Automatic Text Categorization using the Importance of Sentences

Youngjoong Ko, Jinwoo Park, and Jungyun Seo
Department of Computer Science,
Sogang University
1 Sinsu-dong, Mapo-gu
Seoul, 121-742, Korea
{kyj,jwpark}@nlpzodiac.sogang.ac.kr, seojy@ccs.sogang.ac.kr

Abstract

Automatic text categorization is a problem of automatically assigning text documents to predefined categories. In order to classify text documents, we must extract good features from them. In previous research, a text document is commonly represented by the term frequency and the inverted document frequency of each feature. Since there is a difference between important sentences and unimportant sentences in a document, the features from more important sentences should be considered more than other features. In this paper, we measure the importance of sentences using text summarization techniques. Then a document is represented as a vector of features with different weights according to the importance of each sentence. To verify our new method, we conducted experiments on two language newsgroup data sets: one written by English and the other written by Korean. Four kinds of classifiers were used in our experiments: Naïve Bayes, Rocchio, *k*-NN, and SVM. We observed that our new method made a significant improvement in all classifiers and both data sets.

Introduction

The goal of text categorization is to classify documents into a certain number of pre-defined categories. Text categorization is an active research area in information retrieval and machine learning. A wide range of supervised learning algorithms has been applied to this problem using a training data set of categorized documents. For examples, there are the Naïve Bayes (McCallum et al., 1998; Ko et al., 2000), Rocchio (Lewis et al., 1996), Nearest Neighbor (Yang et al., 2002), and Support Vector Machines (Joachims, 1998).

A text categorization task consists of a training phase and a text classification phase. The former includes the feature extraction process and the indexing process. The vector space model has been used as the conventional method for text representation (Salton et al., 1983). This model represents a document as a vector of features using Term Frequency (TF) and Inverted Document Frequency (IDF). This model simply counts TF without considering where the term occurs. But each sentence in a document has different importance for identifying the content of the document. Thus, by assigning a different weight according to the importance of the sentence to each term, we can achieve better results. For this problem, several techniques have been studied. First, term weights were differently weighted by the location of a term, so that the structural information of a document was applied to term weights (Murata et al., 2000). But this method supposes that only several sentences, which are located at the front or the rear of a document, have the important meaning. Hence it can be applied to only documents with fixed form such as articles. The next technique used the title of a document in order to choose the important terms (Mock et al., 1996). The terms in the title were handled importantly. But a drawback of this method is that some titles, which do not contain well the meaning of the document, can rather increase the ambiguity of the meaning. This case often comes out in documents with a informal style such as *Newsgroup* and *Email*. To

overcome these problems, we have studied text summarization techniques with great interest. Among text summarization techniques, there are statistical methods and linguistic methods (Radev et al., 2000; Marcu et al., 1999). Since the former methods are simpler and faster than the latter methods, we use the former methods to be applied to text categorization. Therefore, we employ two kinds of text summarization techniques; one measures the importance of sentences by the similarity between the title and each sentence in a document, and the other by the importance of terms in each sentence.

In this paper, we use two kinds of text summarization techniques for classifying important sentences and unimportant sentences. The importance of each sentence is measured by these techniques. Then term weights in each sentence are modified in proportion to the calculated sentence importance. To test our proposed method, we used two different newsgroup data sets; one is a well known data set, the Newsgroup data set by Ken Lang, and the other was gathered from Korean UseNet discussion group. As a result, our proposed method showed the better performance than basis system in both data sets.

The rest of this paper is organized as follows. Section 1 explains the proposed text categorization system in detail. In section 2, we discuss the empirical results in our experiments. Section 3 is devoted to the analysis of our method. The final section presents conclusions and future works.

1. The Proposed Text Categorization System

The proposed system consists of two modules as shown in Figure 1: one module for training phase and the other module for text classification phase. The each process of Figure 1 is explained in the following sections.

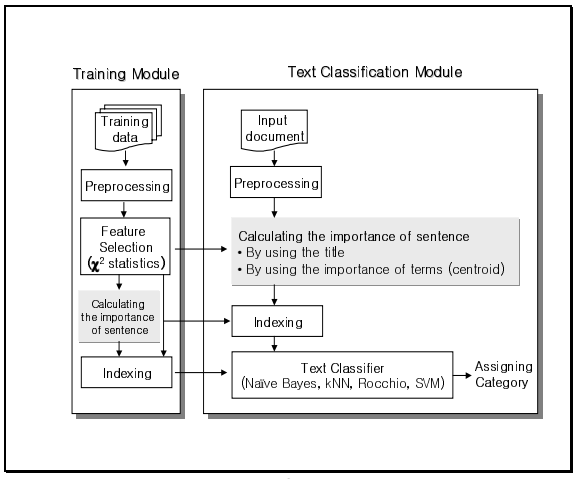

Figure 1. Overview of the proposed system

1.1 Preprocessing

A document from newsgroup data is composed of subject, author, data, group, server, message ID, and body. In our system, we use only the contents of subject and body.

The contents of documents are segmented into sentences. Then we extract content words from each sentence and represent each sentence as a vector of content words. To extract content words, we use two kinds of POS taggers: Brill tagger for English and Sogang POS tagger for Korean. We employ TF values as term weights of content words in each sentence.

1.2 Measuring the importance of Sentences

The importance of each sentence is measured by two methods. First, the sentences, which are more similar to the title, have higher weights. In the next method, we first measure the importance of terms by TF, IDF, and χ^2 statistic values. Then we assign the higher importance to the sentence with more important terms. Finally, the importance of a sentence is calculated by combination of two methods.

1.2.1 The importance of sentences by the title

Generally, we believe that a title summarizes the important content of a document (Endres-Niggemeyer et al., 1998). By Mock (1996), terms occurred in the title have higher weights. But the effectiveness of this method depends on the quality of the title. In many cases, the titles

of documents from Newsgroup or Email do not represent the contents of these documents well. Hence we use the similarity between each sentence and the title instead of directly using terms in the title. The similar sentences to the title contain important terms generally. For example, "I have a question." This title does not contain any meaning about the contents of a document. Nevertheless, sentences with the term, 'question', must be handled importantly because they can have key terms about the question.

We measure the similarity between the title and each sentence, and then we assign the higher importance to the sentences with the higher similarity. The title and each sentence of a document are represented as the vectors of content words. The similarity value of them is calculated by the inner product and the calculated values are normalized into values between 0 and 1 by a maximum value. The similarity value between title T and sentence S_i in a document d is calculated by the following formula:

$$Sim(S_i, T) = \frac{\vec{S}_i \cdot \vec{T}}{\max_{S_i \in d}(\vec{S}_i \cdot \vec{T})} \quad (1)$$

where $\vec{T}$ denotes a vector of the title, and $\vec{S}_i$ denotes a vector of a sentence.

1.2.2 The importance of sentences by the importance of terms

Since the method by the title still depends on the quality of the title, it can be useless in the document with a meaningless title or no title. Besides, the sentences, which do not contain important terms, need not be handled importantly although they are similar to the title. On the contrary, sentences with important terms must be handled importantly although they are dissimilar to the title.

In consideration to these points, we first measure the importance values of terms by TF, IDF, and χ^2 statistic value, and then the sum of the importance values of terms in each sentence is assigned to the importance value of the sentence. Here, since the χ^2 statistic value of a term presents information of the term for document classification, it is added to our method unlike the conventional TF-IDF. In this method, the importance value of a sentence S_i in a document d is calculated as follows:

$$Cen(S_i) = \frac{\sum_{t \in S_i} tf(t) \times idf(t) \times \chi^2(t)}{\max_{S_i \in d} \left\{ \sum_{t \in S_i} tf(t) \times idf(t) \times \chi^2(t) \right\}} \quad (2)$$

where *tf(t)* denotes term frequency of term *t*, *idf(t)* denotes inverted document frequency, and $\chi^2(t)$ denotes χ^2 statistic value.

1.2.3 The combination of two sentence importance values

Two kinds of sentence importance are simply combined by the following formula:

$$Score(S_i) = 1.0 + k_1 \times Sim(S_i, T) + k_2 \times Cen(S_i) \quad (3)$$

In formula (3), k_1 and k_2 are constant weights, which control the rates of reflecting two importance values.

1.2.4 The indexing process

The importance value of a sentence by formula (3) is used for modifying TF value of a term. That is, since a TF value of a term in a document is calculated by the sum of the TF values of terms in each sentence, the modified TF value (*WTF(d,t)*) of the term *t* in the document *d* is calculated by formula (4).

$$WTF(d, t) = \sum_{S_i \in d} tf(S_i, t) \times Score(S_i) \quad (4)$$

where $tf(S_i,t)$ denotes TF of the term *t* in sentence S_i.

By formula (4), the terms, which occurr in a sentence with the higher importance value, have higher weights than the original TF value. In our proposed method, we compute the weight vectors for each document using the *WTF* and the conventional TF-IDF scheme (Salton et al., 1988). The weight of a term *t* in a document *d* is calculated as follows:

$$w(d,t) = \frac{WTF\ (d,t) \times \log\left(\frac{N}{n_t}\right)}{\sqrt{\sum_{i=1}^{T}\left[WTF\ (d,t_i) \times \log\left(\frac{N}{n_{t_i}}\right)\right]^2}} \quad (5)$$

where N is the number of documents in the training set, T is the number of features limited by feature selection, and n_t is the number of training documents in which t occurs.

The weight by formula (5) is used in k-NN, Rocchio, and SVM. But Naïve Bayes classifier uses only *WTF* value.

2. Empirical Evaluation

2.1 Data Sets and Experimental Settings

To test our proposed system, we used two newsgroup data sets written by two different languages: English and Korean.

The **Newsgroups** data set, collected by Ken Lang, contains about 20,000 articles evenly divided among 20 UseNet discussion groups (McCallum et al., 1998). 4,000 documents (20%) were used for test data and the remaining 16,000 documents (80%) for training data. Then 4,000 documents from training data were selected for a validation set. After removing words that occur only once or on a stop word list, the vocabulary from training data has 51,018 words (with no stemming).

The second data set was gathered from the Korean UseNet group. This data set contains a total of 10,331 documents and consists of 15 categories. 3,107 documents (30%) are used for test data and the remaining 7,224 documents (70%) for training data. The resulting vocabulary from training data has 69,793 words. This data set is uneven data set as shown in Table 1.

Table 1 The constitution of Korean newsgroup data set

Category	Training data	Test data	Total
han.arts.music	315	136	451
han.comp.databas	198	86	284
han.comp.devtools	404	174	578
han.comp.lang	1,387	595	1,982
han.comp.os.linux	1,175	504	1,679
han.comp.os.window	517	222	739
han.comp.sys	304	131	435
han.politics	1,469	630	2,099
han.rec.cars	291	126	417
han.rec.games	261	112	373
han.rec.movie	202	88	290
han.rec.sports	130	56	186
han.rec.travel	102	45	147
han.sci	333	143	476
han.soc.religion	136	59	195
Total	7,224	3,107	10,331

We used χ^2 statistics for statistical feature selection (Yang et al., 1997). To evaluate our method, we implemented Naïve Bayes, k-NN, Rocchio, and SVM classifier. The k in k-NN was set to 30 and α=16 and β=4 were used in our Rocchio classifier. This choice was based on our previous parameter optimization learned by validation set. For SVM, we used the linear models offered by SVM^{light}.

As performance measures, we followed the standard definition of recall, precision, and F_1 measure. For evaluation performance average across categories, we used the micro-averaging method and macro-averaging method.

2.2 Experimental Results

We tested our system through the following steps. First, using the validation set of **Newsgroup** data set, we set the number of feature and the constant weights (k_1 and k_2) in the combination of two importance values in the section 1.2.3. Then, using the resulting values, we conducted experiments and compared our system with a basis system; the basis system used the conventional TF and our system used WTF by formula (4).

2.2.1 Setting the number of features

First of all, we set the number of features in each classifier using validation set of training data. The number of features in this experiment was limited from 1,000 to 20,000 by feature selection. Figure 2 displays the performance curves for the proposed system and the basis system using SVM. We simply set both constant

weights (k_1 and k_2) to 1.0 in this experiment.

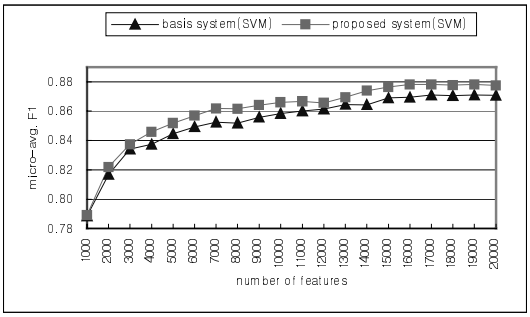

Figure 2. Comparison of proposed system and basis system using SVM

As shown in Figure 2, the proposed system achieved the better performance than the basis system over all intervals. We set the number of features for SVM to 7,000 with regard to the convergence of the performance curve and running time. By the similar method, the number of features in other classifiers was set: 7,000 for Naïve Bayes, 10,000 for Rocchio, and 9,000 for *k*-NN. Note that, over all intervals and all classifiers, the performance of the proposed system was better than that of the basis system.

2.2.2 Setting the constant weights k_1 and k_2

In advance of the experiment for setting the constant weights, we evaluated two importance measure methods and their combination method individually; we used simply the same value for k_1 and k_2 (k_1=k_2) in the combination method (formula (3)). We observed the results in each interval when constant weights were changed from 0.0 to 3.0. In Figure 3, S*im(S,T)* denotes the method using the title, *Cen(S)* the method using the importance of terms, and S*im(S,T)+Cen(S)* the combination method.

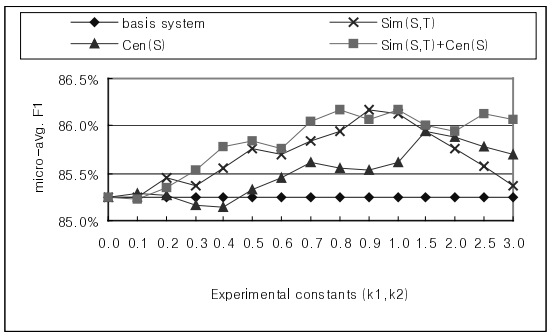

Figure 3. Comparison of importance measure methods in different constant weights (k_1 and k_2)

In this experiment, we used SVM as a classifier and set feature number to 7,000. We mostly obtained a best performance in the combination method.

In order to set the constant weights k_1 and k_2 in each classifier, we carried out the total 900 trials on the validation set because each method had 30 intervals from 0.0 to 3.0 (interval size: 0.1). As a result, we obtained the best performance at 1.5 (k_1) and 0.4 (k_2) for SVM: 1.9 and 3.0 for Naïve Bayes, 2.0 and 0.0 for Rocchio, and 0.8 and 2.8 for *k*-NN. These constant weights of each classifier were used in the following experiments.

2.2.3 Results in two newsgroup data sets

In this section, we reported results in two newsgroup data sets using parameters determined above experiments.

Table 2. Results in English newsgroup data set

	Naïve Bayes		**Rocchio**	
	basis system	proposed system	basis system	proposed system
macro-avg F_1	83.2	**84.4**	79.8	**80.5**
micro-avg F_1	82.9	**84.3**	79.4	**80.3**
	***k*-NN**		**SVM**	
	basis system	proposed system	basis system	proposed system
macro-avg F_1	81.3	**82.7**	85.8	**86.4**
micro-avg F_1	81.1	**82.5**	85.8	**86.3**

Table 3. Results in Korean newsgroup data set

	Naïve Bayes		**Rocchio**	
	basis system	proposed system	basis system	proposed system
macro-avg F_1	78.4	**80.8**	77.8	**79.2**
micro-avg F_1	79.1	**81.3**	78.7	**80.1**
	***k*-NN**		**SVM**	
	basis system	proposed system	basis system	proposed system
macro-avg F_1	78.6	**80.6**	84.8	**85.5**
micro-avg F_1	79.9	**81.3**	86.0	**86.5**

In both data sets, the proposed system produced the better performance in all classifiers. As a result, our proposed system can be useful for all classifiers and both two different languages.

3. Discussions

Salton stated that a collection of small tightly clustered documents with wide separation between individual clusters should produce the best performance (Salton et al., 1975). Hence we employed the method used by Salton et al. (1975) to verify our method. Then we conducted experiments in English newsgroup data set (**Newsgroup** data set) and observed the resulting values.

We define the cohesion within a category and the cohesion between categories. The cohesion within a category is a measure for similarity values between documents in the same category. The cohesion between categories is a measure for similarities between categories. The former is calculated by formula (6) and the latter by formula (7):

$$\vec{C}_k = \frac{1}{|I_k|}\sum_{\vec{d}\in I_k}\vec{d} \quad , \quad Co_{withiin} = \frac{1}{|D|}\sum_{k=1}^{K}\sum_{\vec{d}\in I_k}\vec{d}\cdot\vec{C}_k \qquad (6)$$

$$\vec{C}_{glob} = \frac{1}{|D|}\sum_{k=1}^{K}|I_k|\cdot\vec{C}_k\,,$$

$$Co_{between} = \frac{1}{|D|}\sum_{k=1}^{K}|I_k|\cdot\left(\vec{C}_{glob}\cdot\vec{C}_k\right) \qquad (7)$$

where D denotes the total training document set, I_k denotes training document set in k-th category, $\vec{C}_k$ denotes a centroid vector of k-th category, and $\vec{C}_{glob}$ denotes a centroid vector of the total training documents.

An indexing method with a high cohesion within a category and a low cohesion between categories should produce the better performance in text categorization. First, we measured the cohesion within a category in each indexing method: a basis method by the conventional TF value, a method using the title (S*im(S,T)*), a method using the importance of terms (*Cen(S)*), and a combination method (S*im(S,T)+Cen(S)*). Figure 4 shows the resulting curves in each different constant weight; we used simply the same values for k_1 and k_2 in the combination method.

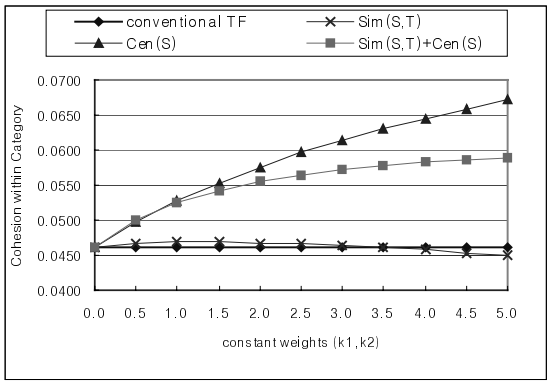

Figure 4. The cohesion within a category

As shown in Figure 4, *Cen(S)* shows the highest cohesion value, but S*im(S,T)* does not have an effect on the cohesion in comparison with the method by conventional TF value.

Figure 5 displays the resulting curves of the cohesion between categories as the same manner in Figure 4.

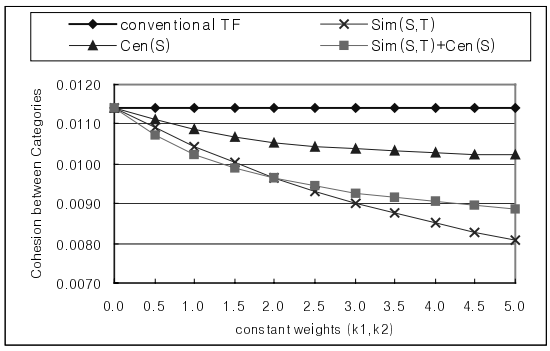

Figure 5. The cohesion between categories

We obtained the lowest cohesion value in S*im(S,T)*. Using *Cen(S)*, the resulting cohesion values are slightly higher than those of the method by conventional TF value. In both Figure 4 and Figure 5, the cohesion values of the combination method show middle values between *Sim(S,T)* and *Cen(S)*.

By the results in Figure 4 and Figure 5, we can observe that our proposed indexing method reforms the vector space for the better

performance: high cohesion within a category and low cohesion between categories. Using the proposed indexing method, the document vectors in a category are located more closely and individual categories are separated more widely. These effects were also observed in our experiments. According to properties of each classifier, *k*-NN has an advantage in a vector space with the high cohesion within a category and Rocchio has an advantage in a vector space with the low cohesion between categories. We achieved the similar results in our experiments. That is, *k*-NN produced a better performance by using *Cen(S)* and Rocchio produced a better performance by using S*im(S,T)*. Table 4 shows the summarized results in each individual method of *k*-NN and Rocchio.

Table 4. Top performance by two different methods

	***k*-NN**		**Rocchio**	
	sim(S,T)	*Cen(S)*	*Sim(S,T)*	*Cen(S)*
macro-avg F_1	80.6	**82.4**	**79.7**	78.8

Conclusions

In this paper, we have presented a new indexing method for text categorization using two kinds of text summarization techniques; one uses the title and the other uses the importance of terms. For our experiments, we used two different language newsgroup data sets and four kinds of classifiers. We achieved the better performance than the basis system in all classifiers and both two languages. Then we verified the effect of the proposed indexing method by measuring the two kinds of cohesion. We confirm that the proposed indexing method can reform the document vector space for the better performance in text categorization. As a future work, we need the additional research for applying the more structural information of document to text categorization techniques and testing the proposed method on other types of texts such as newspapers with fixed form.

References

Endres-Niggemeyer B. et al, (1998) *Summarizing Information.* Springer-Verlag Berlin Heidelberg, pp. 307-338.

Joachims T. (1998) Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In European Conference on Machine Learning (ECML), pp. 137-142.

Ko Y. and Seo J. (2000) Automatic Text Categorization by Unsupervised Learning. In Proceedings of the 18th International Conference on Computational Linguistics (COLING'2000), pp. 453-459.

Lewis D.D., Schapire R.E., Callan J.P., and Papka R. (1996) Training Algorithms for Linear Text Classifiers. In Proceedings of the 19th International Conference on Research and Development in Information Retrieval (SIGIR'96), pp.289-297.

Marcu D., (1999) Discourse trees are good indicators of importance in text. Advances in Automatic Text Summarization. pp.123-136, The MIT Press.

McCallum A. and Nigam K. (1998) A Comparison of Event Models for Naïve Bayes Text Classification. *AAAI* '98 workshop on Learning for Text Categorization. pp. 41-48.

Mock K.J., (1996) Hybrid hill-climbing and knowledge-based techniques for intelligent news filtering. In Proceedings of the National Conference on Artificial Intelligence (AAAI'96).

Murata M., Ma Q., Uchimoto K., Ozaku H., Isahara H., and Utiyama M. (2000) Information retrieval using location and category information. Journal of the Association for Natural Language Processing, 7(2).

Radev D.R., Jing H., and Stys-Budzikowska M. (2000) Summarization of multiple documents: clustering, sentence extraction, and evaluation. In Proceedings of ANLP-NAACL Workshop on Automatic Summarization.

Salton G., Yang C., and Wang A. (1975) A vector space model for automatic indexing. *Communications of the ACM*, Vol. 18, No. 11, pp. 613-620.

Salton G., Fox E.A., and Wu H. (1983) Extended Boolean information retrieval. Communications of the ACM 26 (12), pp. 1022-1036.

Salton G. and Buckley C. (1988) Term weighting approaches in automatic text retrieval. Information Processing and Management, 24:513-523.

Yang Y. and Pedersen J.P. (1997) Feature selection in statistical learning of text categorization. In The Fourteenth International Conference on Machine Learning, pages 412-420.

Yang Y., Slattery S., and Ghani R. (2002) A study of approaches to hypertext categorization, Journal of Intelligent Information Systems, Volume 18, Number 2.