Opinion Mining and Topic Categorization with Novel Term Weighting

Tatiana Gasanova
Institute of Communications Engineering, Ulm University, Germany
tatiana.gasanova@uni-ulm.de

Roman Sergienko
Institute of Communications Engineering, Ulm University, Germany
roman.sergienko@uni-ulm.de

Shakhnaz Akhmedova
Institute of Computer Science and Telecommunications, Siberian State Aerospace University, Russia
shahnaz@inbox.ru

Eugene Semenkin
Institute of Computer Science and Telecommunications, Siberian State Aerospace University, Russia
eugenesemenkin@yandex.com

Wolfgang Minker
Institute of Communications Engineering, Ulm University, Germany
wolfgang.minker@uni-ulm.de

Abstract

In this paper we investigate the efficiency of the novel term weighting algorithm for opinion mining and topic categorization of articles from newspapers and Internet. We compare the novel term weighting technique with existing approaches such as TF-IDF and ConfWeight. The performance on the data from the text-mining campaigns DEFT'07 and DEFT'08 shows that the proposed method can compete with existing information retrieval models in classification quality and that it is computationally faster. The proposed text preprocessing method can be applied in large-scale information retrieval and data mining problems and it can be easily transported to different domains and different languages since it does not require any domain-related or linguistic information.

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 opining 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.

There exist more advanced approaches for text preprocessing to overcome this problem such as TF-IDF (Salton and Buckley, 1988) and ConfWeight methods (Soucy and Mineau, 2005). A novel term weighting method (Gasanova et al., 2013) is also considered, which has

Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 84–89,
Baltimore, Maryland, USA. June 27, 2014.

some similarities with the ConfWeight method, but has improved computational efficiency. It is important to notice that we use no morphological or stop-word filtering before text preprocessing. It means that the text preprocessing can be performed without expert or linguistic knowledge and that the text preprocessing is language-independent.

In this paper we have used *k*-nearest neighbors algorithm, Bayes Classifier, support vector machine (SVM) generated and optimized with COBRA (Co-Operation of Biology Related Algorithms) which has been proposed by Akhmedova and Semenkin (2013), Rocchio Classifier or Nearest Centroid Algorithm (Rocchio, 1971) and Neural Network as classification methods. *RapidMiner* and *Microsoft Visual Studio C++ 2010* have been used as implementation software.

For the application of algorithms and comparison of the results we have used the DEFT ("Défi Fouille de Texte") Evaluation Package 2008 (Proceedings of the 4th DEFT Workshop, 2008) which has been provided by ELRA and publically available corpora from DEFT'07 (Proceedings of the 3rd DEFT Workshop, 2007).

The main aim of this work is to evaluate the competitiveness of the novel term weighting (Gasanova et al., 2013) in comparison with the state-of-the-art techniques for opining mining and topic categorization. The criteria using in the evaluation are classification quality and computational efficiency.

This paper is organized as follows: in Section 2, we describe details of the corpora. Section 3 presents text preprocessing methods. In Section 4 we describe the classification algorithms which we have used to compare different text preprocessing techniques. Section 5 reports on the experimental results. Finally, we provide concluding remarks in Section 6.

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.

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 1. Corpora description (DEFT'07)

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

Table 2. Corpora description (DEFT'08)

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.

3 Text Preprocessing Methods

3.1 Binary preprocessing

We take each word of the document as a binary coordinate and the size of the feature space will be the size of our vocabulary ("bag of words").

3.2 TF-IDF

TF-IDF is a well-known approach for text preprocessing 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 and 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.

3) TF-IDF 3

$$idf_i = \left(\frac{|D|}{n_i}\right)^{\alpha}, \alpha \in (0,1), \quad (3)$$

where n_i is calculated as in TF-IDF 1 and α is the parameter (in this paper we have tested α = 0.1, 0.5, 0.9).

4) TF-IDF 4

The formula is given by equation (3) except n_i is calculated as in TF-IDF 4.

3.3 ConfWeight

Maximum Strength (Maxstr) is an alternative method to find the word weights. This approach has been proposed by Soucy and Mineau (2005). 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:

$$ConfWeight_{ij} = log(tf_{ij} + 1) * Maxstr(i).$$

Numerical experiments (Soucy and Mineau, 2005) 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.

3.4 Novel Term Weighting (TW)

The main idea of the method (Gasanova et al., 2013) 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 (Ishibuchi et al., 1999). 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 Classification Methods

We have considered 11 different text preprocessing methods (4 modifications of TF-IDF, two of them with three different values of α parameter, binary representation, ConfWeight and the novel TW method) and compared them using different classification algorithms. The methods have been implemented using *RapidMiner* (Shafait, 2010) and *Microsoft Visual Studio C++ 2010* for Rocchio classifier and SVM. The classification methods are:

- *k*-nearest neighbors algorithm with distance weighting (we have varied *k* from 1 to 15);
- kernel Bayes classifier with Laplace correction;
- neural network with error back propagation (standard setting in *RapidMiner*);
- Rocchio classifier with different metrics and γ parameter;

- support vector machine (SVM) generated and optimized with Co-Operation of Biology Related Algorithms (COBRA).

Rocchio classifier (Rocchio, 1971) is a well-known classifier based on the search of the nearest centroid. For each category we calculate a weighted centroid:

$$g_c = \frac{1}{|v_c|}\sum_{d\in v_c} d - \gamma \frac{1}{|\overline{v_{c,k}}|}\sum_{d\in \overline{v_{c,k}}} d,$$

where v_c is a set of documents which belong to the class c; $\overline{v_{c,k}}$ are k documents which do not belong to the class c and which are close to the centroid $\frac{1}{|v_c|}\sum_{d\in v_c} d$; γ is parameter corresponds to relative importance of negative precedents. The given document is put to the class with the nearest centroid. In this work we have applied Rocchio classifier with $\gamma \in (0.1; 0.9)$ and with three different metrics: taxicab distance, Euclidean metric and cosine similarity.

COBRA is a new meta-heuristic algorithm which has been proposed by Akhmedova and Semenkin (2013). It is based on cooperation of biology inspired algorithms such as Particle Swarm Optimization (Kennedy and Eberhart, 1995), Wolf Pack Search Algorithm (Yang, 2007), Firefly Algorithm (Yang, 2008), Cuckoo Search Algorithm (Yang and Deb, 2009) and Bat Algorithm (Yang, 2010). For generating SVM-machine the original COBRA is used: each individual in all populations represents a set of kernel function's parameters α, β, d. Then for each individual constrained modification of COBRA is applied for finding vector *w* and shift factor *b*. And finally individual that showed the best classification rate is chosen as the designed classifier.

5 Experimental Results

The DEFT ("Défi Fouille de Texte") Evaluation Package 2008 and publically available corpora from DEFT'07 (*Books*, *Games* and *Debates*) have been used for algorithms application and results comparison. In order to evaluate obtained results with the campaign participants we have to use the same measure of classification quality: precision, recall and F-score.

Precision for each class *i* is calculated as the number of correctly classified articles for class *i* divided by the number of all articles which algorithm assigned for this class. Recall is the number of correctly classified articles for class *i* divided by the number of articles that should have been in this class. Overall precision and recall are calculated as the arithmetic mean of the precisions and recalls for all classes (macro-average). F-score is calculated as the harmonic mean of precision and recall.

Tables 3-7 present the F-scores obtained on the test corpora. The best values for each problem are shown in bold. Results of the all classification algorithms are presented with the best parameters. We also present for each corpus only the best TF-IDF modification.

Classification algorithm	Binary	TF-IDF	Conf Weight	Novel TW
Bayes	0.489	0.506	0.238	0.437
k-NN	0.488	0.517	0.559	0.488
Rocchio	0.479	0.498	0.557	0.537
SVM (COBRA)	0.558	0.580	0.588	**0.619**
Neural network	0.475	0.505	0.570	0.493

Table 3. Classification results for *Books*

Classification algorithm	Binary	TF-IDF	Conf Weight	Novel TW
Bayes	0.653	0.652	0.210	0.675
k-NN	0.703	0.701	**0.720**	0.700
Rocchio	0.659	0.678	0.717	0.712
SVM (COBRA)	0.682	0.687	0.645	0.696
Neural network	0.701	0.679	0.717	0.691

Table 4. Classification results for *Games*

Classification algorithm	Binary	TF-IDF	Conf Weight	Novel TW
Bayes	0.555	0.645	0.363	0.616
k-NN	0.645	0.648	0.695	0.695
Rocchio	0.636	0.646	0.697	0.696
SVM (COBRA)	0.673	0.669	**0.714**	0.700
Neural network	0.656	0.647	0.705	0.697

Table 5. Classification results for *Debates*

Classification algorithm	Binary	TF-IDF	Conf Weight	Novel TW
Bayes	0.501	0.690	0.837	0.794
k-NN	0.800	0.816	0.855	0.837
Rocchio	0.794	0.825	0.853	0.838
SVM (COBRA)	0.788	0.827	0.840	**0.856**
Neural network	0.783	0.830	0.853	0.854

Table 6. Classification results for *T1*

Classification algorithm	Binary	TF-IDF	Conf Weight	Novel TW
Bayes	0.569	0.728	0.712	0.746
k-NN	0.728	0.786	0.785	0.811
Rocchio	0.765	0.825	0.803	0.834
SVM (COBRA)	0.794	0.837	0.813	**0.851**
Neural network	0.799	0.838	0.820	0.843

Table 7. Classification results for *T2*

We can see from the Tables 3-7 that the best F-scores have been obtained with either ConfWeight or novel Term Weighting preprocessing. The algorithm performances on the *Games* and *Debates* corpora achieved the best results with ConfWeight; however, we can see that the F-scores obtained with novel Term Weighting preprocessing are very similar (0.712 and 0.720 for *Games*; 0.700 and 0.714 for *Debates*). Almost all best results have been obtained with SVM except the *Games* database where we achieved the highest F-score with *k*-NN algorithm.

This paper focuses on the text preprocessing methods which do not require language or domain-related information; therefore, we have not tried to achieve the best possible classification quality. However, the result obtained on *Books* corpus with novel TW preprocessing and SVM (generated using COBRA) as classification algorithm has reached 0.619 F-score which is higher than the best known performance 0.603 (Proceedings of the 3rd DEFT Workshop, 2007). Performances on other corpora have achieved close F-score values to the best submissions of the DEFT'07 and DEFT'08 participants.

We have also measured computational efficiency of each text preprocessing technique. We have run each method 20 times using the Baden-Württemberg Grid (bwGRiD) Cluster Ulm (Every blade comprehends two 4-Core Intel Harpertown CPUs with 2.83 GHz and 16 GByte RAM). After that we calculated average values and checked statistical significance of the results.

Figure 1 and Figure 2 compare average computational time in minutes for different preprocessing methods applied on DEFT'07 and DEFT'08 corpora.

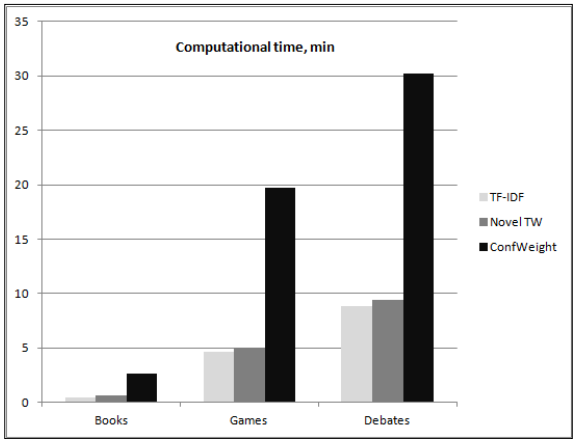

Figure 1. Computational efficiency of text preprocessing methods (DEFT'07)

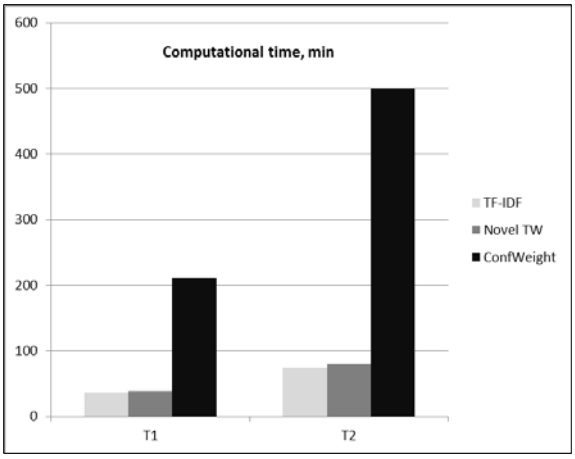

Figure 2. Computational efficiency of text preprocessing methods (DEFT'08)

The average value for all TF-IDF modifications is presented because the time variation for the modifications is not significant.

We can see in Figure 1 and Figure 2 that TF-IDF and novel TW require almost the same computational time. The most time-consuming method is ConfWeight (CW). It requires approximately six times more time than TF-IDF and novel TW for DEFT'08 corpora and about three-four times more time than TF-IDF and novel TW for DEFT'07 databases.

6 Conclusion

This paper reported on text classification experiments on 5 different corpora of opinion mining and topic categorization using several classification methods with different text preprocessing. We have used "bag of words", TF-IDF modifications, ConfWeight and the novel term weighting approach as preprocessing techniques. *K*-nearest neighbors algorithms, Bayes classifier, Rocchio classifier, support vector machine trained by COBRA and Neural Network have been applied as classification algorithms.

The novel term weighting method gives similar or better classification quality than the ConfWeight method but it requires the same amount of time as TF-IDF. Almost all best results have been obtained with SVM generated and optimized with Co-Operation of Biology Related Algorithms (COBRA).

We can conclude that numerical experiments have shown computational and classification efficiency of the proposed method (the novel TW) in comparison with existing text preprocessing techniques for opinion mining and topic categorization.

References

Akhmedova Sh. and Semenkin E. 2013. Co-Operation of Biology Related Algorithms. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2013)*:2207-2214.

Association Française d'Intelligence Artificielle. 2007. *Proceedings of the 3rd DEFT Workshop*. *DEFT '07*. AFIA, Grenoble, France.

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

Ishibuchi H., Nakashima T., and Murata T. 1999. Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Trans. on Systems, Man, and Cybernetics*, 29:601-618.

Kennedy J. and Eberhart R. 1995. Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*:1942-1948.

Le traitement automatique du langage naturel ou de la langue naturelle. 2008. *Proceedings of the 4th DEFT Workshop*. *DEFT '08*. TALN, Avignon, France.

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

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

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

Rocchio J. 1971. Relevance Feedback in Information Retrieval. *The SMART Retrieval System-Experiments in Automatic Document Processing*, Prentice-Hall:313-323.

Yang Ch. 2007. Algorithm of Marriage in Honey Bees Optimization Based on the Wolf Pack Search. *Proceedings of International Conference on Intelligent Pervasive Computing*:462-467.

Yang X.S. 2008. *Nature-Inspired Metaheuristic Algorithms*.

Yang X.S. and Deb S. 2009. Cuckoo search via Levy flights. *Proceedings of World Congress on Nature & Biologically Inspired Computing*:210-214.

Yang X.S. 2010. A New Metaheuristic Bat-Inspired Algorithm. *Proceedings of Nature Inspired Co-operative Strategies for Optimization (NISCO 2010)*:65-74.