International Journal of Fuzzy Logic and Intelligent Systems, vol. 8, no. 3, September 2008 pp. 175-184

Evaluation of User Profile Construction Method by Fuzzy Inference

Byeong Man Kim[1], Sun-Ok Rho[1], Sang Yeop Oh[1], Hyun Ah Lee[1] and Jong-Wan Kim[2]

[1]Kumoh National Institute of Technology, Korea {bmkim, sorho, syoh, halee}@kumoh.ac.kr
[2]Daegu University, Korea jwkim@daegu.ac.kr

Abstract

To construct user profiles automatically, an extraction method for representative keywords from a set of documents is needed. In our previous works, we suggested such a method and showed its usefulness. Here, we apply it to the classification problem and observe how much it contributes to performance improvement. The method can be used as a linear document classifier with few modifications. So, we first evaluate its performance for that case. The method is also applicable to some non-linear classification methods such as GIS (Generalized Instance Set). In GIS algorithm, generalized instances are built from training documents by a generalization function and then the K-NN algorithm is applied to them, where the method can be used as a generalization function. For comparative works, two famous linear classification methods, Rocchio and Widrow-Hoff algorithms, are also used. Experimental results show that our method is better than the others for the case that only positive documents are considered, but not when negative documents are considered together.

Key words: User Modeling, Information Filtering, Classification, Keywords Extraction

1. Introduction

Humans utilizing web search engines or various information retrieval (IR) systems make a query based on restricted vocabulary and expertise for their preferred domain to find the most appropriate contents. Similarly users of information filtering (IF) systems describe their interests in their own profiles to have a recommendation or delivery service of appropriate information.

IR systems provide user friendly service and high-quality retrieval outcomes by web surfing with a provided query and taking user feedback from retrieval results or performing supplementary works such as automatic query term modification and its reweighting. IF systems also do similar kinds of profile modification processes as IR systems.

User profile can be constructed by hand, or learned automatically with the explicit or implicit user feedback. Some systems require users to explicitly specify their profiles, often as a set of keywords or categories. Studies have shown that such explicit feedback from the user is clearly useful [4, 21]. However, it is difficult for a user to exactly and correctly specify their information needs.

Moreover, many users are unwilling to provide relevance judgments on documents in practice [15, 19]. An alternative is to use implicit feedback based on user's behavior to automatically construct user models [6, 9, 14]. In this case, system should construct user profile, often, by extracting automatically representative keywords based on a set of feedback documents.

Kim et al. [8] propose RKEF (Representative Keywords Extraction by Fuzzy inference) method for extracting representative keywords from a few documents that might interest users, where a fuzzy inference technique and a term reweighting scheme based the term co-occurrence similarity are applied. It first extracts candidate terms and choose a number of terms called initial representative keywords (IRKs) among them through fuzzy inference. Then, by expanding IRKs and reweighting them using term co-occurrence similarity, the final representative keywords are extracted.

RKEF method can be applied to the document classification problem. So, we, in this paper, apply it to that problem and observe how much it contributes to performance improvement. The method can be used as a linear document classifier with few modifications. So, we first evaluate its performance for that case. The method is also applicable to some non-linear classification methods such as GIS (Generalized Instance Set) [10, 11]. The basic idea of GIS algorithm is to construct a set of generalized instances to replace original training examples by generalization function and apply k-NN (k-nearest neighbor) algorithm [20] to them. In this paper, RKEF is chosen as a generalization function of GIS and compared to Rocchio and Widrow-Hoff algorithms.

The remainder of the paper is organized as follows: next section introduces RKEF method briefly. Section 3 presents performance evaluation when RKEF method is used as a linear classifier. In Section 4, the performance evaluation when RKEF is used as a generalization function of GIS is given, and concluding remarks are followed in Section 5.

Manuscript received Apr.29, 2007; revised Jun. 30, 2008.
This paper was supported by Research Fund, Kumoh National Institute of Technology.

2. Extraction of representative keywords from a few documents by fuzzy inference

In [8], we suggested RKEF method in order to extract representative keywords (RKs) reflecting user preferred contents from a few example documents. Here we briefly introduce RKEF method that is almost the same as the one in [8] except some details. The entire process of RKEF is composed of three steps:

i. Calculate the importance of candidate keywords by fuzzy inference.
ii. Select initial representative keywords.
iii. Expand and reweight initial representative keywords.

2.1 Calculation of the importance of candidate keywords by fuzzy inference

Representative keywords are a kind of summarization for the documents. Their weights should reflect their representative ability. Therefore, several factors should be considered to decide weights of candidate terms which are terms extracted from a training document set by simple processing. Those factors - normalized term frequency NTF, the normalized document frequency NDF, and the normalized inverse document frequency NIDF - are represented in Equation 1.

$$NTF_i = \frac{TF_i / DF_i}{\max_j [TF_j / DF_j]}$$
$$NDF_i = \frac{DF_i / TD}{\max_j [DF_j / TD]} \quad (1)$$
$$NIDF_i = \frac{IDF_i}{\max_j IDF_j}$$

where, TF_i is the frequency of term t_i in the example (or feedback) documents; DF_i is the number of documents having term t_i in the example documents; TD is the number of the example documents; IDF_i represents the inverse document frequency of term t_i over an entire document collection, not over example documents.

Because the factors essentially have inexact and uncertain characteristics, we combine them by fuzzy inference instead of a simple equation to obtain the weight of candidate keywords. Fig. 1 shows the membership functions of the input/output variables used for fuzzy inference. The term weight TW is derived by Mamdani's fuzzy inference method with the 18 fuzzy rules in Table 1.

The overall procedure for calculating the weight of a candidate term is as follows. Refer [8] for detail.

i. Apply the NDTF, NDF, and NIDF fuzzy values to the antecedent portions of 18 fuzzy rules.
ii. Find the minimum value among the membership degrees of three input fuzzy values.
iii. Classify every 18-membership degree into 6 groups according to the fuzzy output variable TW.

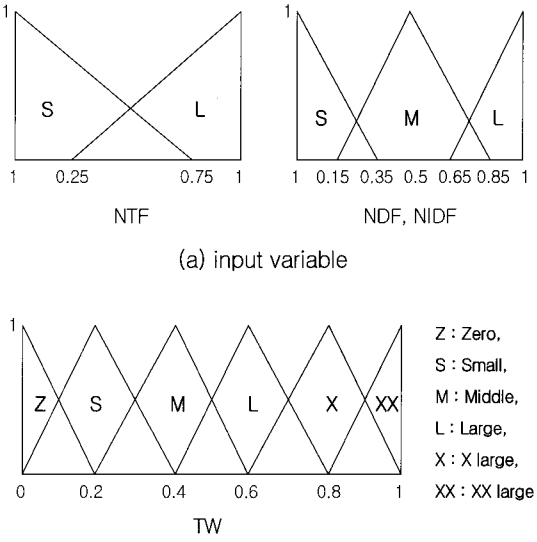

Fig.1. Fuzzy membership functions of input and output variables

Table 1. Fuzzy inference rules

NDF \ NIDF	S	M	L
S	Z	Z	S
M	Z	M	L
L	S	L	X

NTF = S

NDF \ NIDF	S	M	L
S	Z	S	M
M	S	L	X
L	M	X	XX

NTF = L

iv. Calculate the maximum output value for each group and then generate 6 output values.

For instance, let us assume, there is one term, whose NIDF is 0.35, NDF is 0.2 and NDTF is 0.3. The degree of membership is determined by plugging the selected input parameter (NIDF, NDF or NDTF) into the horizontal axis and projecting vertically to the upper boundary of the membership function(s) in Fig. 1. Therefore the result is as follows.

NIDF=0.35 : S=0.00, M=0.57;
NDF=0.20 : S=0.43, M=0.14;
NDTF=0.30 : S=0.53, L=0.07.

Now referring back to the rules, only 8 rules out of 18 rules need to be selected. The effective rules are listed as follows.

1. if (NIDF=S,NDF=S,NDTF=S) then TW=Z
   min{S=0.00,S=0.43,S=0.53}=0.00
2. if (NIDF=S,NDF=M,NDTF=S) then TW=Z
   min{S=0.00,M=0.14,S=0.53}=0.00
3. if (NIDF=M,NDF=S,NDTF=S) then TW=Z
   min{M=0.57,S=0.43,S=0.53}=0.43
4. if (NIDF=M,NDF=M,NDTF=S) then TW=M
   min{M=0.57,M=0.14,S=0.53}=0.14

5. if (NIDF=S,NDF=S,NDTF=L) then TW=Z
   min{S=0.00,S=0.43,L=0.07}=0.00
6. if (NIDF=S,NDF=M,NDTF=L) then TW=S
   min{S=0.00,M=0.14,L=0.07}=0.00
7. if (NIDF=M,NDF=S,NDTF=L) then TW=S
   min{M=0.57,S=0.43,L=0.07}=0.07
8. if (NIDF=M,NDF=M,NDTF=L) then TW=L
   min{M=0.57,M=0.14,L=0.07}=0.07

Then we calculate the maximum output value for each group and then generate 6 output values, as shown in Fig. 2, which consist a fuzzy set of TW as follows.

$$TW = \{Z\backslash 0.43,\ S\backslash 0.07,\ M\backslash 0.14,\ L\backslash 0.07,\ X\backslash 0.0,\ XX\backslash 0.0\}$$

At last, the center of gravity is used to defuzzify the output into one value.

$$TW=(0.43\times 0.1+0.07\times 0.2+0.14\times 0.4+0.07\times 0.7)\ /\ (0.1+0.2+0.4+0.7) = 0.116$$

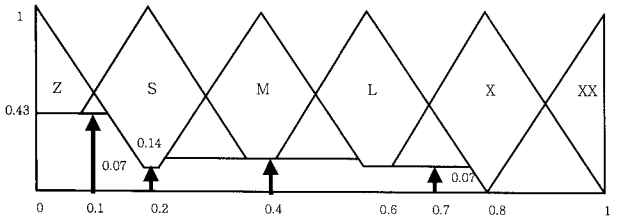

Fig. 2. Defuzzification of the output

As shown in Table 1, we give a higher weight to a term with a high NDF if it's not a general term. This comes from our intuition that the terms across many relevant documents are highly probable to be a representative of documents. We also give benefit to the term with a higher NTF. Compare two tables in Table 1.The rules are not optimal and so we can get a better rule set by changing some parts of rule tables like those in [8]. In this paper, we didn't try to find optimal one because the rules in Table 1 are enough to show merits of our method.

2.2 Selection of initial representative keywords

After the weights of candidate terms are calculated through fuzzy inference, we need prioritize them to select IRKs. We could simply select candidate terms with a higher weight than a threshold as IRKs. However we observed that the performance of the approach heavily depended on the threshold value and the performance was also not good especially in the case that some documents do not contain any IRK. So, we impose a constraint that each example document should contain at least one or more initial representative keywords. The algorithm for selection of IRKs is shown in Fig. 3.

Procedure get_ITS(DS, TS)

DS: Example Documents Set (input)
TS: Candidate Terms Set (input)
ITS: Initial Representative Terms Set, initialized to empty. (output)
TS': Temporary Terms Set, initialized to TS.
d, t: Document and Term element respectively.

1] Repeat
2] Select a document element as d from DS.
3] Repeat
4] Select the highest element as t in TS' according to the weight.
5] If t appears in d and not member of ITS then add t to ITS.
6] Remove t from TS.
7] Until t appears in d.
8] Remove d from DS.
9] Assign TS to TS'.
10] Until DS is empty.
11] Return ITS.

Fig. 3. The algorithm for selection of IRKs

2.3 Term expansion and reweighting of representative keywords

The final representative keywords (FRKs) come from IRKs by expanding IRKs and reweighting them. If 5 terms are required to represent a user's preference and the number of IRKs is 3, then 2 terms with highest weights except IRKs are selected additionally. Once the FRKs are obtained, they are reweighted by a relevance feedback technique. Several relevance feedback techniques [16] have been proposed in the literature of information retrieval. Among them, we thought the one showing a good performance, if not the best, is enough to demonstrate our idea. So in RKEF method our previous work [7] is used to reweight FRKs because it's already implemented and its performance is better than classical ones such as Rocchio and Ide [7].

In [7], the relevance degree is used to determine co-occurrence similarity between a candidate term and query terms. In our work, IRKs are treated as query terms. Thus the relevance degree of a term is calculated by

$$RD_{ik} = 1 - \log_p \sqrt{\frac{\sum_{j=1}^{n} (\overline{tf}_{j,k} - tf_{i,k})^2}{n}} + 1 \qquad (2)$$

where, RD_{ik} is the relevance degree between the set of IRKs and candidate term t_i in a document d_k ; $\overline{tf}_{j,k}$ is the frequency of IRK t_j in a document d_k ; $tf_{i,k}$ is the frequency of candidate term t_i in a document d_k ; n is the number of IRKs, p is a control parameter. In this work, p is set to 10. The RD_{ik} is

treated as 0 if it has a negative value. The equation gives merit to those terms which are collocated with IRKs.

After calculating the relevance degree of FRKs, Equation 3 is used to compute their weights in the set of feedback documents.

$$w_i^{'} = \sum_{k \in C} (x_{ik} \times RD_{ik}) \tag{3}$$

where, $w_i^{'}$ is the weight of FRK t_i in the feedback document set; x_{ik} is the weight of term t_i in a document d_k; C is the set of positive feedback documents.

Finally, the final weight of FRK t_i, $\overline{w_i}$, is calculated by

$$\overline{w_i} = w_i + w_i^{'} \tag{4}$$

where, w_i is the initial weight of term t_i and calculated by the following.

$$w_i = (0.5 + \frac{0.5 \times freq_i}{\max_j freq_j}) \times \log(\frac{N}{n_i}) \tag{5}$$

where, frq_i is the frequency of term t_i in the query (i.e. the set of IRKs); n_i is the frequency of documents where t_i appears; N is the total number of documents.

3. The performance evaluation when RKEF is used as a linear classifier

Our previous work [8] is similar to constructing representative keywords of a set of example documents in the field of automatic text classification, especially linear approaches among various document classification methods – decision tree, decision rule, neural networks, Rocchio, Widrow-Hoff, k-NN, GIS, and SVM [5, 12, 13, 18, 20, 22]. Thus, we, here, review two representative linear classification algorithms, Rocchio and Widrow-Hoff, before evaluating our method over the two methods.

3.1 Linear classifier

In IR systems, a text is generally represented as a feature vector, $x = (x_1, x_2, ...x_d)$, where x_j is the weight value that feature j takes on for this document, and d is the number of features. For example, d might be the number of distinct non-stop words in a document collection and x_j be the frequency of a specific word in this document.

In order to rank documents, a document retrieval system typically applies a d-ary function f to each vector x, producing a score $f(x)$. Documents with the largest values of $f(x)$ rank at the top of retrieval results. A text categorization system might similarly compute scores $f(x)$ and assign a category only to those documents. The functions are linear, that is, they can be represented as the dot product of a weight vector w and the feature vector x [12]:

$$f(x) = w \cdot x = \sum_{j=1}^{d} w_j x_j \tag{6}$$

Rocchio and Widrow-Hoff linear classifiers among many classifiers train corpus of training documents and derive the weight vectors or centroid vectors to classify new documents correctly.

3.1.1 Rocchio classifier

Rocchio classifier [12, 18] is based on the relevance feedback algorithm originally used for the vector space retrieval model. It has been extensively used for document classification. The Rocchio algorithm operates in batch mode and produces a new weight vector w from an existing weight vector w_1 and a set of training examples:

$$w_i = \alpha w_{1,i} + \beta \frac{\sum_{j \in C} x_{i,j}}{n_C} - \gamma \frac{\sum_{j \notin C} x_{i,j}}{n - n_C} \tag{7}$$

where $x_{i,j}$ is the weights of term i in document j, n is the number of training examples, C is the set of positive training examples, and n_C is the number of positive training examples. The parameters α, β, γ control the relative impact of the initial weight vector, the positive examples, and the negative examples, respectively. If $\alpha = 0$, $\beta = 1$, and $\gamma = 1$, then the difference in the mean of weight vectors for positive and negative training instances is calculated as the weight vector of specific features for a document collection.

Classifiers produced with the Rocchio algorithm are restricted to having nonnegative weights, so that instead of using the raw w from Equation 7, we use the following $w^{'}$ where

$$w_i^{'} = \begin{cases} w_i & \text{if } w_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

3.1.2 Widrow-Hoff classifier

Widrow-Hoff classifier is an online method since it runs through the training examples one at a time updating a weight vector at each step [12]. The weight vector is initially set to all zeros vector, $w_1 = (0,0,...,0)$. At each step, the new weight vector w_{i+1} is calculated from the old weight vector w_i using training example x_i with label y_i. The class label y_i is 1 if a training document x_i is in the set of positive or relevant training documents, otherwise 0. Since the term $2(w \cdot x - y)x$ in Equation 9 is the gradient of the squared error $(w \cdot x - y)^2$, the Widrow-Hoff algorithm tries to move in a direction in which this error is decreasing.

$$w_{i+1,j} = w_{i,j} - 2\eta(w_i \cdot x_i - y_i)x_i \tag{9}$$

where, the learning rate η controls how quickly the weight vector w_i is allowed to change.

3.2 Experimental Evaluation

Since the RKEF method is originally proposed for the case of a few positive feedback documents, we first compare to the Rocchio and Widrow-Hoff algorithms for a few positive feedback documents. And then the experiments are performed for the case with many positive feedbacks.

As we know, a different number of FRKs has different representative power. In [8], to find out the optimal number of FRKs when a few positive feedback documents are given, a series of experiments were conducted increasing the number of FRKs from 5 to 30 by 5. The result showed that RKEF method is better than the others in all cases, especially when 10 FRKs are used. However, in this work, we use all candidate terms as FRKs for just simplicity.

3.2.1 Performance evaluation considering a few positive feedback documents

Experimental environment. We used Reuters-21578 data as our experimental document set [1]. In this paper, TOPICS category set having 135 categories of Reuters-21578 in which unlabelled documents were previously eliminated is selected. We divide the documents according to the "ModeApte" split. Among the 135 categories, we choose 90 ones including at least one training example and one testing example. Then, we finally select 21 categories that have from 10 to 30 training documents. The 3019 documents of those categories are used as testing documents. The document frequency information from 7770 training documents in 90 categories is used to calculate IDF of terms. We exclude negative documents under the assumption that users generally present positive documents coincident with their preferences.

The weight vectors of terms in the two methods compared with RKEF are calculated by TF × IDF [12]. Control parameters are set to the following values in this experiment: α=0, β=1, γ=0 in Rocchio algorithm, η=0.25, y_i=1 (positive documents) in Widrow-Hoff algorithm [12] and p=10 in Equation 2 of Section 2.3 in our method.

Documents are ranked by the cosine similarity [2, 20] with the FRKs. We used the average precision values derived through an interpolation procedure for 11 standard recall levels as a classification measure [2, 18, 23].

Experimental results. Table 2 shows the average result of the RKEF compared to the two existing classifiers for 21categories. As shown in Table 2, the RKEF method outperforms the two ones. Please note that a few documents having from 10 to 30 training documents are used. RO represents the Rocchio classifier and WH the Widrow-Hoff algorithm, respectively in Table 2.

Table 2. Performance of 21 categories in the REUTERS corpus and comparison with two existing classifiers

Category	11 points average precision		
	RO	WH	RKEF
lumber	0.346	0.354	0.550
dmk	0.044	0.042	0.084
sunseed	0.376	0.375	0.451
lei	0.273	0.273	0.363
soy-meal	0.539	0.447	0.772
fuel	0.429	0.436	0.518
soy-oil	0.185	0.185	0.323
heat	0.483	0.480	0.626
lead	0.556	0.557	0.614
housing	0.373	0.373	0.352
strategic-met	0.127	0.137	0.120
hog	0.513	0.533	0.485
orange	0.933	0.933	0.975
tin	0.959	0.966	0.986
rapeseed	0.443	0.428	0.575
wpi	0.764	0.708	0.728
pet-chem	0.405	0.482	0.308
silver	0.377	0.508	0.770
zinc	0.880	0.799	0.921
retail	0.030	0.024	0.194
sorghum	0.489	0.342	0.591
Average	0.454	0.447	0.538

3.2.2 Performance evaluation considering a set of many positive feedback documents

We found that the proposed RKEF method yielded good results in the case of positive documents from 10 to 30 through the previous experiment in Section 3.2.2. Here, we would like to do a performance evaluation when more positive feedback documents are given than the previous experiment.

Experimental environment. Values of control parameters are the same as the previous experiment. We choose upper 20 categories including lots of training examples among 90 ones explained in the previous work.

Experimental results. Table 3 shows the average result of the RKEF compared to the two conventional classifiers for upper 20 categories having lots of training examples. As shown in Table 3, the proposed method yields a little better performance than the other two. This result indicates that a large document set need to be clustered and then the RKEF should have been applied to each group of a few documents.

Table 3. Performance of upper 20 categories with lots of training examples

Category	11 points average precision		
	RO	WH	RKEF
nat-gas	0.492	0.494	0.591
soybean	0.639	0.589	0.708
veg-oil	0.626	0.630	0.625
gold	0.855	0.843	0.831
gnp	0.816	0.820	0.888
coffee	0.936	0.979	0.951
oilseed	0.483	0.425	0.434
sugar	0.739	0.776	0.720
dlr	0.636	0.686	0.698
money-suppl	0.334	0.587	0.697
corn	0.644	0.624	0.677
ship	0.822	0.745	0.781
wheat	0.764	0.798	0.802
interest	0.636	0.720	0.641
trade	0.717	0.661	0.705
crude	0.778	0.801	0.791
grain	0.802	0.871	0.837
money-fx	0.582	0.537	0.613
acq	0.576	0.727	0.718
earn	0.961	0.948	0.908
Average	0.692	0.713	0.731

4. The performance evaluation when RKEF is used as a generalization function of GIS

In the previous section, we observe that RKEF can improve the document classification performance when it's used as a linear classification method. Here, we show how RKEF is applicable to non-linear document classification and give performance evaluation for that case.

Grouping documents into several clusters, we can apply RKEF method to each cluster of documents. For this type of experiment, GIS approach [10, 11, 18] is used because it is well known in text categorization and RKEF method can be used in generating generalized instances.

4.1 GIS (Generalized Instance Set) classifier

GIS algorithm unifies the linear classifiers and the k-NN (K-Nearest Neighbor) classifier [10, 11]. The k-NN algorithm is applied to Expert Network (ExpNet) for text categorization [20]. In a k-NN classifier, it does not directly generate the weight vectors from training examples like linear classifiers. On the contrary, the cosine similarity $sim(X,D_j)$ between each training document D_j and the request document X is calculated. The training examples are sorted by the cosine similarity in descending order and then the k top-ranking documents are selected. The final score of the request document to each category is computed from considering the similarity metric of these k chosen documents and their category association.

The k-NN algorithm shows good performance without constructing document model (or representative vector). But it's too sensitive to noisy data. So, in GIS, generalized instances (GIs) are used instead of original documents to overcome the weakness of k-NN classifier. GIs are generated by a generalization function which groups documents into several clusters and builds a model for each cluster. We call the set of models GIS. Various algorithms can be used for the generalization task. In [10], two linear classifiers, the Rocchio and Widrow-Hoff algorithm, are used to construct a set of GIs to replace the original training instances.

After generalized instances are generated, documents can be classified using Equation 10 similar to the one used in the k-NN by regarding these GIs as training documents. $Assoc(G,C)$ is defined as the association factor between a generalized instance G and the category C.

$$Score(X,C) = \sum_{G \in GS} Sim(G,X) \times Assoc(G,C) \qquad (10)$$

$$Assoc(G,C) = \frac{P_k}{P}$$

where G is a generalized instance, GS is the generalized instance set, P is the number of positive documents in the category C among the training set, P_k is the number of documents in k nearest neighbors of G among positive documents in the category C. $Assoc(G,C)$ represents a measure how much a set of positive documents in the category C contribute to construct a generalized instance G. For example, $Assoc(G,C)$ is 1 if every document relating to construct a generalized instance G is assigned to the category C. On the other hand, if no document in the category C contributes to construct a generalized instance G, then $Assoc(G,C)$ is 0.

The category of the new document X is determined based on its value of *Score* function. Namely, if the score is greater than a threshold, category C is assigned to document X. In other words, GIS classifier is a method applying k-NN classifier to a generalized instance set instead of original training documents.

4.2 Experimental Evaluation

The RKEF method was initially proposed under the assumption that only positive feedback documents were given. Regardless of this assumption, one may have a question on the performance of RKEF when it was applied to document sets containing negative documents. So, here, we first give the performance evaluation when only positive documents are

considered and then when negative documents are also considered.

4.2.1 Performance evaluation considering only positive documents

Experimental environment. Values of control parameters are the same as the previous experiment. We choose upper 20 categories including lots of training examples among 90 ones explained in the previous work. We apply Rocchio, Widrow-Hoff, and the proposed algorithms as a generalization function. We carry out a series of experiments increasing the number of k used in the generalization function from 10 to 150 by 10. In GIS algorithm, the result of clustering is controlled by not the number of clusters but the number of documents in a cluster, say k.

Experimental results. The best results of different combinations for 15 values of k are given in Table 4, where GIS-RO, GIS-WH and GIS-RKEF represent a method to combine GIS algorithm and the Rocchio's, the Widrow-Hoff's, and the RKEF classifier respectively. As shown in Table 4, the RKEF method yields more improved performance than the others when only positive feedback documents are used in the generalization function.

Table 4. Performance comparison when only positive feedback documents are used in the generalization function

Category	Best		
	GIS-RO	GIS-WH	GIS-RKEF
nat-gas	0.518	0.599	0.643
soybean	0.642	0.654	0.738
veg-oil	0.657	0.651	0.756
gold	0.861	0.863	0.846
gnp	0.831	0.835	0.871
coffee	0.947	0.936	0.989
oilseed	0.497	0.508	0.601
sugar	0.793	0.807	0.882
dlr	0.719	0.726	0.751
money-suppl	0.624	0.607	0.726
corn	0.658	0.654	0.797
ship	0.831	0.821	0.854
wheat	0.808	0.803	0.861
interest	0.731	0.738	0.793
trade	0.733	0.740	0.749
crude	0.809	0.808	0.846
grain	0.859	0.866	0.867
money-fx	0.616	0.615	0.663
acq	0.707	0.708	0.792
earn	0.963	0.962	0.962
Average	0.740	0.745	0.799

4.2.2 Performance evaluation considering negative documents

In GIS algorithm, training documents are grouped into several clusters and a generalization function is applied to each cluster to get the general instance (or representative vector). So, a cluster usually contains not only positive examples but also negative ones, which makes us hard to apply RKEF to all example documents in a cluster. Therefore, in this work, we apply RKEF method to positive examples and negative examples separately and then combine them by the following equation:

$$w_i^C = \alpha \overline{w}_i^P - \beta \overline{w}_i^N \tag{11}$$

where, P is the set of positive feedback documents included in the given cluster C; N is the set of negative documents; $\overline{w}_i^P$ and $\overline{w}_i^N$ represent the weights of FRKs extracted by RKEF method from P and N, respectively; w_i^C is the weight of term t_i in the cluster C.

Experimental environment. Experimental domain is the same as the experiments in Section 4.2.1. Since the previous experiments are performed based on the positive documents only, α=0, β=1, γ=0 in the Rocchio algorithm and η=0.25 in the Widrow-Hoff algorithm are used as control parameter values. So, we change the parameter values for negative documents. That is, α=0, β=1, γ=1 are used in the Rocchio, η=0.25 in the Widrow-Hoff, and α=1 and β=1 in Equation 11.

Experimental results. The best ones among experimental results with 15 values of k for each generalization method based on positive and negative documents are given in Table 5. In Table 4, the RKEF method yield more improved performance than the others when only positive feedback documents are used in the generalization function of the GIS algorithm. On the other hand, GIS-RKEF does not show better performance than the GIS-RO when considering negative feedback documents together. Such results might be caused by applying the fuzzy inference rules and membership functions designed only for positive feedback documents to the negative documents without any modification.

Table 5. Average comparison when both of positive and negative documents are used in the generalization function (GIS-RO, GIS-WH, GIS-RKEF)

Category	Best		
	GIS-RO	GIS-WH	GIS-RKEF
nat-gas	0.723	0.694	0.667
soybean	0.780	0.740	0.779
veg-oil	0.739	0.716	0.701
gold	0.862	0.866	0.853
gnp	0.932	0.915	0.930
coffee	0.988	0.991	0.985

oilseed	0.663	0.665	0.613
sugar	0.910	0.914	0.917
dlr	0.805	0.777	0.787
money-suppl	0.726	0.725	0.727
corn	0.898	0.857	0.901
ship	0.880	0.866	0.876
wheat	0.893	0.874	0.929
interest	0.803	0.790	0.802
trade	0.788	0.770	0.810
crude	0.880	0.838	0.892
grain	0.937	0.960	0.946
money-fx	0.694	0.686	0.688
acq	0.877	0.822	0.875
earn	0.967	0.966	0.964
Average	0.837	0.819	0.832

5. Conclusions

In this paper, we conducted a series of experiments for verifying RKEF method that extracts important keywords from a few positive example documents by fuzzy inference and relevance feedback techniques. We observed that the method still has merits over some representative linear approaches in text classification such as Roccio and Widraw-Hoff when applying the method to the larger document sets. We also observed that the performance of the method is increased at somewhat extent when combined with GIS algorithm, which means the method shows benefits for a small set of positive feedback documents as originally designed. When negative example documents are considered together with positive ones, the performance was not good as expected, which leads us the partial conclusion that the fuzzy inference rules and membership functions do not work well for negative documents because they are designed only considering positive documents. It also imposes the possibility of performance improvement of RKEF method if appropriate rules and membership functions are designed for negative example documents.

In information retrieval environment, a user judges the relevance of one or more of retrieved documents and these judgments are fed back to the system to improve the initial search result [17]. Buckey et al. [3] experimentally verified that the recall-precision effectiveness is roughly proportional to the log of the number of known relevant documents. In other words, the greater the amount of feedback from the user to the system, the better is the search effectiveness of the system. However, this expectation is often not met in Web information retrieval because most of users are reluctant to provide hundreds of relevance judgments. They give relevance judgments for a few documents, for example, around 10 documents. So, RKEF method will be useful in capturing user preference for that case.

References

[1] http://www.research.att.com/~lewis/reuters21578.html

[2] R. Baeza-Yates andB. Ribeiro-Neto, *Modern Information Retrieval*, ACM Press, NY, USA, 1999.

[3] C. Buckley and G. Salton, "Optimization of relevance feedback weights," *Proc. of 18th Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, 1995.

[4] D. Goldberg, D. Nichols, B. M. Oki and Douglas Terry, "Using Collaborative Filtering to Weave an Information Tapestry. Commun," *ACM*, 35, 1992.

[5] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *Proc. of European Conference on Machine Learning*, pp. 137-142, 1998.

[6] J. Kim, D.W. Oard and K. Romanik, "User modeling for information filtering based on implicit feedback," *Proc. of ISKO-France*, 2000.

[7] Byeong Man Kim, Ju Youn Kim and Jongwan Kim, "Query Term Expansion and Reweighting using Term Co-Occurrence Similarity and Fuzzy Inference," *Proc. of IFSA/NAFIPS*, pp.715-720, 2001.

[8] Byeong Man Kim, Qing Li, Kwang-Ho Lee and Bo-Yeong Kang, "Extraction of Representative Keywords Considering Co-occurrence in Positive Documents," *FSKD 2005 : Fuzzy Systems and Knowledge Discovery*, Lipo Wang and Yaochu Jin, Eds., LNAI 3614, Springer-Verlag, pp. 752-761, 2005.

[9] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L.R. Gordon and J. Riedl, "GroupLens: Applying collaborative filtering to Usenet News," *CACM*, 40(3), pp. 77-87, 1997.

[10] K. Lam and C. Ho, "Using a generalized instance set for automatic textcategorization," *Proc. of 21th Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 88-89, 1998.

[11] Kwok-Yin Lai and Wai Lam, "Automatic Textual Document Categorization Using Multiple Similarity-Based Models," *Proc. of SDM01*, 2001.

[12] D.D. Lewis, R.E. Schapore, J.P. Call and R. Papka, "Training algorithms for linear text classifiers," *Proc. of 19th Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 298-306, 1996.

[13] Tom M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

[14] D. M. Nichols, "Implicit ratings and filtering," *Proc. of the 5th DELOS Workshop on Filtering and Collaborative Filtering*, pp. 10-12, 1997.

[15] M. Pazzani and D. Billsus, "Learning and revising user profiles: the identification of interesting Web sites," *Machine Learning*, 1997.

[16] Ian Ruthven and M. Lalmas, "A survey on the use of relevance feedback for information access systems," *Knowledge Engineering Review*, 18 (2), pp. 95 – 145, 2003.

[17] R. Schapire, Y. Singer and A. Singal, "Boosting and Rocchio Applied to Text Filtering," *Proc. of 21th Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.

[18] F. Sebastiani, "Machine Learning in Automated Text," *Technical Report IEI-B4-31-1999*, Istituto di Elaborazione dellInformazione, 1999.

[19] Y. Seo and B. Zhang, "Personalized Web Document Filtering Using Reinforcement Learning," *Applied Artificial Intelligence*, 2001.

[20] Y. Yang, "Expert network: effective and efficient learning from human decisions in text categorization and retrieval," *Proc. of 17th Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994.

[21] T. W. Yan and H. Garcia-Molina, "SIFT- A tool for wide-area information dissemination," *Proc. of the 1995 USENIX Technical Conference*, 1995.

[22] Y. Yang and X. Liu, "A re-examination of text categorization methods," *Proc. of 22nd Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

[23] Y. Yang, "An evaluation of statistical approaches to text," *Journal of Information Retrieval*, pp. 67-88, 1999.

Byeong Man Kim
received the BS degree in Dept. of computer Engineering from Seoul National University, Korea in 1987, and the MS and the PhD degree in computer science from Korea Advanced Institute of Science and Technology, Korea in 1989 and 1992, respectively. He has been with Kumoh National University of Technology since 1992 and is currently a professor. He worked on distributed Web agents at Computer Science Department, Colorado State University, USA as a visiting researcher from 2005 - 20006. His current research areas include multimedia information retrieval, context-aware recommending system, data mining, and Web agents.

Phone : +82-54-478-7544
Fax : +82-54-478-7539
E-mail : bmkim@kumoh.ac.kr

Sun Ok Rho
received the BS and the MS degree in Dept. of Computer Science from Kumoh National Institute of Technology in 1999, and 2001, respectively. He has written several papers in the areas of artificial intelligence and information filtering. His current research areas include text mining and user modeling.

Phone : +82-54-478-7567
Fax : +82-54-478-7539
E-mail : sorho@se.kumoh.ac.kr

Sangyeop Oh
received the BS in Dept. of Physics, the MS and the PhD degree in Dept. of Computer Engineering from Korea Advanced Institute of Science and Technology in 1992, 1994, and 2001, respectively. He was a director of Search Solutions, Inc., Korea from 2001 - 2002 and a visiting researcher at Electrical Engineering and Computer Science Department of University of Michigan, USA from 2002 - 2003. He had been with Kumoh National Institute of Technology from 2004 to 2007 as an assistant professor. His current research areas include intelligent information retrieval and social networking.

Phone : +82-16-670-2928
Fax : +82-54-478-7539
E-mail : syoh@kumoh.ac.kr, neocella@naver.com

Hyun Ah Lee
received her M.S. degree in Computer Science in 1998 and Ph.D. degree in 2004 both from Korea Advanced Institute of Science and Technology(KAIST). She worked at Daum Soft, Inc. as a senior researcher during 2000~2004, and is a professor at the School of Computer and Software Engineering at Kumoh National Institute of Technology from 2004. Her research interests are in automatic knowledge extraction, knowledge engineering, machine translation, and natural language processing.

Phone : +82-54-478-7546
Fax : +82-54-478-7539
E-mail : halee@kumoh.ac.kr

Jong Wan Kim

received the BS, the MS, and the PhD degree in Dept. of Computer Engineering from Seoul National University, Korea in 1987, 1989, and 1994, respectively. He has been with Daegu University since 1995 and is currently a professor. From 2006 - 2007, he was a visiting professor at Computer and Information Science Department of University of Oregon, working on the user preference ontology based anti-spam systems with the partial support of KRF. He has written several papers in the areas of information filtering, fuzzy systems, and anti-spam systems. His current research areas include artificial intelligence and data mining.

Phone : +82-53-850-6575
Fax : +82-53-850-6589
E-mail : jwkim@daegu.ac.kr