9-2002

Personalized classification for keyword-based category profiles

Aixin SUN
Nanyang Technological University

Ee Peng LIM
Singapore Management University, eplim@smu.edu.sg

Wee-Keong NG

DOI: https://doi.org/10.1007/3-540-45747-X_5

Proceedings of 6th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2002), Rome, Italy, Sep, 2002

Personalized Classification for Keyword-based Category Profiles *

Aixin Sun, Ee-Peng Lim, and Wee-Keong Ng

Centre for Advanced Information Systems
School of Computer Engineering
Nanyang Technological University, Singapore
{p140971904, aseplim, awkng}@ntu.edu.sg

Abstract. Personalized classification refers to allowing users to define their own categories and automating the assignment of documents to these categories. In this paper, we examine the use of keywords to define personalized categories and propose the use of Support Vector Machine (SVM) to perform personalized classification. Two scenarios have been investigated. The first assumes that the personalized categories are defined in a flat category space. The second assumes that each personalized category is defined within a pre-defined general category that provides a more specific context for the personalized category. The training documents for personalized categories are obtained from a training document pool using a search engine and a set of keywords. Our experiments have delivered better classification results using the second scenario. We also conclude that the number of keywords used can be very small and increasing them does not always lead to better classification performance.

1 Introduction

Text classification refers to the process of assigning documents to suitable predefined categories. In the context of World Wide Web, the text classification problem becomes much more difficult due to several reasons. Firstly, in terms of problem size, classification of text documents on the web has to deal with huge number of documents and users. Efficient classification methods are therefore necessary. Secondly, text classification methods have to deal with documents with diverse content and users with diverse interests. The traditional assumption of fixed pre-defined categories clearly cannot cater for all user interests. In this paper, we therefore focus on classifying documents according to diverse user interests by introducing *personalized classification*. In other words, we would like the users create their own personalized categories and the classifiers for these categories will be automatically constructed for classifying documents under such categories. Personalized classification is clearly useful in many applications such as online news classification, new book recommendation, and others.

To develop a personalized classification method, we have to answer the following questions:

* The work is partially supported by the SingAREN 21 research grant M48020004.

- *How is a personalized category defined?*
  In the traditional text classification problem, a category is defined with a set of training documents. This assumption can no longer hold for personalized classification as it is not possible for users to painstakingly select adequate number of training documents for their personalized categories. If training documents cannot be directly given, what are the other alternatives for users to define personalized categories? In this work, we will study the possibility of defining personalized categories using keywords since it is much simpler for users to select keywords for their personalized categories.

- *How can the classifier for a personalized category be constructed?*
  Since training documents are not available at the point a personalized category is defined, it is therefore necessary to obtain a set of good training documents for the construction of classifier based on the earlier user input (or keywords). Once the training documents are derived, one can apply different techniques to construct the classifier for the personalized category.

- *How can a personalized classification method be evaluated?*
  The effectiveness of classification methods can be determined by experiments. For the evaluation of personalized classification, it is necessary to determine the appropriate experimental setup and performance measures for comparing the different classification methods.

- *How can the changes to the user interest affect a personalized category and its classifier?*
  As a user changes his or her interest domain, the corresponding personalized category will be affected. In this case, feedback from users will be critical as they allow the personalized classifiers to be revised. The amount of feedback and their frequency are important issues to be addressed. Furthermore, the evaluation of such personalized classification methods with feedback mechanism will require different kinds of experiments.

Among the above research issues, we have chosen to address mainly the first, second and third in this paper. We have also divided the personalized classification process into 4 distinct steps, namely:

1. Definition of personalized category profile, where the profile refers to the information supplied by a user to create the category.
2. Selection of training documents.
3. Construction of classifiers.
4. Classification of documents.

These steps are similar to *non-adaptive batch filtering task* as defined by the TREC competition [14]. We will elaborate further on this in Section 2.

Our work is unique in the use of keywords to define personalized category profiles. In most previous classification work, a category profile is often defined by a set of training documents provided by the user. To minimize user efforts, we have instead chosen keyword-based category profiles. Personalized classification using keyword-based category profiles has several distinct advantages. Apart from little user efforts, the keywords give some opportunity to derive a larger set of training documents for the construction of classifiers, instead of being restricted by the small number of training documents given by the user.

In this paper, two approaches of personalized classification for keyword-based category profiles are introduced, namely *flat* and *hierarchical* personalized classification. The former assumes each personalized category is defined independently while the latter explores the possibility of having the personalized categories defined under some pre-defined general categories. The classification methods using SVM classifiers for the two approaches have also been developed. Our experiments on the two approaches have shown that the hierarchical personalized classification outperforms the flat one significantly. We also found out the number of keywords used can be very small and increasing them does not always lead to better classification performance.

This paper is organized as follows. In Section 2, research related to personalized classification are discussed. Our proposed personalized classification processes are described in Section 3. The experimental are described in Section 4 and our results are reported in Section 5. Finally, in Section 6, we conclude our paper and give some pointers to our future research.

2 Related Work

Personalized classification based on document content is considered as a type of *text filtering*, an information seeking process in which documents are selected from a dynamic text stream to satisfy a relatively stable and specific information need [10, 11]. In text filtering, one or more set of features each representing a different user interest domain is first derived. Text documents are then retrieved based on their semantic similarity with each set of features. Text filtering techniques have been well-studied in the Text REtrieval Conference (TREC) [14].

TREC classifies filtering task into three types, namely, *adaptive filtering*, *batch filtering* and *routing* [12]. In adaptive filtering, a user supplies a small set (e.g. 2 or 3) of training documents relevant to his/her interest. The decisions of whether the newly coming documents are relevant to the user will be determined immediately upon the documents' arrival. For each retrieved document, the category profile is updated with the feedback given by the user. In this way, filtering decisions can be improved over time. In batch filtering and routing, the system may start with a large set of training documents to construct the category profile. For each newly arrived document, batch filtering involves deciding if the document is relevant or not. In the case of routing, the given document is ranked among other newly arrived documents. When a batch filtering method can update its category profile by collecting user feedback, it is known to be *adaptive*. Much work in text filtering has been reported in TREC [12]. The classifiers involved include k-Nearest-Neighbor (kNN) classifier [1], incremental Rocchio classifier [17] and SVM classifier [7].

Ault and Yang applied the multi-class kNN algorithm to batch and adaptive filtering tasks in TREC-9 [1]. The kNN classifier represented each document x as a vector $\boldsymbol{x}$, and computed the relevance to a category C, $s(C, \boldsymbol{x})$. In their work, different approaches to transform relevance score into a binary decision on assigning that document to the category were discussed in detail. The main difficulty in applying kNN classifier to filtering was the determination of the k value.

Lee *et al.* proposed a batch filtering method combining query zone (QZ), SVM, and Rocchio techniques [7]. The Rocchio's algorithm was used to construct category profiles, while the query zone technique was used to provide negative training documents to the SVM classifier. The final decision of whether a document was relevant or not is dependent on the voting results between SVM output and the profile-document similarity with thresholds. However, the performance of this hybrid filtering method was not promising compared to that based on SVM classifiers only.

At present, our personalized classification problem is similar to batch filtering as they both require a set of training documents for constructing their classifiers. However, our personalized classification methods simplify the user efforts by adopting keyword-based category profiles. The main challenge in our method is in the derivation of appropriate training documents for any personalized categories. To our best knowledge, there has not been research conducted to evaluate personalized classification (or batch filtering) using keyword-based categories. On the other hand, our proposed methods have not considered the possibility of user feedback. Our proposed methods can therefore be treated as non-adaptive batch filtering. In our experiments, we have therefore adopted the performance measures used in the TREC's filtering track. User feedback is a powerful mechanism to improve the accuracy of classification. The extension of our proposed methods to handle user feedback will be part of our future work.

3 Personalized Classification Processes

3.1 Flat Personalized Classification

Personalized classification using keyword-based category profiles requires users to provide a few keywords for each personalized category profile. When the personalized categories are defined independently of one another, the category space is said to be *flat* (i.e., there is no structural relationship defined among the categories), and the corresponding classification method is known as *flat personalized classification method (FPC)*.

Our proposed flat personalized classification process is shown in Figure 1. For each personalized category to be defined, a category profile needs to be created based on the user supplied keywords. Using these keywords, a search engine ranks all the documents from a training document pool according to their relevance to the category profile. Here, we assume that a fairly large pool of training documents is available and from it we are able to find some training

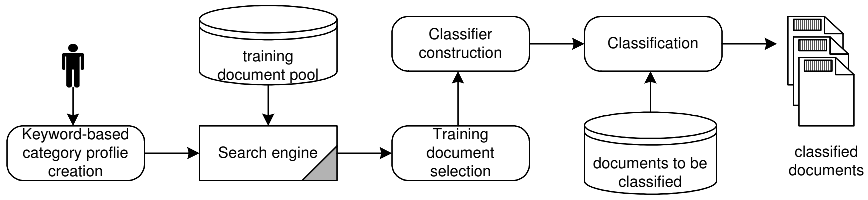

Fig. 1. Flat personalized classification process

documents for the personalized classifiers. Although this assumption may not be applicable in some situations, it is nevertheless a reasonable assumption for many real-life applications, e.g. personalized classification for online news, technical reports, etc.. Using the ranks (and scores) provided by the search engine, training documents are further selected and used in the construction of personalized classifiers, i.e. the classifiers for personalized categories. Each personalized category will be associated with a personalized classifier that determines if new documents should be assigned to the category.

3.2 Hierarchical Personalized Classification

As the number of keywords provided for each personalized category is usually very small, they may not always be able to characterize the category very well. A straightforward extension is to allow the personalized categories to be defined within some pre-existing general categories that provide the broad context. For example, within the *computing* general category, one may be able to define a personalized category for documents related to mobile devices. In this way, documents that are related to non-computing mobile devices will be excluded from the personalized category. The general categories are pre-defined and and therefore, good sets of training documents can be easily obtained for them. When each personalized category is defined within some general category, we say that the corresponding classification method a *hierarchical personalized classification method (HPC)*.

Our proposed hierarchical personalized classification process is shown in Figure 2. In this process, there are two kinds of classifiers to be constructed, one for the general categories and another for the personalized categories. We call them the *general classifiers* and *personalized classifiers* respectively. Hence, each general and personalized category will be associated with a general classifier and personalized classifier respectively. We currently assume that the training documents of the general categories are determined manually. On the other hand, the training documents of the personalized categories will be obtained from the training documents of the corresponding general categories with the search engine using the user-provided keywords.

Given a document to be classified, it is first classified by the general classifiers. Only when the document is assigned to a general category, it will be

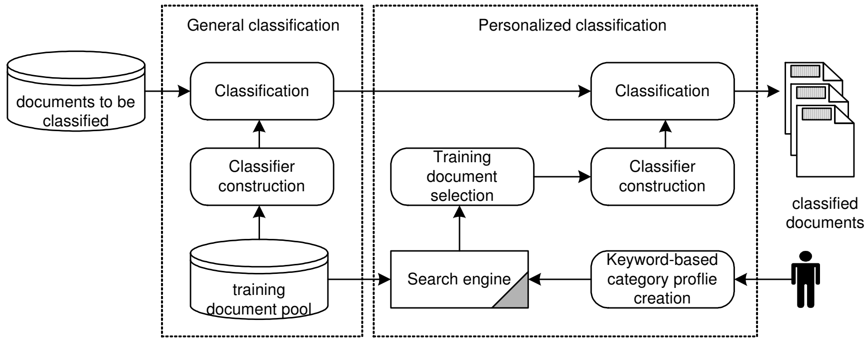

Fig. 2. Hierarchical personalized classification Process

further classified by the personalized classifiers associated with the personalized categories under the general category.

4 Experimental Setup

4.1 Data Set

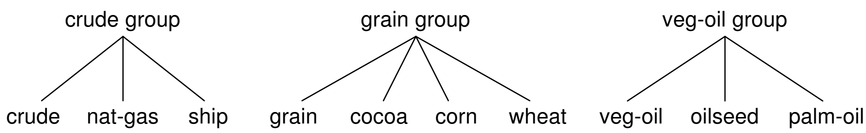

Fig. 3. The general and personalized categories used in the experiments

In our experiment, Reuters-21578 news collection[1] was used. The Reuters corpus is one of the most popular data sets used in text classification [15]. The 21578 documents in this collection are organized into 135 categories. Each document may have zero, one or more categories labelled to it. Since the Reuters' categories are not organized in a hierarchical manner, we manually derived 3 hierarchies from the 135 categories similar to the ones used by Kohler and Sahami [6] (see Figure 3). The 10 personalized categories are grouped under the 3 general categories (i.e., *crude group*, *grain group*, and *veg-oil group*).

We used *Lewis Split* provided by the Reuters collection to obtain training documents and test documents. The pool of training documents included all training documents (i.e., LEWISSPLIT="TRAIN") of the 10 personalized categories. In addition, equal number of documents not belonging to the any of the 10 categories were also added to the pool to serve as the "noises" to the

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

search engine. A total of 2314 training documents were included in the pool. The test document set in our experiments included all the test documents (i.e., LEWISSPLIT="TEST") belonging to any of the 10 personalized categories and same number of test documents not belonging to the 10 personalized categories. There were in total 910 test documents used in our experiments. After stopword removal and stemming, a feature vector was obtained for each document without applying any other feature selection. The feature vector recorded the words that appear in the document and their frequencies. The stopword list and the stemming algorithm were taken directly from the BOW library [8].

4.2 Search Engine for Training Document Pool

Given the keyword-based category profiles, positive and negative training documents are selected from the training document pool for the construction of the personalized classifiers. In our work, a small-scale search engine is implemented to search the training documents in the pool.

The search engine computes for each training document its $tfidf$ rank based on the keywords provided. The $tfidf$ weight of a index term w_k in a document d_j is computed from the term frequency $Freq(w_k, d_j)$ and the inverse document frequency.

$$tfidf(w_k, d_j) = Freq(w_k, d_j) \times \log_2 \frac{N}{DF(w_k)} \tag{1}$$

where N is the number of documents in the given document collection, $DF(w_k)$ is the number of documents in the given document collection with the word w_k occurs at least once. The rank of a document is defined by the sum of the $tfidf$ weights of all the keywords in the personalized category profile denoted by *cp*.

$$rank(d_j) = \sum_{w_k \in cp} tfidf(w_k, d_j) \tag{2}$$

4.3 Flat Personalized Classification

In flat personalized classification, our experiments simulated the user-provided keywords for each of the 10 personalized categories in Figure 3 by using some feature selection techniques. In the following, we will describe the generation of keywords for personalized categories and the selection of training documents for the construction of personalized classifiers.

Generation of Keyword-based Category Profiles In our work, we would like a keyword-based category profile to consist of a small set of keywords that can describe or represent the content of the personalized category to be constructed. Intuitively, personalized classification will work optimally when the keywords provided by the user have high discriminatory power in distinguishing documents under a personalized category from those under the other categories. We assume that users are in a good position to define the appropriate keywords for their category profiles. This assumption is important as it was adopted by our experiments to *simulate* the user-chosen keywords generation using *feature selection* techniques.

A number of feature selection techniques have been described in the survey by Sebastiani [13]. Among them, document frequency and information gain (IG) are most commonly used. However, when a user wishes to construct a personalized category, he will most likely choose the ones that appear in the personalized category with the highest probability while lowest probability in the other categories. We therefore adopted *Odds Ratio* (OR) feature selection technique to generate the keywords for each personalized category. There are several variants of Odds Ratio. In our experiment, we use the exponential form due to its simplicity and excellent performance [9].

$$OR(w_k, C_i) = e^{P(w_k|C_i) - P(w_k|\overline{C_i})} \tag{3}$$

where $OR(w_k, C_i)$ is the Odds Ratio value (between e^{-1} and e) for the word w_k in category C_i, $P(w_k|C_i)$ is the conditional probability of w_k occurring in C_i and $P(w_k|\overline{C_i})$ is the conditional probability of w_k occurs in other categories (i.e., not in C_i). $P(w_k|C_i)$ can be easily calculated using word frequency as shown in Equation 4, where $Freq(w_k, C_i)$ is the times of occurrences of w_k in C_i. $P(w_k|\overline{C_i})$ can be calculated similarly by replacing C_i with $\overline{C_i}$ in Equation 4.

$$P(w_k|C_i) = \frac{Freq(w_k, C_i)}{\sum_{w_j \in C_i} Freq(w_j, C_i)} \tag{4}$$

In the case of flat personalized classification, each personalized category C_p consists of all training documents that belong to the category, and $\overline{C_p}$ consists of all training documents (see Section 4.1) that do not belong to the category. By selecting the words with the highest OR values, the keyword-based category profile of a personalized category is obtained. In our experiments, we evaluated the performance of personalized classification using top 1, 2, 3, 4, 5, 10 and 20 keywords.

Construction of Personalized Category Classifiers In our experiments, *Support Vector Machine (SVM)* classifiers were chosen for the classification tasks. SVM is good at finding the best surface that separates the positive and negative training examples at the widest margin, and has been successfully applied to text classification problems [3, 5]. Since SVM classifiers are binary classifiers, they need to be trained with both positive and negative examples. Our experiments used the SVM^{light} Version 3.50 package implemented by Joachims to construct the SVM classifiers [4].

To construct the personalized SVM classifier for a personalized category in Figure 3, we selected the top-ranked 50 training documents returned by the search engine (using the category keywords) as positive training documents and the bottom-ranked 50 documents as negative training documents. Unlike the document features used by the search engine in document ranking, the SVM classifiers were trained using the binary feature vectors of the training documents.

4.4 Hierarchical Personalized Classification

In hierarchical personalized classification, each personalized category is defined under some pre-defined general category and the choice of keywords for the personalized category profile will therefore be related to the context provided by the general category. Each general category has a well-defined set of training documents which are used in the construction of a general classifier. On the other hand, each personalized classifier is constructed using the training documents selected from the training documents of its general category. In the following, we will describe the generation of keyword-based category profiles, and the construction of general and personalized classifiers in detail.

Generation of Keyword-based Category Profiles In hierarchical personalized classification, the keywords of a personalized category must be chosen carefully so that they can discriminate documents relevant to the personalized category from the rest in the same general category. Similar to flat personalized classification, we simulated the user-provided keywords based on Odds Ratio.

Given a personalized category C_l, C_l contains the training documents under the category, and $\overline{C_l}$ consists of training document of its parent general category that are not under the C_l. For instance, the training documents of $\overline{corn}$ are the documents that belong to *grain group* but not *corn*. The Odds Ratio values can be easily computed using Equation 3.

Construction of General Classifiers In our experiments, the general classifiers are also based on SVM. For each general category, we chose all the training documents that belong to its child categories as the positive training documents and all the other training documents from the training document pool to be the negative training documents. Again, the training documents were represented in binary feature vectors when they were used in the construction of the general classifiers. Altogether 3 general classifiers were constructed for the three general categories *crude group*, *grain group* and *veg-oil group*. Each general classifier would determine whether a test document should be classified into the corresponding general category.

Construction of Personalized Classifiers A personalized classifier is built for each personalized category to determine whether a test document should be assigned to it. However, before that could happen, such a test document must be first accepted by the general classifier of the parent general category.

Hence, in the construction of a personalized classifier for HPC, the positive training documents are chosen from the training documents of the parent general category. We use the search engine to rank the training documents of the parent general category. The top-ranked 50 documents are selected as positive training documents and the bottom-ranked 50 documents are selected as negative training documents for the construction of the personalized classifier.

4.5 Performance Measurement

Most of the classification and filtering tasks have been measured using the classical information retrieval notations of *precision* and *recall* [12, 13, 15] denoted by Pr and Re respectively. In TREC-9, two other measures have been used, namely *linear utility measure* and *precision-oriented measure*. These two measures are denoted by $T9U$ and $T9P$ respectively. Let TP be the number of relevant retrieved documents; FP be the number of irrelevant retrieved documents; FN be the number of relevant but not retrieved documents.

$$Pr = \frac{TP}{TP + FP} \tag{5}$$

$$Re = \frac{TP}{TP + FN} \tag{6}$$

$$T9P = \frac{TP}{\max(Target, TP + FP)} \tag{7}$$

$$T9U = \max(2 \times TP - FP, MinU) \tag{8}$$

$T9P$ is the ratio between the relevant retrieved documents over all the retrieved documents. The $Target$ is set to 50 in TREC-9, that is, a target of 50 documents are retrieved over the period of simulation for each category by controlling the classifier (or filter engine) threshold. The actual number of retrieved document may be slightly less or greater than the target. Note that only if the number of retrieved documents is less than the target, Pr will be different from $T9P$. $T9U$ is an utility measure where each relevant retrieved document is given a credit of 2 and each irrelevant retrieved document is given a credit of -1. The lower bound is defined by $MinU$. The $MinU$ is -100 for OHSU topics and -400 for MeSH topics in TREC-9. As for the Reuters collection, we used -100 for $MinU$ in our experiments.

To measure the overall classification performance, the category based average of the measures were computed. For example, given n categories, $C_1, C_2, \cdots, C_n$, $MacPr$ refers to the mean precision over $C_1, C_2, \cdots, C_n$. In this work, only the macro-averages of the performance measures, i.e., $MacPr$, $MacRe$, $MnT9U$ and $MnT9P$, are reported.

5 Experimental Results

5.1 Flat Personalized Classification

For each personalized category, category profiles containing 1 to 5, 10 and 20 keywords were tested. The classification results with the test documents were presented in Table 1.

Table 1. Overall results of FPC

Keywords	$MacPr$	$MacRe$	$MnT9P$	$MnT9U$
1	0.49	0.47	0.49	23
2	0.47	0.43	0.47	20
3	0.45	0.40	0.45	18
4	0.47	0.41	0.47	20
5	0.44	0.40	0.44	17
10	0.45	0.40	0.45	17
20	0.40	0.36	0.40	10

Table 2. Detailed results of FPC (with 4 keywords)

Category	Pr	Re	$T9P$	$T9U$
crude	0.88	0.23	0.88	82
grain	0.82	0.28	0.82	73
veg-oil	0.40	0.54	0.40	10
nat-gas	0.28	0.47	0.28	-8
ship	0.76	0.43	0.76	64
cocoa	0.20	0.56	0.20	-20
corn	0.44	0.39	0.44	16
wheat	0.44	0.31	0.44	16
oilseed	0.32	0.34	0.32	-2
palm-oil	0.12	0.60	0.12	-32

Similar to most of experiments in TREC-9, our experiments were optimized to the $T9P$ measure. That is, $MnT9U$, $MacPr$ and $MacRe$ were computed when 50 was the target number of documents to be retrieved for each category. As the number of target retrieved documents was controlled by a threshold upon outputs of the classifiers for each category, 50 documents can be easily retrieved for each category. This is the reason that our $MacPr$ values were equal to $MnT9P$.

As shown in Table 1, the macro-precision and recall values were within the range 0.36 and 0.49, while $MnT9U$ values were between 10 and 23. These results were comparable to the ones reported for non-adaptive batch filtering in the TREC-9 final report [12] despite different data sets were used. On the whole, the performance of FPC was poor. As SVM had been shown to deliver good classification results [3], we believe that the performance had not been good enough due to the quality of the training documents. As the training documents were selected from a pool of documents using only keywords, it would not be always possible to get exactly 50 correct *positive* training documents from the top 50 documents returned by the search engine, especially for personalized categories that do not have large number of training documents. An example of such personalized categories is *palm-oil* which has only 10 training documents.

Another conclusion that can be drawn from the results is that the number of keywords supplied was not the key factor that affects the performance of flat personalized classification. If very few keywords, for example only one or two, were used, the information carried by the keywords would have been too limited to fully describe the category. Hence, it would be difficult to obtain the correct set of training documents. As the number of keywords increased, more "noise" was added into the category profile. The noise would prevent the correct training documents from being highly ranked by the search engine. From the experimental results, we noticed that about 4 keywords were good enough to describe each personalized category. On the other hand, the 4 keywords must be carefully selected. The detailed results for each personalized category using 4-keyword category profiles are shown in Table 2. It can be noticed that some of the categories received high precision while low recall, for instance, *crude* and *grain*. The reason behind is the number of training documents. It is always easier for the search engine to get 50 high-quality positive training documents for the categories with large number of training documents, e.g., more than 100. If one category has less than 50 positive training documents in the training document pool, even if the search engine ranks all these documents in the top 50, some noise documents will be included as positive training documents for the classifier. In Reuters collection, categories have large number of training documents may have large number of test documents too. Since we only retrieve 50 documents (as our target), a large proportion of test documents for these categories can not be included, and that results the low recall values, e.g., *crude* and *grain*. However, for the categories that have fewer test documents, the recall values could be quite high, e.g., *cocoa* and *palm-oil*.

5.2 Hierarchical Personalized Classification

Similarly, we conducted experiments on hierarchical personalized classification with category profiles containing 1 to 5, 10 and 20 keywords. The results are shown in Table 3.

Table 3. Overall results of HPC

Keywords	$MacPr$	$MacRe$	$MnT9P$	$MnT9U$
1	0.63	0.62	0.63	45
2	0.62	0.60	0.62	42
3	0.61	0.60	0.61	42
4	0.61	0.60	0.61	42
5	0.59	0.57	0.59	39
10	0.57	0.56	0.57	36
20	0.55	0.54	0.55	32

The hierarchical personalized classification method performed much better than flat classification method. The improvements were consistent across all numbers of keywords and performance measures. The reason was that some of the classification efforts had been done by the well-trained general classifiers. For the personalized classifiers, the classification space was limited to the documents accepted by their general classifiers. Consistent with the conclusion given by Dumais and Chen, our experiments confirmed that the performance of hierarchical classification for category tree is better than that for the flat categories [2].

Table 4. Detailed results of HPC (with 4 keywords)

Category	Pr	Re	$T9P$	$T9U$
crude	0.96	0.25	0.96	94
grain	0.94	0.32	0.94	91
veg-oil	0.64	0.86	0.64	46
nat-gas	0.44	0.73	0.44	16
ship	0.98	0.55	0.98	97
cocoa	0.26	0.72	0.26	-11
corn	0.54	0.48	0.54	31
wheat	0.58	0.41	0.58	37
oilseed	0.60	0.64	0.60	40
palm-oil	0.20	1.00	0.20	-20

Once again, the experiment showed that by increasing the number of keywords did not help much in the classification performance. Table 4 presents the results when the number of keywords was 4 for each category profile. Macro sign test (S-test) [16] comparing the detailed results of HPC and FPC showed that in all measures, Pr, Re, $T9P$ and $T9U$, the P-value, $P(Z \geq 10) = 0.000976$. This indicates that the improvement of HPC over FPC was significant.

6 Conclusion

In this paper, we have designed, implemented and evaluated two personalized classification methods known as the flat and hierarchial personalized classification methods. The former allows personalized categories to be defined within a flat category space. The latter requires each personalized category to be defined under some pre-existing general category. Both methods allow users to specify their personalized category profiles by a few keywords.

Our experimental results showed that the hierarchical personalized classification method yielded better performance as the classifiers of the general categories were able to filter away irrelevant documents that could not be effectively recognized by the personalized classifiers built upon only a few user-specified keywords. Nevertheless, compared to the classification systems built upon purely training documents, there are still rooms for performance improvement in our keyword-based personalized classification methods. In particular, our experiments had been conducted on a rather small document collection. A more comprehensive set of experiments could be conducted on a large document collection to examine if the same observations in this paper still hold in a different setting. In the current work, we have also implicitly assumed that the personalized categories are fairly static. In real-life, users may have their interest changed over time. To cope with such evolution and also to improve the accuracy of personalized classification, it is necessary to consider user feedback in the future research.

References

1. T. Ault and Y. Yang. kNN at TREC-9. In *Proc. of the 9th Text REtrieval Conference (TREC-9)*, Gaithersburg, Maryland, 2000.
2. S. T. Dumais and H. Chen. Hierarchical classification of Web content. In *Proc. of the 23rd ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 256–263, Athens, GR, 2000.
3. S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proc. of the 7th Int. Conf. on Information and Knowledge Management*, pages 148–155, 1998.
4. T. Joachims. SVM^{light}, An implementation of Support Vector Machines (SVMs) in C. http://svmlight.joachims.org/.
5. T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proc. of the 10th European Conf. on Machine Learning*, pages 137–142, Chemnitz, DE, 1998.
6. D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *Proc. of the 14th Int. Conf. on Machine Learning*, pages 170–178, Nashville, US, 1997.
7. K.-S. Lee, J.-H. Oh, J. Huang, J.-H. Kim, and K.-S. Choi. TREC-9 experiments at KAIST: QA, CLIR and batch filtering. In *Proc. of the 9th Text REtrieval Conference (TREC-9)*, Gaithersburg, Maryland, 2000.
8. A. K. McCallum. BOW: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/∼mccallum/bow, 1996.
9. D. Mladenic. Feature subset selection in text-learning. In *Proc. of the 10th European Conf. on Machine Learning*, pages 95–100, 1998.
10. D. W. Oard. The state of the art in text filtering. *User Modeling and User-Adapted Interactions: An International Journal*, 7(3):141–178, 1997.
11. M. J. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331, 1997.
12. S. Robertson and D. A. Hull. The TREC-9 filtering track final report. In *Proc. of the 9th Text REtrieval Conference (TREC-9)*, Gaithersburg, Maryland, 2000.
13. F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.
14. TREC. Text REtrieval Conference. http://trec.nist.gov/.
15. Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2):69–90, 1999.
16. Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proc. of the 22nd ACM Int. Conf. on Research and Development in Information Retrieval*, pages 42–49, Berkeley, USA, Aug 1999.
17. Y. Zhang and J. Callan. YFilter at TREC-9. In *Proc. of the 9th Text REtrieval Conference (TREC-9)*, Gaithersburg, Maryland, 2000.