UDC 004.93

Vestnik SibGAU
Vol. 16, No. 1, P. 119–123

FEATURES SELECTION FOR TEXT CLASSIFICATION BASED ON CONSTRAINTS FOR TERM WEIGHTS

R. B. Sergienko*, M. Shan Ur Rehman, A. E. Khan, T. O. Gasanova, W. Minker

Ulm University
43, Albert-Einstein-Allee, Ulm, 89081, Germany
*E-mail: roman.sergienko@uni-ulm.de

Text classification is an important data analysis problem which can be applied in different domains including airspace industry. In this paper different text classification problems such as opinion mining and topic categorization are considered. Different text preprocessing techniques (TF-IDF, ConfWeight, and the Novel TW) and machine learning algorithms for classification (Bayes classifier, k-NN, SVM, and artificial neural network) are applied. The main goal of the presented investigations is to decrease text classification problem dimensionality by using features selection based on constraints for term weights. Such features selection provides significant reduction of dimensionality and less computational time for calculations. Besides, the use of constraints for term weights could increase classification effectiveness. We have observed such increase for three out of five problems. In the remaining two problems, no significant change and a decrease of classification effectiveness was observed.

Keywords: topic categorization, text classification, opinion mining, features selection, term weighting, constraint.

Вестник СибГАУ
Т. 16, № 1. С. 119–123

ОТБОР ПРИЗНАКОВ ДЛЯ КЛАССИФИКАЦИИ ТЕКСТОВ НА ОСНОВЕ ОГРАНИЧЕНИЙ ДЛЯ ВЕСОВ ТЕРМОВ

Р. Б. Сергиенко*, М. Шан Ур Реман, А. Э. Хан, Т. О. Гасанова, В. Минкер

Ульмский университет
Германия, 89081, г. Ульм, Аллея Альберта Эйнштейна, 43
*E-mail: roman.sergienko@uni-ulm.de

Классификация текста – актуальная задача анализа данных, которая может найти применение в различных областях, включая аэрокосмическую индустрию. Рассматриваются различные задачи классификации текста, такие как извлечение мнения и категоризация темы. Применяются различные походы предобработки текстовой информации (TF-IDF, ConfWeight, Novel TW) и различные алгоритмы машинного обучения для классификации (классифкатор Байеса, метод ближайших соседей, метод опорных векторов, искусственные нейронные сети). Главная задача представленных в статье исследований – уменьшение размерности задачи классификации текста за счёт отбора признаков на основе ограничений для весов термов. Такое снижение размерности обеспечивает значимое снижение размерности и сокращает время для вычислений. Кроме того, использование ограничений на веса термов может повысить точность классификации на некоторых задачах. Такое увеличение наблюдалось на трёх задачах из пяти, на одной задаче не наблюдалось значимых изменений и ещё на одной зафиксировано незначительное снижение точности классификации.

Ключевые слова: категоризация темы, классификация текста, извлечение мнения, отбор признаков, взвешивание термов, ограничение.

1. Introduction

Nowadays, Internet and social media generate a huge amount of textual information. It is increasingly important to develop methods of text processing such as text classification. Text classification is very important for such problems as automatic opinion mining (sentiment analysis) and topic categorization of different articles from newspapers and Internet.

Text classification can be considered to be a part of natural language understanding, where there is a set of predefined categories and the task is to automatically assign new documents to one of these categories. The method of text preprocessing and text representation influences the results that are obtained even with the same classification algorithms.

The most popular model for text classification is vector space model. In this case text categorization may be considered as a machine learning problem. Complexity of text categorization with vector space model is compounded by the need to extract the numerical data from text information before applying machine learning methods. Therefore text categorization consists of two parts: text

preprocessing and classification using obtained numerical data.

All text preprocessing methods are based on the idea that the category of the document depends on the words or phrases from this document. The simplest approach is to take each word of the document as a binary coordinate and the dimension of the feature space will be the number of words in our dictionary ("bag of the words" method [1]).

There exist more advanced approaches for text preprocessing to overcome this problem such as TF-IDF [2] and Confident Weights (ConfWeight) methods [3]. The task of these text preprocessing methods is to assign weight to each word in a document ranging between [0, 1] which shows the importance of contribution of a word in a document. A novel term weighting method [4] is also considered, which has some similarities with the ConfWeight method, but has improved computational efficiency. In [4] the novel term weighting method was applied for natural language call routing and in [5] it was applied for opinion mining and topic categorization problems.

For large databases it is important to reduce dimensionality of the problem for effective application of machine learning algorithms. The main purpose of dimensionality reduction is to decrease the processing time for a particular machine learning algorithm while still producing the same or acceptable classification results. There are two different ways to reduce dimensionality: features extraction and features selection. In the first approach a small number new features is generated from previous ones. In the second approach useless and non-informative features are removed. In [6] the novel features extraction method for classification was proposed. The method uses terms clustering and optimization of cluster weights with cooperative coevolutionary algorithm [7]. In this paper we propose dimensionality reduction method for text classification based on features selection.

Term weighting techniques provide a natural method for features selection based on constraints for term weights. In text preprocessing, it can be noted that some words in documents are actually meaningless and they do not effectively contribute to the end result. In principle these words are not useful for text classification, moreover due to these words we end up with a bigger feature space. Therefore, the main goal of the investigations presented in the paper is to decrease text classification problem dimensionality with features selection based on constraints for term weights. Also a comparison between classification effectiveness without constraints and with difference values of the constraints for different text classification problems is presented.

In this paper we have used *k*-nearest neighbors algorithm, Bayes Classifier, support vector machine (SVM), and Neural Network as classification methods. *RapidMiner* has been used as implementation software [8].

For the application of algorithms and comparison of the results we have used the DEFT ("Défi Fouille de Texte") Evaluation Package 2008 [9] which has been provided by ELRA and publically available corpora from DEFT'07 [10]. Some results of text classification on the databases are available in the papers [11–14].

This paper is organized as follows: Section 2 describes the corpora. Section 3 explains different term weighting methods. In Section 4 we discuss our experimental results. Finally, we draw some conclusions in Section 5.

2. Corpora Description

The focus of DEFT 2007 campaign is the sentiment analysis, also called opinion mining. We have used 3 publically available corpora: reviews on books and movies (*Books*), reviews on video games (*Games*) and political debates about energy project (*Debates*). The topic of DEFT 2008 edition is related to the text classification by categories and genres. The data consists of two corpora (T1 and T2) containing articles of two genres: articles extracted from French daily newspaper Le Monde and encyclopedic articles from Wikipedia in French language. This paper reports on the results obtained using both tasks of the campaign and focuses on detecting the category.

All databases are divided into a training (60 % of the whole number of articles) and a test set (40 %). To apply our algorithms we extracted all words which appear in the training set regardless of the letter case and we also excluded dots, commas and other punctual signs. We have not used any additional filtering as excluding the stop or ignore words.

Table 1

Corpora description (DEFT'07)

Corpus	Size	Classes
Books	Train size = 2074 Test size = 1386 Vocabulary = 52507	0: negative, 1: neutral, 2: positive
Games	Train size = 2537 Test size = 1694 Vocabulary = 63144	0: negative, 1: neutral, 2: positive
Debates	Train size = 17299 Test size = 11533 Vocabulary = 59615	0: against, 1: for

Table 2

Corpora description (DEFT'08)

Corpus	Size	Classes
T1	Train size = 15223 Test size = 10596 Vocabulary = 202979	0: Sport, 1: Economy, 2: Art, 3: Television
T2	Train size = 23550 Test size = 15693 Vocabulary = 262400	0: France, 1: International, 2: Literature, 3: Science, 4: Society

3. Text Preprocessing Methods

TF-IDF. TF-IDF [2] is a well-known unsupervised approach for term weighting based on multiplication of term frequency tf_{ij} (ratio between the number of times the i^{th} word occurs in the j^{th} document and the document size) and inverse document frequency idf_i:

$$tf_{ij} = \frac{t_{ij}}{T_j}, \quad (1)$$

where t_{ij} is the number of times the i^{th} word occurs in the j^{th} document; T_j is the document size (number of the words in the document).

There are different ways to calculate the weight of each word. In this paper we run classification algorithms with the following variants:

1) TF-IDF 1

$$idf_i = \log \frac{|D|}{n_i}, \quad (2)$$

where $|D|$ is the number of document in the training set; n_i is the number of documents that have the i^{th} word;

2) TF-IDF 2.

The formula is given by equation (2) except n_i is calculated as the number of times i^{th} word appears in all documents from the training set.

Confidence Weights (ConfWeight). Maximum Strength (Maxstr) is an alternative method to find the word weights. This approach has been proposed by Soucy and Mineau [3]. It implicitly does feature selection since all frequent words have zero weights. The main idea of the method is that the feature *f* has a non-zero weight in class *c* only if the *f* frequency in documents of the *c* class is greater than the *f* frequency in all other classes. The ConfWeight method uses Maxstr as an analog of IDF:

$$ConfWeight_{ij} = \log(tf_{ij} + 1) \cdot Maxstr(i) . \quad (3)$$

Numerical experiments [3] have shown that the ConfWeight method could be more effective than TF-IDF with SVM and *k*-NN as classification methods. The main drawback of the ConfWeight method is computational complexity. This method is more computationally demanding than TF-IDF method because the ConfWeight method requires time-consuming statistical calculations such as Student distribution calculation and confidence interval definition for each word.

Novel Term Weighting (TW). The main idea of the method [4] is similar to ConfWeight but it is not so time-consuming. The idea is that every word that appears in the article has to contribute some value to the certain class and the class with the biggest value we define as a winner for this article.

For each term we assign a real number term relevance that depends on the frequency in utterances. Term weight is calculated using a modified formula of fuzzy rules relevance estimation for fuzzy classifiers [15]. Membership function has been replaced by word frequency in the current class. The details of the procedure are the following:

Let L be the number of classes; n_i is the number of articles which belong to the i^{th} class; N_{ij} is the number of the j^{th} word occurrence in all articles from the i^{th} class; $T_{ij} = N_{ij} / n_i$ is the relative frequency of the j^{th} word occurrence in the i^{th} class.

$R_j = \max_i T_{ij}$, $S_j = \arg(\max_i T_{ij})$ is the number of class which we assign to the j^{th} word.

The term relevance, C_j, is given by

$$C_j = \frac{1}{\sum_{i=1}^{L} T_{ji}} \left(R_j - \frac{1}{L-1} \sum_{\substack{i=1 \\ i \neq S_j}}^{L} T_{ij} \right). \quad (4)$$

C_j is higher if the word occurs more often in one class than if it appears in many classes. We use novel TW as an analog of IDF for text preprocessing.

The learning phase consists of counting the C values for each term; it means that this algorithm uses the statistical information obtained from the training set.

4. Experimental Results

We have considered 4 different text preprocessing methods (2 modifications of TF-IDF, ConfWeight, and the novel TW method) and compared them using different classification algorithms. The methods have been implemented using *RapidMiner* [8]. The classification methods are:

– *k*-nearest neighbors algorithm with distance weighting (we have varied *k* from 1 to 15) (*k*-NN);

– kernel Bayes classifier with Laplace correction (Bayes);

– artificial neural network with error back propagation (standard setting in *RapidMiner*) (ANN);

– linear support vector machine (standard setting in *RapidMiner*) (SVM).

There is no predefined method to search or investigate the constraint values as it heavily depends on the document under test. Some constraint values might work for a specific document with a particular term weighting technique but it might not be effective for another text preprocessing technique for the same document, as a result the task of searching an appropriate constraint value is highly experimental.

The judgment of a constraint value depends on final result (macro *F*-measure as classification effectiveness criterion) it produces. The approach to investigate these values is to generate as many result files as possible with randomly but sensible chosen values and then comparing results with the results of without constraint.

Keeping this approach in mind, we have generated *F*-measure result with 10 different constraint values for a specific document with a particular text preprocessing technique. The values were chosen randomly but at the same time if a particular value didn't produce acceptable result another value was put into test. These values are varied from 0,01 to 0,35.

The train and test files for different preprocessing techniques were generated via *Microsoft Visual Studio C++ 2010* and then merged into one train file containing 60 % of train and 40 % of test set which is further used by *RapidMiner* to generate precision and recall result files. Since the size of the Corpus is large the processing or compilation of train files were performed using the computational power of *University of Ulm* Cluster Computers. In the final step precision and recall results are used by *F*-measure project in *Microsoft Visual Studio C++ 2010* to produce *F*-measure score. *F*-measure Score is calculated as it gives another perspective to see our results which in most of the cases overshadow the accuracy result generated by *RapidMiner*.

The numerical results with values of *F*-measure are presented in tab. 3–7 for each text classification problem. The best results for different values *k* of *k*-NN method and for different constraint values are shown. Constraint values are in brackets. Tab. 8 shows the overall comparison of maximum *F*-measure results with the best techniques, which gives a clear picture whether appropriate results are obtained if constraints are applied or not.

Table 3

The numerical results for *Books*

Term weighting	Without constraint				With constraint			
	Bayes	*k*-NN	SVM	ANN	Bayes	*k*-NN	SVM	ANN
TFIDF 1	0,495	0,517 *k* = 3	0,499	0,505	0,516 (0,04)	0,504 *k* = 15 (0,16)	0,527 (0,04)	0,499 (0,04)
TFIDF 2	0,506	0,516 *k* = 1	0,511	0,505	0,508 (0,08)	0,504 *k* = 15 (0,25)	0,525 (0,17)	0,509 (0,06)
ConfWeight	0,238	0,559 *k* = 15	0,238	0,570	0,238 (0,03)	0,544 *k* = 12 (0,05)	0,238 (0,03)	0,546 (0,03)
Novel TW	0,437	0,488 *k* = 14	0,516	0,493	0,238 (0,03)	0,493 *k* = 15 (0,08)	0,490 (0,03)	0,485 (0,03)

Table 4

The numerical results for *Games*

Term weighting	Without constraint				With constraint			
	Bayes	*k*-NN	SVM	ANN	Bayes	*k*-NN	SVM	ANN
TFIDF 1	0,652	0,672 *k* = 3	0,665	0,677	0,681 (0,13)	0,693 *k* = 4 (0,01)	0,669 (0,25)	0,687 (0,20)
TFIDF 2	0,651	0,671 *k* = 7	0,661	0,664	0,684 (0,19)	0,696 *k* = 5 (0,16)	0,677 (0,08)	0,679 (0,04)
ConfWeight	0,210	0,720 *k* = 15	0,210	0,717	0,210 (0,03)	0,731 *k* = 14 (0,01)	0,210 (0,03)	0,731 (0,01)
Novel TW	0,675	0,699 *k* = 13	0,675	0,691	0,210 (0,03)	0,695 *k* = 11 (0,03)	0,684 (0,01)	0,675 (0,01)

Table 5

The numerical results for *Debates*

Term weighting	Without constraint				With constraint			
	Bayes	*k*-NN	SVM	ANN	Bayes	*k*-NN	SVM	ANN
TFIDF 1	0,637	0,637 *k* = 15	0,642	0,638	0,673 (0,01)	0,675 *k* = 15 (0,13)	0,678 (0,20)	0,680 (0,05)
TFIDF 2	0,639	0,634 *k* = 15	0,640	0,632	0,670 (0,28)	0,671 *k* = 11 (0,03)	0,676 (0,10)	0,678 (0,15)
ConfWeight	0,363	0,695 *k* = 15	0,634	0,705	0,363 (0,01)	0,699 *k* = 13 (0,01)	0,637 (0,01)	0,710 (0,01)
Novel TW	0,616	0,695 *k* = 15	0,702	0,697	0,363 (0,02)	0,694 *k* = 15 (0,02)	0,699 (0,10)	0,698 (0,10)

Table 6

The numerical results for *T1*

Term weighting	Without constraint				With constraint			
	Bayes	*k*-NN	SVM	ANN	Bayes	*k*-NN	SVM	ANN
TFIDF 1	0,591	0,816 *k* = 15	0,804	0,830	0,803 (0,01)	0,811 *k* = 11 (0,35)	0,810 (0,18)	0,818 (0,30)
TFIDF 2	0,690	0,808 *k* = 15	0,812	0,808	0,807 (0,35)	0,810 *k* = 15 (0,20)	0,810 (0,30)	0,817 (0,30)
ConfWeight	0,837	0,855 *k* = 14	0,848	0,853	0,529 (0,01)	0,850 *k* = 13 (0,09)	0,835 (0,05)	0,857 (0,01)
Novel TW	0,794	0,837 *k* = 13	0,834	0,854	0,753 (0,05)	0,829 *k* = 12 (0,20)	0,838 (0,25)	0,846 (0,02)

Table 7

The numerical results for *T2*

Term weighting	Without constraint				With constraint			
	Bayes	*k*-NN	SVM	ANN	Bayes	*k*-NN	SVM	ANN
TFIDF 1	0,844	0,846 *k* = 15	0,846	0,847	0,844 (0,05)	0,846 *k* = 13 (0,17)	0,846 (0,35)	0,847 (0,17)
TFIDF 2	0,842	0,847 *k* = 14	0,846	0,847	0,844 (0,26)	0,846 *k* = 15 (0,20)	0,846 (0,23)	0,847 (0,26)
ConfWeight	0,500	0,825 *k* = 15	0,824	0,829	0,498 (0,01)	0,824 *k* = 14 (0,05)	0,824 (0,01)	0,831 (0,01)
Novel TW	0,777	0,862 *k* = 12	0,859	0,847	0,781 (0,05)	0,862 *k* = 14 (0,05)	0,860 (0,10)	0,862 (0,10)

Table 8

The overall comparison

Problem	Without constraint			With constraint			
	F-measure	Term weighting	Classification algorithm	*F*-measure	Term weighting	Classification algorithm	Constraint
Books	**0,570**	ConfWeight	ANN	0,546	ConfWeight	ANN	0,03
Games	0,720	ConfWeight	*k*-NN (*k* = 15)	**0,731**	ConfWeight	ANN	0,01
Debates	0,705	ConfWeight	ANN	**0,710**	ConfWeight	ANN	0,01
T1	0,855	ConfWeight	*k*-NN (*k* = 14)	**0,857**	ConfWeight	ANN	0,01
T2	0,862	Novel TW	*k*-NN (*k* = 12)	0,862	Novel TW	ANN	0,10

5. Conclusions

Features selection for text classification based on constraints for term weights was investigated with different term weighting method (TF-IDF, Confident Weights, and the Novel TW), different classification algorithms (Bayes classifier, *k*-NN, SVM, and artificial neural network) for different text classification problems (opinion mining, topic categorization). Such features selection provides significant reduction of dimensionality and less computational time for calculations. Besides, the use of constraints for term weights could increase classification effectiveness. We have observed such increase for three out of five problems. In the remaining two problems, no significant change and a decrease of classification effectiveness was observed.

References

1. Joachims T. Learning to classify text using support vector machines: Methods, theory and algorithms. Kluwer Academic Publishers, 2002, p. 205.

2. Salton G. and Buckley C. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*. 1988, p. 513–523.

3. Soucy P., Mineau G. W. Beyond TFIDF Weighting for Text Categorization in the Vector Space Model. *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*. 2005, p. 1130–1135.

4. T. Gasanova R. Sergienko W. Minker E. Semenkin, Zhukov E. A Semi-supervised Approach for Natural Language Call Routing. *Proceedings of the SIGDIAL 2013 Conference*, August 2013, p. 344–348.

5. Gasanova T., Sergienko R., Akhmedova S., Semenkin E., Minker W. Opinion Mining and Topic Categorization with Novel Term Weighting. *ACL 2014*. 2014, p. 84.

6. Gasanova T., Sergienko R., Semenkin E., Minker W. Dimension Reduction with Coevolutionary Genetic Algorithm for Text Classification. *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Vienna University of Technology, Austria, September 2014, vol. 1, p. 215–222.

7. Potter M. A., De Jong K. A. Cooperative coevolution: an architecture for evolving coadapted subcomponents. Trans. Evolutionary Computation, 8, Jan. 2000, p. 1–29.

8. Shafait F., Reif M., Kofler C., and Breuel T. M. Pattern Recognition Engineering. *RapidMiner Community Meeting and Conference*, 2010, p. 9.

9. DEFT (DÉfi Fouille de Textes). Available at: http://deft.limsi.fr/.

10. European Language Recourses Association. DEFT'08 Evaluation Package. Available at: http://catalog.elra.info/product_info.php?cPath=42_43&products_id=1165.

11. Bechet F., Beze M. E., Torres-Moreno J.-M. Proceedings of the 4th DEFT Workshop (Avignon, France, June 8–13, 2008). DEFT '08. TALN, Avignon, France, 2008, p. 27–36.

12. Charnois T., Doucet A., Mathet Y., Rioult F. Proceedings of the 4th DEFT Workshop (Avignon, France, June 8–13, 2008). DEFT '08. TALN, Avignon, France, 2008, p. 37–46.

13. Charton E., Camelin N., Acuna-Agost R., Gotab P., Lavalley R., Kessler R., Fernandez S. Proceedings of the 4th DEFT Workshop (Avignon, France, June 8–13, 2008). DEFT '08. TALN, Avignon, France, 2008, p. 47–56.

14. Cleuziou G., Poudat C. Proceedings of the 4th DEFT Workshop (Avignon, France, June 8–13, 2008). DEFT '08. TALN, Avignon, France, 2008, p. 57–64.

15. Ishibuchi H., Nakashima T., Murata T. Trans. on Systems, Man, and Cybernetics, 1999, vol. 29, p. 601–618.