Multi-Dimensional Text Classification

Thanaruk THEERAMUNKONG
IT Program, SIIT, Thammasat University
P.O. Box 22 Thammasat Rangsit Post Office,
Pathumthani, Thailand, 12121
ping@siit.tu.ac.th

Verayuth LERTNATTEE
IT Program, SIIT, Thammasat University
P.O. Box 22 Thammasat Rangsit Post Office,
Pathumthani, Thailand, 12121
verayuth@siit.tu.ac.th

Abstract

This paper proposes a multi-dimensional framework for classifying text documents. In this framework, the concept of multi-dimensional category model is introduced for representing classes. In contrast with traditional flat and hierarchical category models, the multi-dimensional category model classifies each text document in a collection using multiple predefined sets of categories, where each set corresponds to a dimension. Since a multi-dimensional model can be converted to flat and hierarchical models, three classification strategies are possible, i.e., classifying directly based on the multi-dimensional model and classifying with the equivalent flat or hierarchical models. The efficiency of these three classifications is investigated on two data sets. Using *k*-NN, naïve Bayes and centroid-based classifiers, the experimental results show that the multi-dimensional-based and hierarchical-based classification performs better than the flat-based classifications.

1 Introduction

In the past, most of previous works on text classification focus on classifying text documents into a set of flat categories. The task is to classify documents into a predefined set of categories (or classes) (Lewis and Ringuetee, 1994; Eui-Hong and Karypis, 2000) where there are no structural relationships among these categories. Many existing databases are organized in this type of flat structure, such as Reuters newswire, OHSUMED and TREC. To improve classification accuracy, a variety of learning techniques are developed, including regression models (Yang and Chute, 1992), nearest neighbour classification (Yang and Liu, 1999), Bayesian approaches (Lewis and Ringuetee, 1994; McCallum et al., 1998), decision trees (Lewis and Ringuetee 1994), neural networks (Wiener et al.,1995) and support vector machines (Dumais and Chen, 2000). However, it is very difficult to browse or search documents in flat categories when there are a large number of categories. As a more efficient method, one possible natural extension to flat categories is to arrange documents in topic hierarchy instead of a simple flat structure. When people organize extensive data sets into fine-grained classes, topic hierarchy is often employed to make the large collection of classes (categories) more manageable. This structure is known as category hierarchy. Many popular search engines and text databases apply this structure, such as Yahoo, Google Directory, Netscape search and MEDLINE. There are many recent works attempting to automate text classification based on this category hierarchy (McCallum et al., 1998; Chuang W. T. et al., 2000). However, with a large number of classes or a large hierarchy, the problem of sparse training data per class at the lower levels in the hierarchy raises and results in decreasing classification accuracy of lower classes. As another problem, the traditional category hierarchy may be too rigid for us to construct since there exist several possible category hierarchies for a data set.

To cope with these problems, this paper proposes a new framework, called multi-dimensional framework, for text classification. The framework allows multiple pre-defined sets of categories (viewed as multiple dimensions) instead of a single set of categories like flat categories. While each set of classes with some training examples (documents) attached to each class, represents a criterion to classify a new text document based on such examples, multiple sets of classes enable several criteria. Documents are classified based on these multiple criteria (dimensions) and assigned a class per criterion (dimension). Two merits in the multi-dimensional approach are (1) the support of multiple viewpoints of classification, (2) a solution to data sparseness problem. The efficiency of multi-dimensional classification is investigated using three classifiers: *k*-NN, naïve Bayes and centroid-based methods.

2 Multi-Dimensional Category Model for Text Classification

Category is a powerful tool to manage a large number of text documents. By grouping text documents into a set of categories, it is possible for us to efficiently keep or search for information we need. At this point, the structure of categories, called *category model*, becomes one of the most important factors that determine the efficiency of organizing text documents. In the past, two traditional category models, called flat and hierarchical category models, were applied in organizing text documents. However, these models have a number of disadvantages as follows. For the flat category model, when the number of categories becomes larger, it faces with difficulty of browsing or searching the categories. For the hierarchical category model, constructing a good hierarchy is a complicated task. In many cases, it is not intuitive to determine the upward/downward relations among categories. There are several possible hierarchies for the same document set. Since hierarchies in the hierarchical category model are static, browsing and searching documents along the hierarchy are always done in a fix order, from the root to a leaf node. Therefore, the searching flexibility is lost.

As an alternative to flat and hierarchical category models, the multi-dimensional category is introduced. So far the concept of multi-dimensional data model has been very well known in the field of database technology. The model was shown to be powerful in modeling a data warehouse or OLAP to allow users to store, view and utilize relational data efficiently (Jiawei and Micheline, 2001). This section describes a way to apply multi-dimensional data model to text classification, so called multi-dimensional category. The proposed model is an extension of flat category model, where documents are not classified into a single set of categories, instead they are classified into multiple sets. Each set of categories can be viewed as a dimension in the sense that documents may be classified into different kinds of categories. For example in Figure 1, a set of news issues (documents) can be classified into three dimensions, say TOPIC, ZONE and TOPIC, each including {sports, economics, politics, social, entertainment, science and technology}, {domestic, intra-continental, inter-continental} and {good news, bad news, neutral news}, respectively. A news issue in a Thailand newspaper titled "Airplanes attacked World Trader Center" can be classified into "social news", "inter-continental", "bad news" in the first, second and third dimensions, respectively.

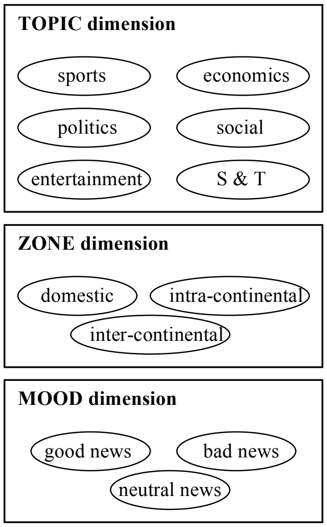

Figure 1. Three-dimension category model for classifying news documents

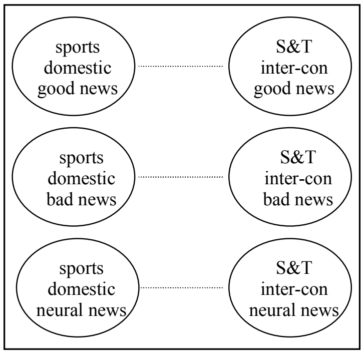

Figure 2. Flat category model for the model in Figure 1

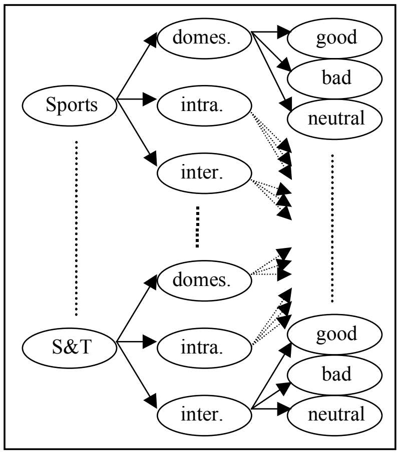

Figure 3. Hierarchical category model for the model in Figure 1

Comparing with flat and/or hierarchical category models, the multi-dimensional model has the following merits. First, it is more natural than flat model in the sense that a document could be classified basing on not a single criterion (one dimension) but multiple criteria (multiple dimensions). Secondly, in contrast with hierarchical model, it is possible for us to browse or search documents flexibly without the order constraint defined in the structure. Lastly, the multi-dimensional category model can be basically transformed to and represented by flat category or hierarchical category models, even the converses are not always intuitive.

In the previous example, the corresponding flat and hierarchical models for the multi-dimensional model in Figure 1 are illustrated Figure 2 and 3, respectively. The total number of derived flat categories equals to the product of the number of categories in each dimension, i.e., 54(=6x3x3). In the derived hierarchical model, the number of leaf categories is also equivalent to 54 but there exist 24 (=6+6x3) internal categories. Note that the figure shows only one possible hierarchy where the dimensions ordered by TOPIC, ZONE and MOOD. However, there are totally 6 (=3!) possible hierarchies for the model in Figure 1.

From a viewpoint of category representation, the fact that the derived flat model enumerates all combinations among categories, makes the representation of a class be more precise than the class in multi-dimensional model. However, from the viewpoint of relationship constraints in these models, the derived flat category model ignores the relationship among categories while the derived hierarchical model explicitly declares such relationship in a rigid manner, and the multi-dimensional model is a compromise between these two previous models. These different aspects affect classification efficiency as shown in the next section.

3 Multi-Dimensional Classification

Described in the previous section, a multi-dimensional category model can be transformed into flat and hierarchical category models. As a result, there are three different classification strategies: flat-based, hierarchical-based and multi-dimensional-based methods.

3.1 Flat-based classification

The naïve method to classify documents according to a multi-dimensional model is flat-based classification. After transforming a multi-dimensional category model to flat category model, traditional flat classification is applied directly to the derived flat categories. The granularity of the derived flat categories is finer than the original multi-dimensional categories since all combinations of classes in the dimensions are enumerated. This fact implies that a flat category represents the class more precisely than a multi-dimensional category and then one can expect high classification accuracy. However, on the other hand, the number of training data (documents) per class is reduced. As a consequence, flat classification may face with the sparseness problem of training data. This may cause a classifier harder to classify and then reduce classification accuracy. In the view of computational cost, a test document has to be compare to all enumerated classes, resulting in high computation.

3.2 Hierarchical-based classification

The second method is to transform a multi-dimensional category model to a hierarchical category model and then apply the standard hierarchical classification on the derived hierarchical model. There are several possible models generated from a multi-dimensional model due to the order of dimensions as described in section 2. The classification is held along the hierarchy from the root to a leaf. The decision of the class, which a document belongs to, is made in step by step. The classifications of different levels occupy different granularities of training data. Nodes at the level closed to the root will have coarser granularity. This makes such nodes represent classes less imprecisely but there are more training data (documents) for these nodes. On the other hand, nodes near leaves will have finer granularity and then have more precise representation but have less training data. The classification accuracy varied with the order of dimensions in the hierarchy.

3.3 Multi-dimensional-based classification

It is possible to directly classify a document using the multi-dimensional category model. The class of the document for each dimension is determined independently. We called this multi-dimensional-based classification. Compared with flat-based classification, the granularity of multi-dimensional classification is coarser. For each dimension, it classifies a document based on categories in that dimension instead of classifying it into the set of finer categories as done in flat classification. Although the multi-dimensional category is not precisely represent any finer categories, the number of training data (documents) per class is relatively high. As a consequence, multi-dimensional classification gains high accuracy for each dimension and results in high accuracy for the overall classification accuracy when there are a small number of training data. It also performs faster than flat-based classification since there are fewer classes needed to be compared.

4 Implementation

To investigate efficiency of text classification on the multidimensional category model, three well-known classification algorithms called *k*-nearest neighbors (*k*-NN), naïve Bayesian (NB) and centroid-based (CB) approaches are applied.

4.1 *k*-NN Classifier

As a similarity-based method, the *k*-nearest neighbor classifier (*k*-NN) is applied to our text classification. First, the classifier calculates *k* most similar documents (i.e., *k* nearest neighbors) of the test document being classified. The similarity of this document to a class is computed by summing up the similarities of documents among the *k* documents, whose classes are equivalent to such class. The test document is assigned the class that has the highest similarity to the document. Two parameters involved are the definition of similarity and the number *k*. While the standard similarity is defined as *tf×idf*, a variant $(0.5+0.5tf/tf_{max})\times idf$ that performed better in our preliminary experiments, is applied in this work. The parameter *k* is determined by experiments as shown in the next section.

4.2 Naïve Bayes Classifier

The standard naïve Bayesian (NB) is applied as a statistical approach to our text classification in this work. For each document, the classifier first calculates the posterior probability $P(c_i|d)$ of class c_i that the document belongs to different classes and assigns it to the class with the highest posterior probability. Basically, a document d can be represented by a bag of words $\{w_1, w_2, \ldots, w_n\}$ in that document (i.e., a vector of occurrence frequencies of words in the document). NB assumes that the effect of a word's occurrence on a given class is independent of other words' occurrence. With this assumption, a NB classifier finds the most probable class $c_i \in C$, called a maximum a posteriori (MAP) c_{MAP} for the document, where $C=\{c_1, c_2, \ldots, c_k\}$ is a set of predefined classes.

$$\begin{aligned} c_{MAP} &\equiv \arg\max_{c_i} \frac{P(\{w_1, w_2, \ldots, w_n\} \mid c_i)P(c_i)}{P(\{w_1, w_2, \ldots, w_n\})} \qquad (1) \\ &\equiv \arg\max_{c_i} \prod_{j=1}^{n} P(w_j \mid c_i)P(c_i) \end{aligned}$$

4.3 Centroid-based Classifier

Applied in our implementation is a variant of centroid-based classification (CB) with different weight methods from the standard weighting *tf-idf*. A centroid-based classifier (CB) is a modified version of *k*-NN classifier. Instead of comparing the test document with all training documents, CB calculates a centroid (a vector) for all training documents in each class and compares the test document with these centroids to find the most probable (similar) class. A simple centroid-based classifier represents a document with a vector each dimension of which expresses a term in the document with a weight of *tf×idf*. The resultant vector is normalized with the document length to a unit-length vector. A different version of a centroid vector is so-called a prototype vector (Chuang, W. T. et al., 2000). Instead of normalizing each vector in the class before calculating a centroid, the prototype vector is calculated by normalizing the summation of all vectors of documents in the class. Both methods utilizing centroid-based and prototype vectors obtained high classification accuracy with small time complexity. In our implementation, we use a variant of the prototype vector that does not apply the standard *tf-idf* but use either of the following weighting formulas. These weighting formulas, we called CB1 and CB2, were empirically proved to work well in (Theeramunkong and Lertnattee, 2001).

$$\frac{tf \times idf \times icsd}{\sqrt{tf_{rms}} \times sd}(\mathrm{CB1}) \quad or \quad \frac{tf \times idf}{\sqrt{tf_{rms}} \times sd}(\mathrm{CB2}) \qquad (2)$$

icsd stands for inter-class standard deviation, tf_{rms} is the root mean square of document term frequency in a class, and *sd* means standard deviation. After this weighting, a prototype vector is constructed for each class. Due to the length limitation of the paper, we ignore the detail of this formula but the full description can be found in (Theeramunkong and Lertnattee, 2001).

5 Experimental Results

Two data sets, WebKB and Drug information collection (DI) are used for evaluating our multi-dimensional model. These two data sets can be viewed as a two-dimensional category model as follows. Composed of 8,145 web pages, the WebKB data set is a collection of web pages of computer science departments in four universities with some additional pages from other universities. The original collection is divided into seven classes (1st dimension): student, faculty, staff, course, project, department and others. Focusing on each class, five subclasses (2nd dimension) are defined according to the university a web page belongs to: Cornell, Texas, Washington, Wisconsin and miscellaneous. In our experiment, we use the four most popular classes: student, faculty, course and project. This includes 4,199 web pages. Drug information, the second data set, is a collection of web documents that have been collected from www.rxlist.com. This collection is composed of 4,480 web pages providing information about widely used drugs in seven topics (1st dimension): adverse drug reaction, clinical pharmacology, description, indications, overdose, patient information, and warning. There exists exactly one page for each drug in each class, i.e., the number of recorded drugs is 640 (=4480/7). Moreover We manually grouped the drugs according to major pharmacological actions, resulting in five classes (2nd dimension): chemotherapy (Chem), neuro-muscular system (NMS), cardiovascular & hematopoeitic (CVS), hormone (Horm) and respiratory system (Resp).

The multi-dimensional classification is tested using four algorithms: *k*-NN, NB and two centroid-based classifiers (CB1 and CB2). In the *k*-NN, the parameter *k* is set to 20 for WebKB, and set to 35 for DI. For the centroid-based method, the applied weighting systems are those shown in Section 4.3. All experiments were performed with 10-fold cross validation. That is, 90% of documents are kept as a training set while the rest 10% are used for testing. The performance was measured by classification accuracy defined as the ratio between the number of documents assigned with correct classes and the total number of test documents. As a preprocess, some stop words (e.g., a, an, the) and all tags (e.g., <B>, </HTML>) were omitted from documents to eliminate the affect of these common words and typographic words.

In the rest, first the results on flat and hierarchical classification on the data sets are shown, followed by that of multi-dimensional classification. Finally overall discussion is given.

5.1 Flat-based Classification

In this experiment, test documents are classified into the most specified classes say D_{12}, which are the combinations of two dimensions, D_1 and D_2. Therefore, the number of classes equals to the product of the number of classes in each dimension. That is 20 (=5×4) classes for WebKB and 35 (=7×5) classes for DI. A test document was assigned the class that gained the highest score from the classifier applied. Table 1 displays the classification accuracy of flat classification on WebKB and DI data sets. Here, two measures, two-dimension and single-dimension accuracy, are taken into account.

	WebKB			DI		
	$D_{12}\rightarrow D_1$	$D_{12}\rightarrow D_2$	D_{12}	$D_{12}\rightarrow D_1$	$D_{12}\rightarrow D_2$	D_{12}
k-NN	68.02	84.69	57.32	79.46	66.14	60.04
NB	80.23	78.76	62.66	93.75	73.97	69.61
CB1	77.54	91.52	71.59	96.14	72.08	69.42
CB2	71.52	89.12	63.42	89.49	80.58	73.28

Table 1. Flat classification accuracy (%)

In the table, D_{12} shows the two-dimension accuracy where the test document is completely assigned to the correct class. $D_{12}\rightarrow D_1$ and $D_{12}\rightarrow D_2$, the single-dimension accuracy, mean the accuracy of the first and second dimensions where the classes in D_1 and D_2 dimensions are generated from the result class D_{12}, respectively. The result shows that the centroid-based classifiers perform better than *k*-NN and NB. CB1 and CB2 works well on WebKB and DI, respectively. Even low two-dimension accuracy, high single-dimension accuracy is obtained.

5.2 Hierarchical-based Classification

Since there are two dimensions in the data set, hierarchical-based classification can be held in two different ways according to the classifying order. In the first version, documents are classified based on the first dimension to determine the class to which those documents belong. They are further classified again according to the second dimension using the model of that class. The other version classifies documents based on the second dimension first and then the first dimension. The results are shown in Table 2. In the tables, D_1, D_2 and D^*_{12} mean the accuracy of the first dimension, the second dimension and the two-dimension accuracy, respectively. $D_1+D^*_{12}=>D_2$ expresses the accuracy of the second dimension that used the result from the first dimension during classifying the second dimension. $D_2+D^*_{12}=>D_1$ represents the accuracy of the second dimension that used the result from the first dimension during classifying the first dimension.

From the results, we found that the centroid-based classifiers also perform better than *k*-NN and NB, and CB1 works well on WebKB while CB2 gains the highest accuracy on DI. In almost cases, the hierarchical-based classification performs better than the flat-based classification. Moreover, an interesting observation is that classifying on the worse dimension before the better one yields a better result.

	WebKB			DI		
	D_1	$D_1+D^*_{12}=>D_2$	D^*_{12}	D_1	$D_1+D^*_{12}=>D_2$	D^*_{12}
k-NN	69.85	84.31	58.61	80.20	73.17	60.20
NB	80.54	78.85	62.42	95.00	73.35	70.38
CB1	80.42	91.28	73.90	96.23	73.44	69.26
CB2	76.04	88.59	66.87	91.43	80.09	74.24

	WebKB			DI		
	$D_2+D^*_{12}=>D_1$	D_2	D^*_{12}	$D_2+D^*_{12}=>D_1$	D_2	D^*_{12}
k-NN	67.42	83.34	56.04	79.29	76.36	61.25
NB	79.92	87.45	69.35	93.08	83.75	78.33
CB1	77.99	90.02	70.18	95.60	73.44	70.36
CB2	71.44	92.36	65.78	88.33	84.87	76.05

Table 2. Hierarchical classification accuracy(%)
(upper: D_1 before D_2 , lower: D_2 before D_1)

5.3 Multi-dimensional Classification

In the last experiment, multi-dimensional classification is investigated. Documents are classified twice based on two dimensions independently. The results of the first and second dimensions are combined to be the suggested class for a test document. The classification accuracy of multi-dimensional classification is shown in Table 3.

	WebKB			DI		
	D_1	D_2	$D_1+D_2\rightarrow D_{1+2}$	D_1	D_2	$D_1+D_2\rightarrow D_{1+2}$
k-NN	69.85	83.34	57.37	80.20	76.36	61.85
NB	80.54	87.45	69.66	95.00	83.75	79.51
CBC1	80.42	90.02	72.52	96.23	73.44	70.05
CBC2	76.04	92.36	69.90	91.43	84.87	77.99

Table 3. Multi-dimensional.classification accuracy (%)

In the tables, D_1 and D_2 mean the accuracy of the first and second dimensions, respectively. $D_1+D_2\rightarrow D_{1+2}$ is the two-dimension accuracy of the class which is the combination of classes suggested in the first dimension and the second dimension. From the results, we found that CB1 performs well on WebKB but NB gains the highest accuracy on DI. The multi-dimensional classification outperforms flat classification in most cases but sometime the hierarchical-based classification performs well.

5.4 Overall Evaluation and Discussion

Two accuracy criteria are (1) all dimensions are correct or (2) some dimensions are correct. The classification accuracy based on the first criterion is shown in all previous tables as the two-dimension accuracy. As the second criterion, the classification accuracy can be evaluated when some dimensions are correct. The result is summarized in Table 4. The multi-dimensional classification outperforms other two methods for WebKB but the hierarchical-based classification sometimes works better for DI.

	WebKB				DI			
	F	H1	H2	M	F	H1	H2	M
k-NN	72.80	77.08	75.38	78.28	76.36	76.69	77.83	76.59
NB	83.86	79.70	83.69	89.38	79.50	84.18	88.42	84.00
CB1	84.11	85.85	84.01	84.84	84.53	84.84	84.52	85.22
CB2	85.04	82.32	81.90	88.15	80.32	85.76	86.60	84.20

Table 4. Classification accuracy (%) when some dimensions are correct.

From this result, some observations can be given as follows. There are two tradeoff factors that affect classification accuracy of multi-dimensional category model: training set size and the granularity of classes. The flat-based classification in the multi-dimensional model deals with the finest granularity of classes because all combinations of classes from predefined dimensions are combined to form a large set of classes. Although this precise representation of classes may increase the accuracy, the flat-based classification suffers with sparseness problem where the number of training data per class is reduced. The accuracy is low when the training set is small. The multi-dimensional-based classification copes with the coarsest granularity of the classes. Therefore the number of training document per class is larger than flat-based classification approach but the representation of classes is not exact. However, It works well when we have a relatively small training set. The hierarchical-based classification occupies a medium granularity of classes. However, the size of training set is smaller than multi-dimensional approach at the low level of the hierarchy. It works well when the training set is medium.

6 Conclusion

In this paper, a multi-dimensional framework on text classification was proposed. The framework applies a multi-dimensional category for representing classes, in contrast with traditional flat and hierarchical category models. Classifying text documents based on a multi-dimensional category model can be performed using the multi-dimensional-based classification or the flat and hierarchical classifications. By experiments on two data sets and three algorithms, *k*-NN, naïve Bayes and centroid-based methods, the results show that the multi-dimensional-based and hierarchical-based classifications outperform flat-based one.

References

Chuang W. T. et al. (2000), *A Fast Algorithm for Hierarchical Text Classification*. Data Warehousing and Knowledge Discovery, 409-418.

Dumais S. T. and Chen H. (2000) *Hierarchical Classification of Web Content,* In Proc. of the 23rd International ACM SIGIR, pp. 256-263.

Eui-Hong H. and Karypis G. (2000) *Centroid-Based Document Classification: Analysis & Experimental Results*. In Proc. of European Conference on PKDD, pp. 424-431.

Jiawei H. and Micheline K. (2001) *Data Mining: Concepts and Techniques*. Morgan Kaufmann publishers.

Lewis D. D. and Ringuette M. (1994) *A Comparison of Two Learning Algorithms for Text Categorization*. In Proc. of Third Annual Symposium on Document Analysis and Information Retrieval, pages 81-93.

McCallum A. et al. (1998) *Improving Text Classification by Shrinkage in a Hierarchy of Classes*, In Proc. of the 15th International Conference on Machine Learning, pp. 359-367.

Theeramunkong T. and Lertnattee V. (2001) *Improving Centroid-Based Text Classification Using Term-Distribution-Based Weighting System and Clustering*. In Proc. of International Symposium on Communications and Information Technology (ISCIT 2001), pp. 33-36.

Wiener E. D. et al. (1995) *A Neural Network Approach to Topic Spotting*. In Proc. of SDAIR-95, the 4th Annual Symposium on Document Analysis and Information Retrieval. pp. 317-332.

Yang Y. and Chute C. G. (1992) *A Linear Least Square Fit Mapping Method for Information Retrieval from Natural Language Texts*. In Proc. of the 14th International Conference on Computational Linguistics, pp. 358-362.

Yang, Y. and Liu X. (1999) *A Re-examination of Text Categorization Methods*. In Proc. of the 22nd ACM SIGIR Conference, 42-49.