For a good comparative study on the basic feature selection techniques in text categorization see [11], while [10] gives a broad overview on different machine learning techniques applied to the problem.

In this paper we study the use of Wikipedia derived knowledge in enhancing text categorization. The next section constitutes the main part of the article describing the steps of the proposed method. We describe the experiments in section 3 and discuss the results in section 4.

2. Wikipedia-based text categorization

2.1. **Wikipedia.** The Wikipedia is the largest encyclopedia edited collaboratively on the internet comprising of ≈ 1.6 million concepts totalling ≈ 8 gigabytes of textual data. It is written in a clear and coherent style with many concepts explained sufficiently deeply thus making it a wonderful resource for natural language research. Our need for a semantic relatedness metric between words could be easily approached using the distribution of words in the different Wikipedia concepts.

2.2. **Document representation.** Document representations can be very sparse in the vector space model, because they are indexed by a few word taken from the training corpus.

Through the article any vector is considered a row vector.

Gabrilovich and Markovitch [6] use word distributional representation for measuring word and text *relatedness*, using Wikipedia articles. We adopt their technique to represent documents in this concept space.

Given a document $\mathbf{d}$ containing $|W|$ terms it can be transformed to the Wikipedia concept space by

$$\underbrace{\begin{pmatrix} w_1 & w_2 & \dots & w_{|W|} \end{pmatrix}}_{\mathbf{d}} \cdot \underbrace{\begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1|C|} \\ c_{21} & c_{22} & \cdots & c_{2|C|} \\ \vdots & \vdots & \ddots & \vdots \\ c_{|W|1} & c_{|W|2} & \cdots & c_{|W||C|} \end{pmatrix}}_{\mathbf{W}}$$

where the w's are the weights of the words (e.g. tfidf, idf calculated upon the categorization corpus or Wikipedia) and c_{ij} represents the weight of the word i in Wikipedia concept j. It is easy to observe that the matrix $\mathbf{W}$ is a term×concept (document) matrix, which transforms the document to the concept space, therefore by calculating the *similarity* of two text gives us the well-known GVSM-kernel [12], [3]

$$K_{\text{GVSM}}(\mathbf{d_1}, \mathbf{d_2}) = \mathbf{d_1 W}(\mathbf{d_2 W})^{\mathrm{T}} = \mathbf{d_1 W W}^{\mathrm{T}} \mathbf{d_2}^{\mathrm{T}}$$

where $\mathbf{WW}^{\mathrm{T}}$ is a term×term correlation matrix, now built upon external information.

We have used the same representation for documents, that is we transformed documents from the term space to the concept space by multiplying the document×term matrix ($\mathbf{X}$) by $\mathbf{W}$. Although both $\mathbf{X}$ and $\mathbf{W}$ are sparse matrices (with a density of 0.67% and 0.2%, respectively), their product results in a much more denser structure. Though – similarly to [6] – we filtered out Wikipedia articles considered less important, the resulting matrix is still huge and dense, thus making difficult to store and actually use the constructed document vectors.

2.3. **Dimensionality reduction.** To address the high dimensionality of vector spaces in natural language processing, dimensionality reduction techniques are often used. Dimensionality reduction helps at making feature vectors small enough to be handled by machine learning methods and many times has the beneficial effect of removing noise and thus slightly increasing classification accuracy and reducing overfitting.

2.3.1. *Latent Semantic Analysis.* Latent Semantic Analysis was introduced by Deerwester et al. [4] in information retrieval. It is a dimensionality reduction technique based on singular value decomposition. Given the term×document matrix $\mathbf{A}$, it is decomposed as $\mathbf{A} = \mathbf{USV}^{\mathrm{T}}$ where $\mathbf{U}$ and $\mathbf{V}$ are matrices of orthonormal columns and $\mathbf{S}$ is diagonal. By controlling the number of singular values in $\mathbf{S}$ one can achieve a dimensionality reduction in both the term×term matrix $\mathbf{U}$ and in the document×document matrix $\mathbf{V}$.

We used this method to reduce the dimensionality of the word×concept matrix that we built from Wikipedia.

2.4. **Learning over training data.** For learning the decision boundaries we applied support vector machines using the LIBSVM [1] library with linear kernels. Support vector machines (SVMs) were introduced by Vapnik for binary classification. Although the formulation of the problem can be extended in some ways to handle multi-class classification, in most cases – because of the lower computational cost – binarization techniques like one-vs-rest, one-vs-one, error correcting output coding, etc. are used for supporting non-binary classification.

In its simplest form the SVM maximizes the margin $(2/\|\mathbf{w}\|)$ of the hyperplane which separates positive and negative examples, that is under the following condition:

$$y_i \cdot (\mathbf{w}^{\mathrm{T}}\mathbf{x} + b) \geq 1, \quad \forall \mathbf{x}_i \in P \cup N$$

where P and N denotes the set of positive and negative samples respectively. The decision function is simply

$$f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b)$$

The separating hyperplane of maximal margin depends only on the vectors supporting the marginal hyperplanes, and this is the reason why is called support vector machine.

Joachims [7] experimentally proved that the classes induced by the documents from the Reuters corpus are more or less linearly separable using the vector space model. We assumed the same holds for our model.

3. Experimental methodology and results

The first step constituted building an inverted index for the words appearing in Wikipedia, excluding stop words and also the first 300 most frequent words. Because the large amount of Wikipedia articles turns the problem into one of unmanageable size, as we have mentioned earlier, we filtered out some of them, namely we eliminated those containing less than 500 words or having less or equal than 5 forward links to other Wikipedia articles. In this way we processed 327 652 articles.

For testing the model we used the Reuters-21578 text categorization corpus, ModApté split with $90 + 1$ categories. The Reuters collection contains documents which appeared on the Reuters newswire in 1987. These documents were manually categorized by the personnel from Reuters Ltd. and Carnegie Group Inc. in 1987. The collection was made available for scientific research in 1990. Originally, there were 21 578 documents, but some of them, namely 8681 were unused in the split, moreover, some of it were not categorized – they were put in the unknown category. Removing this "virtual" class, the training and the test corpus contains 9583 and 3744 documents respectively, defining 90 classes. Some of the documents are assigned to more than one category, the average number of classes per document being 1.24 [2].

In the preprocessing step we selected the top 5209 word stems of the Reuters corpus using the χ^2 term ranking method [11]. For these 5209 terms the inverted index was built based on the Wikipedia, that is, each term is represented as a vector of occurences in the vector space of Wikipedia concepts. The number of dimensions of this vector space is 327 652.

With this data we performed four experiments on the Reuters corpus.

In the first experiment (χ^2[5209]) we measured the performance of the system with the terms extracted from the corpus itself. We used these results as a baseline for another term selection method [8] and we used them for baseline here as well. In [8] we proposed a term selection method based on segmenting the textual data for categorization, then clustering these text segments in each category to obtain the largest clusters and using the terms (stems) from the merged clusters as features. The results obtained for the Reuters corpus were similar to the performance of the χ^2 term ranking method, however in our method there is no need to determine the optimal number of features. Therefore we used χ^2 term selection with our feature threshold.

In the second experiment (χ^2[5209]+LSA) we expressed the documents of the Reuters corpus with Wikipedia concepts through transforming the words in each document into Wikipedia concept space. However, as the dimensionality of the Wikipedia concept space is prohibitively large, we reduced it using the LSA method to an arbitrarily chosen 2000 number of dimensions. Effectively, in the transformation $\widehat{\mathbf{X}} = \mathbf{XW}$ where $\mathbf{X}$ is the data of the corpus (document×word) and $\mathbf{W}$ is the Wikipedia matrix (word×concept) we replace $\mathbf{W}$ with $\mathbf{U}$ from the singular value decomposition of $\mathbf{W} \approx \mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^{\mathrm{T}}$ keeping only the first $k = 2000$ columns in $\mathbf{U}$. We did not actually perform this singular value decomposition, because it is not needed to find $\mathbf{U}$ which is actually the matrix containing the eigenvectors of $\mathbf{WW}^{\mathrm{T}}$ which can be found by principal component analysis. This way the final number of dimensions of each document in the corpus is 2000 and the document vectors are dense.

In the third experiment (χ^2[2000]) we selected only the first 2000 top ranking word stems from the corpus as terms using the χ^2 method to compare our dimensionality reduction based on semantic relatedness of words to term ranking. The documents are represented as sparse vectors with 2000 dimensions.

In the fourth experiment (χ^2[5209]+noLSA) we did not perform a principal component analysis of $\mathbf{WW}^{\mathrm{T}}$ so we transformed the corpus with $\mathbf{WW}^{\mathrm{T}}$ to use all the semantic relatedness between words obtained from Wikipedia. Document vectors became dense with their original dimension of 5209.

The results we obtained are shown on figure 1.

The performance is measured using the common precision-recall breakeven point – the intersection of the precision and recall curves if such a point exists – and the F_1 measure.

	mP	mR	mBEP	mF1	MP	MR	MBEP	MF1
χ^2[5209]	88.46	84.59	86.52	86.48	71.61	61.25	66.43	66.02
χ^2[5209]+LSA	86.68	82.59	84.63	84.58	62.97	53.61	58.29	57.91
χ^2[2000]	87.34	84.21	85.77	85.75	64.76	58.14	61.45	61.27
χ^2[5209]+noLSA	48.95	35.42	42.18	41.10	8.79	5.35	7.07	6.65

Figure 1. Performance results obtained for the Reuters corpus given in percentage. Notation: mP=micro-precision, mR=micro-recall, mBEP=micro-breakeven, mF1=micro-F_1, MP=macro-precision, MR=macro-recall, MBEP=macro-breakeven, MF1=macro-F_1

4. Discussion

As the results show, using Wikipedia in this way did not help classification performance. We also tried to use nonlinear kernels as inhomogeneous polynomial and RBF with optimized parameters by cross validation, but none of them produced a significant improvement, so we did not insert these results in the paper. Including the semantic relatedness or cooccurrence information in the document vectors, the performance drops, meaning that it only shows up as noise in the data. It appears that Wikipedia-based semantic relatedness does not model well the similarity between the documents from the same Reuters class. However, Gabrilovich and Markovich [5] showed that by augmenting the features of the bag-of-words model with closely related Wikipedia concepts results in significantly better performance.

Using the Wikipedia-based representation for word similarity Gabrilovich and Markovich [6] obtained a much more higher correllation with human judgement as for the cosine similarity. This means that the document representation as a weighted sum of the contained words' vectors is inappropriate, otherwise the kernel $\mathbf{X}\mathbf{U}_k\mathbf{U}_k^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}$ would improve categorization performance.

One possible improvement could be to cut off extremities from the transformation matrix to increase the sparseness of document representations and decrease possible noise.

The method should be tested on other corpora, to verify that semantic relatedness – derived from Wikipedia – can help categorization. The Reuters collection is very unbalanced, causing serious difficulties for text categorization systems.

Our experiments show that still the bag-of-words model performs better than the semantic space model of documents.

5. Acknowledgements

We would like to acknowledge the financial support of the grant CEEX/1474 by the Romanian Ministry of Education and special thanks to the anonymous reviewers of our paper for the supporting critics.

References

[1] Chih-chung Chang and Chih-jen Lin. LIBSVM: a library for support vector machines (version 2.31), September 07 2001.

[2] Koby Crammer and Yoram Singer. A new family of online algorithms for category ranking. In *SIGIR*, pages 151–158. ACM, 2002.

[3] Nello Cristianini, John Shawe-Taylor, and Huma Lodhi. Latent semantic kernels. *J. Intell. Inf. Syst*, 18(2-3):127–152, 2002.

[4] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41, June 1990.

[5] Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*. AAAI Press, 2006.

[6] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of The 20th International Joint Conference on Artificial Intelligence (IJCAI)*, January 2007.

[7] Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In Claire Nedellec and Céline Rouveirol, editors, *ECML*, volume 1398 of *Lecture Notes in Computer Science*, pages 137–142. Springer, 1998.

[8] Zsolt Minier, Zalán Bodó, and Lehel Csató. Segmentation-based feature selection for text categorization. In *Proceedings of the 2nd International Conference on Intelligent Computer Communication and Processing*, pages 53–59. IEEE, September 2006.

[9] G. Salton, A. Wong, and A. C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18:229–237, 1975.

[10] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[11] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of ICML-97, 14th International Conference on Machine Learning*, 1997.

[12] Yiming Yang, Jaime G. Carbonell, Ralf D. Brown, and Robert E. Frederking. Translingual information retrieval: Learning from bilingual corpora. *Artificial Intelligence*, 103(1–2):323–345, 1998.

(1) Department of Mathematics and Computer Science, Babeş-Bolyai University, RO-400084, Cluj-Napoca, Romania

E-mail address: {zbodo, minier, lehel.csato}@cs.ubbcluj.ro