A Dynamic Centroid Text Classification Approach by Learning from Unlabeled Data

Cuicui Jiang, Dingju Zhu andQingshan Jiang

AbstractThe centroid-based classification has proved to be a simple and yet efficient method for text classification. However, the performance of centroid-based classifier depends heavily on the quantity of labeled training set. It is easy and cheap to collect enormous unlabeled data from digital resources, while it is difficult and costly to label these data for training classifiers. To address this problem, we propose a dynamic centroid text classification approach which learns from unlabeled texts to construct dynamic centroids. The main idea of the approach is to take the unlabeled texts with high classifying confidence into consideration to adjust the centroids dynamically. Experiments on two public corpora have indicated the effectiveness of our text classification approach in the case of spare labeled training set.

KeywordsText classification · Dynamic centroid · Confidence · Unlabeled texts

1 Introduction

Text classification is the task of assigning natural language texts to predefined categories. Due to the tremendous growth of digital documents available from the online resources and the ensuing need to organize them, TC has gained much attention in the fields of Information Retrieval and Natural Language Processing.

During the early phase of its development, TC approaches mainly concentrated on manually building expert systems that capable of making TC decisions by means of knowledge engineering (KE) techniques. The drawback of KE approaches is the knowledge acquisition bottleneck well known from the expert systems literature[1]. The subsequent machine learning (ML) approaches for text classification has gained booming development since the early '90th. In ML approaches, a general inductive process automatically builds a classifier for a category C_j by observing the characteristics of a set of labeled texts under C_j or $\widetilde{C_j}$, and from these characteristics, the inductive process gleans the characteristics that a new unseen text should have in order to be classified underC_j. Some of the widely studied and used ML methods include centroid-based method[2], K-Nearest-Neighbor (kNN), Naive Bayes, Support Vector Machine (SVM), Decision Tree and Neural Networks[1,3]. Among various text classification methods, centroid-based classification has proved to be a simple and yet efficient method[2, 4].

Despite the fact that machine learning text classification approach lowers the dependency on expert knowledge in domain and enhances the robustness of classifier, its shortcoming is also obvious: the performance of classifier relies heavily on the availability of labeled training set. That is, it requires a considerable amount of pre-classified texts to learn from in order to train the classifier. However, in most cases, especially those with online resources, it is easy

Cuicui Jiang (✉)
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
e-mail:cc.jiang@siat.ac.cn

Dingju Zhu
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

Qingshan Jiang
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

and cheap to collect enormous unlabeled data from the Internet, digital libraries, and databases, while it is difficult and costly to label these data manually for classifiers training[5]. Learning from labeled and unlabeled data has become one of the research hot-spots in the field of machine learning. Some research have been made on learning from unlabeled data[6,7], and some methods has been applied to text classification, such as transductive learning[8] and co-training[9].

In this paper we propose an improved centroid-based text classification approach named DCTC (Dynamic Centroid Text Classification). The main point of this approach is to take the unlabeled texts with high confidence into consideration to modify the centroid dynamically, so as to reduce the influence of insufficiency of labeled training set.

The reminder of the paper is organized as follows. Section 2 provides an overview of the traditional centroid-based classification method. Section 3 describes the dynamic centroid text classification approach in detail. Section 4 evaluates this new approach based on experimental comparison. Finally, Section 5 summarizes the new approach and the whole paper.

2 The Centroid-based Classification

The centroid-based classification method is extremely simple and easily understandable. Its essential ideais that the more similar a document is to a class, the more likely the document belongs to that class. The similarity between a document and a class is measured by the dot-product of the vectors that represent the document and class. The framework of centroid-based text classification is described below:

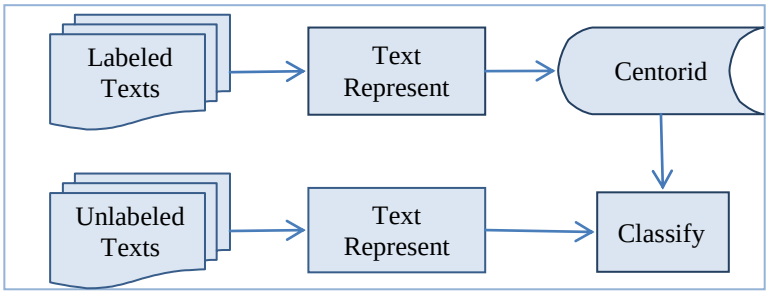

Fig.1 Framework of Centroid-based Text Classification

2.1 Text Representation

In centroid-based classification method, the texts are represented using the vector space model (VSM). In VSM, each text d is regarded as a vector in the term space. That is $\vec{d}=(w_1, w_2, \ldots, w_n)$, where w_k represent the weight of term t_k. The terms can be considered as words, phrases, or chunks of phrases that appearing in corpus. And the term weights can be compute in several ways. The simplest weighting method is the binary weight—1 denoting presence and 0 absence of the term in the text; another method is the Term Frequency (TF) weighting that uses term frequency to weight a term, which is $\mathrm{w}(t_k, d_i) = tf(t_k, d_i)$. The most widely used weighting method is the Term Frequency-Inverse Document Frequency (TF-IDF) method, which defined as:

$$\mathrm{w}(t_k, d_i) = tf(t_k, d_i) \times \log(N/n_k) \quad (1)$$

WhereN denotes the total number of texts and n_k represents the number of texts that contain term t_k. Since $tf(t_k, d_i) \geq 0$, and $\log(N/n_k) \geq 0$, the value of weights are always non-negative. The essential assumption behind TF-IDF function is that the more the times a term appearing in a text, the more significant the term is to symbolizing this text; meanwhile, a term occurring frequently in many texts has limited discrimination power, thus it needs to be

de-emphasized. TF-IDF has been widely studied and applied, and proved to be an effective text representation method.In this paper, we identify terms with words and weight terms using TF-TDF.

2.2 Centorid

Centroid refers to the prototype vector of each category. It comes in three variant forms, the sum centroid, the average centroid and the normalized centroid. For category C_j, the centroids are defined as $\overrightarrow{C_j}(\text{sum}) = \sum_{d \in C_j} \vec{d}, \overrightarrow{C_j}(\text{avg}) = \frac{1}{N_j} \sum_{d \in C_j} \vec{d}$ where N_j denotes the total number of texts in the training set that belong to category C_j, and $\overrightarrow{C_j}(\text{norm}) = \frac{1}{\|\overrightarrow{C_j}\|} \sum_{d \in C_j} \vec{d}$, where $\|\overrightarrow{C_j}\|$ denotes the 2-norm of the centroid vector $\overrightarrow{C_j}$.

No matter which kind of centroids is chosen, the idea behind them is the same—represent a category using documents belong to it. In the case of a relatively small training set, the representativeness of the labeled data to categories is a key issue that needs to be considered.

2.3 Centroid-based Classification

In centroid-based classification, an unlabeled text is classified based on its similarity between each category. For text d_i and category C_j, the similarity is measured with the following formula:

$$\text{Sim}(d_i, C_j) = \frac{\overrightarrow{d_i} \cdot \overrightarrow{C_j}}{\|\overrightarrow{d_i}\| \times \|\overrightarrow{C_j}\|} \quad (2)$$

where $\|\overrightarrow{d_i}\|$ denotes the 2-norm of the text vector $\overrightarrow{d_i}$, and $\|\overrightarrow{C_j}\|$ denotes the 2-norm of centroid vector $\overrightarrow{C_j}$. This similarity measurement is also known as the cosine similarity. Since $\text{sim}(d, C) = \cos(\vec{d}, \vec{C})$, and the elements of the vectors are always non-negative, the value of arbitrary $\text{sim}(d, C)$ is confined to $[0,1]$. The larger the similarity value is, the closer the two vectors in the vector pace are, thus more likely the document d belongs to a certain category C.

Supposing there are m categories in the corpus, and for each category, a centroid is acquired based on the pre-classified texts labeled under it. Then, for an unlabeled text d, the similarities between d and all the centroids are computed. Finally, the text d is assigned to the category which has the maximum similarity to it. That is, the class of text d is determined by:

$$\arg\max\left(\text{Sim}(d, C_j)\right), j = 1,2,\ldots,m. \quad (3)$$

Due to its simplicity and linearity, the centroid-based classifier has been widely used for text classification, web page classification and other classifying applications[10]. Meanwhile, to overcome the inductive bias or model misfit[11] of centroid-based classifier and further improve its performance, many researchers have proposed various improvement strategies. Most of the strategies concentrate on learning from error feedbacks to adjust centroids, such as the batch-update centroid classifier[4] and the iteratively-adjusted centroid classifier[12]. Some of the improvements aim at adjusting centroids by all the documents in training set[11]. And some of the improvements obtained by modifying feature selecting[13] and term weighting[14]methods.

Comparing with the previous well-known modified centroid-based classification methods, our approach aims at an entirely new goal and takes a different learning strategy. That is, reducing the dependency of centroid-based classifier on the scale of training set by learning from unlabeled data.

3 Dynamic Centroid Text Classification

The traditional centroid-based classification is extremely luminous. However, its learning process is supervised and depends on large amount of labeled data to obtain high accuracy. In most text classification tasks, especially those involved with online resources, it is the data labeling but not gathering that really thorny. Meanwhile, unlabeled data carry information of classes[5]. To address this issue and take advantage of unlabeled data, we come up with an improved centroid-based text classification approach DCTC that gradually learns from unlabeled texts and adjusts the centroids dynamically. The framework of the DCTC approach is depicted in the Fig.2:

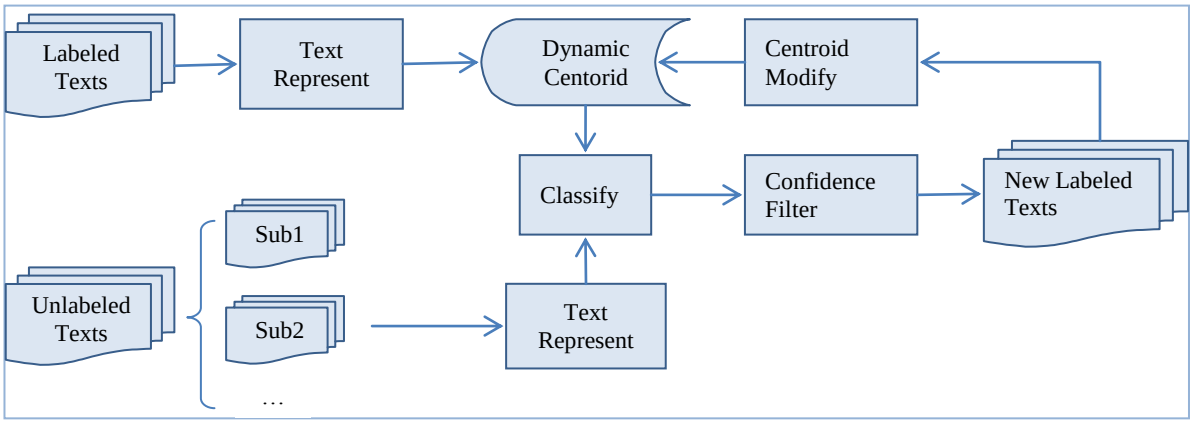

Fig.2 Framework of Dynamic Centroid Text Classification

3.1 Algorithm DCTC

The core process of algorithm DCTC is the updating of centroid with unlabeled texts, and it is detailed below.

First of all, compute the initial centroid of each category using the labeled texts. We used the sum centroid in our experiments.

Secondly, randomly select a number of unlabeled texts and classify them based on current centroids, using the cosine similarity measurement.

Thirdly, for each text classified in the previous step, calculate the Minimum Similarity Interval (MSI) between the text and its assigned category.

Supposing that text d_i is classified to categoryC_j and there total number of categories is M, then the MSI between d_i andC_j is defined as:

$$\mathrm{MSI}(d_i, C_j) = \min\left(\left|\mathrm{Sim}(d_i, C_j) - \mathrm{Sim}(d_i, C_k)\right|\right) \quad (4)$$

where $k = 1,2,\ldots,M$ and $k \neq j$.While $\mathrm{sim}(d_j, C_j)$ indicates the degree that text d_i belongs to categoryC_j , $\mathrm{MSI}(d_i, C_j)$ indicates the discrimination of d_i in differentiating categoryC_j and other categories. Relatively speaking, the lager the $\mathrm{MSI}(d_i, C_j)$ value is, the closer term vector$\overrightarrow{d_i}$ lies to centroid$\overrightarrow{C_j}$ and the farther $\overrightarrow{d_i}$ deviates from other centroids in the vector space.

Considering both similarity and MSI, the classifying confidence, namely, the reliability of the classifying result that d_i belongs to C_j, can be measured with the following formula:

$$\mathrm{Con}(d_i, C_j) = \mathrm{Sim}(d_i, C_j) \times \mathrm{MSI}(d_i, C_j) \quad (5)$$

Finally, use the newly classified texts with high classifying confidence to adjust the previous centroid of each category, respectively. We set up a confidence threshold θ_c as

selection criteria and select out texts with confidence larger than θ_c. Then the modified dynamic centroid DC_j is determined by the following formula:

$$\overrightarrow{DC_j} = \overrightarrow{C_j} + \alpha \times \sum_{d \in C_j \,\&\, Con(d,C_j) > \theta_c} \vec{d} \quad (6)$$

where $\overrightarrow{C_j}$ is the previous centroid of category C_j. And α, which we called learn-rate, is a control parameter that allows setting the relative importance of unlabeled texts in constructing the new centroids. In the case of $\alpha = 0$ or $\theta_c > 1$, the dynamic centroid regresses into the traditional centroid.

In order to take enough quantity of reliable texts into consideration and enhance the representativeness of centroids for the related categories, the centroids are modified via several iterations. To be specific, when the new centroids are computed, the next iteration starts: a number of unlabeled texts are selected and classified based on the new centroids, MSIs and confidences are calculated and the current centroids are modified, and the process iterates. Considering that more training samples do not guarantee more accurate classifying model[4], we set up a maximum number of iterations to avoid the possible over-fitting and poor generalization problem caused by too many supporting data.

The Dynamic Centroid Text Classification algorithm can be summarized as below:

Algorithm: DCTC

Input:

L: Labeled text set
U: Unlabeled text set
SU: Random subset of U with a fixed scale
M: Maximum number of iterations
θ_c: Confidence threshold
α: Learn-rate

Output:

C: the final centroids for each category

Begin

1 Calculate initial centroids C according to L;
2 for $i = 1$ to M do
2.1 for each text d in SU_i do
2.1.1 Classify d using Equation (2) (3);
2.1.2 Calculate $\mathrm{Confidence}(d, C)$ usingEquation (4) (5);
2.1.3 if $\mathrm{Confidence}(d, C) > \theta_c$ then
Add d to New-Labeled-Text-Set NL;
2.2 $\mathrm{AdjustCentroids}(NL, C)$ usingEquation (6);
2.3 Empty NL;
3 Output C;

End

When the iterations terminate, we select the centroids with best performance during the iterations as the final centroids, and then apply the final centroids in classifying the remaining texts.

3.2Complexity Analyses

One of the significant advantages of traditional centroid-based classification is linear computing complexity. Compared with the original algorithm, in the training phase, algorithm

DCTC has two additional procedures: confidence calculating and centroids adjusting. As to confidence calculating, the computing takes two pass through the training set on the foundation of calculated similarities. For centroids adjusting, in the worst case that all the classified texts are utilized in adjusting centroids, the computing complexity is $O(tn)$, where t is the dimension of term vector and n is the total number of the training set. And the following classifying phase is the same with traditional centroid-based classifying. Accordingly, the algorithm complexity of DCTC is still linear on the number of the texts and the feature vector dimension.

4 Experimental Evaluation

4.1 Datasets

In the experiments, we used two Chinese texts corpora: Sogou text classification corpus (SgTCC) [15] and Fudan text classification corpus (FdTCC)[16].

SgTCC: Sogou text classification corpus is an open Chinese text dataset supported by the R&D Center of SOHU, for the purpose of offering a standard test platform for Chinese text classification. It includes 9 categories and each category contains about 1990 texts, thus it is a balanced dataset. To reduce the computing scale, meanwhile guarantee the analyzability of the data, we select 5 categories and use all the texts under these categories as our experimental data.

FdTCC: Fudan text classification corpus is a Chinese text classification dataset collected and organized by the Natural Language Processing Team of Computer Information and Technology Department of Fudan University. The training corpus of FdTCC includes 20 categories and it is an unbalanced dataset. For the purpose of reducing computing scale, we select 5 categories and texts under these categories as our experimental data.

The detail information of datasets used in our experiments is displayed in Table1.

Table 1 Detail information of datasets

Dataset	Category	Number of texts	Number in total
SgTCC	Economy	1992	9947
	Health	1990	
	Military	1990	
	Sports	1985	
	Travel	1990	
FdTCC	Economy	1600	3343
	Medical	51	
	Military	74	
	Politics	1024	
	Sports	494	

4.2 Performance Metrics

The performance metrics using in our experiments are recall, precision and F1-measure, which are defined as:

$$\text{Recall} = \frac{\text{number of correct positive predictions}}{\text{number of positive examples}}$$

$$\text{Precision} = \frac{\text{number of correct positive predictions}}{\text{number of positive predictions}}$$

$$\mathrm{F1} = \frac{2 \times \mathrm{Recall} \times \mathrm{Precision}}{\mathrm{Recall} + \mathrm{Precision}}$$

The above metrics are for each category separately and have a local significance for performance evaluating. Since we are dealing with multiple categories, we use the averaging measure of these metrics, namely, macro-average and micro-average as the final performance evaluation criteria, which are defined as:

$$\mathrm{Macro\_F1} = \frac{2 \times \mathrm{avg(Recall)} \times \mathrm{avg(Precision)}}{\mathrm{avg(Recall)} + \mathrm{avg(Precision)}}$$

$$\mathrm{Micro\_F1} = \frac{\text{total number of correct predictions}}{\text{total number of examples}}$$

The macro-average weights equally all the classes, regardless of how many documents belong to it. The micro-average weights equally all the documents, thus favoring the performance on common classes.

4.3 Experimental Design

To evaluate the performance of the DCTC algorithm, we design two series of experiments, one for parameters optimization of DCTC and another for performance comparison with the centroid-based method and k-NN method.

The vital control parameters in DCTC algorithm includes the confidence threshold θ_c, and the learn rate α. For the purpose of parameter optimization, we fix one of the parameters and varied the rest one, and then select out the values which produce the best performance. To make the selection more reliable, we conduct two groups of trials; each group includes several experiments that based on a certain amount of labeled texts.

We then validate the performance of the DCTC by comparison with another two classical text classification methods on two datasets, respectively. The comparative experiments are conducted based on the same labeled and unlabeled data.

4.4 Experimental Results and Analysis

1. Parameters Optimization

The confidence thresholdθ_c is set to select out the classified texts with high reliability. To consider it intuitively, neither too small nor too large value of θ_c is proper: too small values result in over amount of less reliable supporting set, while too large values lead to insufficient supporting set. Whenθ_c exceeds 1, there will be no texts selected out for centroid adjusting and the dynamic centroid regresses into the traditional centroid.

Fig.3 illustrates the classifying performance under different values ofθ_c, assessed by MacroF1. The learn-rate is set to 0.4 and the maximum number of iterations is set to 6 in all the experiments displayed in Fig. 3. The scale of labeled texts is 30 on SgTCC and 25 on FdTCC.

From Fig. 3, we can see that the trend of influence that θ_c has on DCTC performance is consisted with what we intuitively speculated on both datasets. When the confidence threshold is smaller than 0.35, the performance is relatively poor and increases rapidly with the increasing of θ_c in a certain range. When the value of θ_c exceeds a certain range, the performance slightly descends and then keeps flat. The optimum values occur around 0.4 on SgTCC and around 0.5 on FdTCC.

In order to optimize the value of learn-rateα, we setθ_c to a fixed value 0.4 on dataset SgTCC and another fixed value 0.5 on FdTCC, which generate the best result in previous experiments, and vary the value of α. Other parameters are set as the same values as in the previous experiments. The experimental results are displayed in Fig. 4.

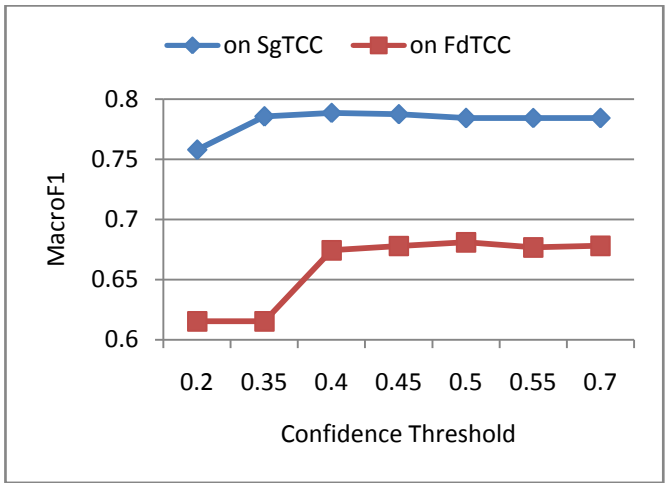

Fig.3 Performance curves of DCTC vs. confidence threshold on two datasets

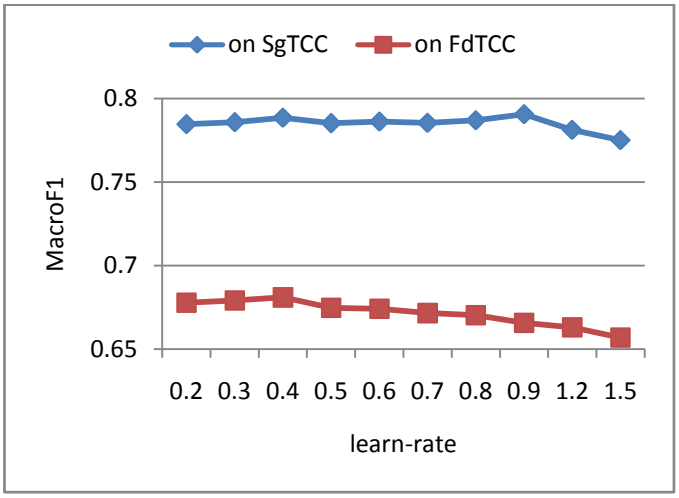

Fig.4 Performance curves of DCTC vs. learn-rate on two datasets

From Fig. 4, we can see that the optimum values of learn-rate occur around 0.9 on SgTCC and around 0.4 on FdTCC. Although the points of the best performance on two datasets are different, the trends of the curves are similar. When the learn-rate is smaller than 0.4, the performance slowly increases with the increasing of learn-rate. When the learn-rate exceeds a certain value (0.9 on SgTCC, and 0.4 on FdTCC), the performances apparently decrease on both datasets. The ideal value for learn-rate is between 0.4 and 0.9.

We can conclude from both Fig. 3 and Fig. 4 that though the optimum values of parameters for different datasets vary, the trends of the parameters influences on performance are consistent on both datasets.

2. Performance Comparison

After parameters optimization, we made a performance comparison of DCTC with the centroid-based method and the kNN method on two datasets. For DCTC, confidence threshold and

learn-rate are set to 0.4 and 0.9 on SgTCC, 0.5 and 0.4 on FdTCC. As to kNN, k is set to 7 on SgTCC and 5 on FdTCC. The experiments results are displayed in Table 2.

According to Table 2, the performance of DCTC steadily outperforms that of Centroid-base method and kNN method on dataset SgTCC. On its best case, the MacroF1 of DCTC is larger than that of centroid-based method by 1.46%, and the MicroF1 of DCTC is larger than that of centroid-based method by 1.29% when the scale of labeled texts is 30. With the increasing of the number of labeled texts, the performance advantage of DCTC remains, but the performance gap reduces. The performance of DCTC just slightly outperforms that of Centroid method on dataset FdTCC. That is to say, the advantage of DCTC on FdTCC is not as apparent as it is on SgTCC. Besides, comparing MacroF1 with MicroF1, we can see that the gap between them on SgTCC is much smaller than that on FdTCC, which indicates that all the three methods are sensitive to the imbalance of dataset. Since SgTCC is a balanced corpus and FdTCC is an unbalanced corpus, we can infer that DCTC is fitter for balanced dataset.

Table 2 Performance comparison among DCTC, Centroid and kNN

Dataset	MacroF1			MircroF1			Labeled Texts
	DCTC	Centroid	kNN	DCTC	Centroid	kNN	
SgTCC	0.7907	0.7761	0. 4915	0.7846	0.7717	0.4917	30
SgTCC	0.8138	0.8092	0. 6548	0.8098	0.8050	0.6473	50
FdTCC	0.6649	0.6629	0.3287	0.8011	0.8005	0.4200	15
FdTCC	0.6810	0.6792	0.4433	0.8147	0.8140	0.6115	25

It is worth mention that when the scale of labeled texts is relatively very small (say, 25 labeled texts out of 9947 texts), the DCTC approach can still reach a satisfying performance. This shows that our approach works well on spare training set corpora by learning from unlabeled data.

5 Conclusions

In this paper, we propose an improved centroid text classification approach which can learn from unlabeled texts to construct a dynamic centroid classifier. The main idea of this approach is to take the unlabeled texts with high confidence into consideration to adjust the centroids gradually. We conducted a series of experiments for parameters optimization and performance evaluation. The results indicate that our method outperforms traditional centroid method and kNN method on two public corpora. And it can reach a desirable performance by learning from unlabeled data.

We also notice that the advantage of our approach on unbalanced dataset is not as apparent as it on balanced dataset. This shows us one of the directions for future work—research on the suitability of DCTC for unlabeled dataset. Meanwhile, there are other ways to further improve the performance of the new approach, such as a more reasonable iteration terminating criteria, but not a fix number of iterations, or a more suitable centroid adjusting method. And the new approach should be verified on more datasets in the future work.

AcknowledgementsThis work was supported by the Shenzhen Internet Industry Development Fund under grant No.HLE201104220082A, and the National Natural Science Foundation of China under Grants No.61175123.

References

1. F. Sebastiani.(2002). Machine Learning in Automated Text Categorization,.

2. E. Han, G. Karypis. (2000). Centroid-Based Document Classification Algorithms: Analysis and Experimental Results, in: PKDD, pp. 424-432.
3. X. Guixian. (2010). The Research on Technology of Text Classification, Beijing.
4. T. Songbo. (2008). An improved centroid Classifier for text categorization.
5. K. Nigam. (1998). A. McCallum, S. Thrun, T. Mitchell, Learning to Classify Text from Labeled and Unlabeled Documents.
6. T. Zhang, F. Oles. (2000). The value of unlabeled data for classification problems, in: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 1191-1198.
7. R. Raina, A. Battle, H. Lee, B. Packer, A.Y. Ng. (2007). Self-taught learning: transfer learning from unlabeled data, in: Proceedings of the 24th International Conference on Machine Learning, pp. 759-766.
8. G. Ifrim, G. Weikum. (2006). Transductive Learning for Text Classification Using Explicit Knowledge Models, in: PKDD, pp. 223-234.
9. A. Blum, T. Mitchell. (1998). Combining labeled and unlabeled data with co-training, in: Conference on Computational Learning Theory, pp. 92-100.
10. C. Jebari. (2012). MLICC:A Multi-Label and Incremental Centroid-Based Classification of Web Pages by Genre, in: NLDB, pp. 183-190.
11. C. Shen, B. Wu. (2012). A New Algorithm Based on Centroid for Text Categorization, in: International Conference on Fuzzy Systems and Knowledge Discovery, pp. 1265-1269.
12. W. Deqing, Z. Hui. (2013).Support-vector-based iteratively adjusted centroid classifier for text categorization.
13. X. Hua. (2010).Research of Text Categorization Based on Feature Selection and Centorid Construction, in, Dalian University of Technology.
14. T. T, Nguyen, K. Chang, S.C. Hui. (2013). Supervised term weighting centroid-based classifiers for text categorization.
15. Sogou Text Classification Corpus.http://www.sogou.com/labs/dl/c.html.Accessed 15 Aug. 2013.
16. Fudan Text Classification Corpus.http://www.datatang.com/data/44139. Accessed 15 Aug. 2013.