Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 00 (2013) 000–000
Procedia Computer Science
www.elsevier.com/locate/procedia

Information Technology and Quantitative Management (ITQM2013)

The Role of Text Pre-processing in Sentiment Analysis

Emma Haddi[a], Xiaohui Liu[a], Yong Shi[b]

[a] Department of Information System and Computing, Brunel University, London,UB8 3PH, UK
[b] CAS Research Centre of Fictitious Economy & Data Science, Chinese Academy of Sciences, Beijing, 100080, PR China

Abstract

It is challenging to understand the latest trends and summarise the state or general opinions about products due to the big diversity and size of social media data, and this creates the need of automated and real time opinion extraction and mining. Mining online opinion is a form of sentiment analysis that is treated as a difficult text classification task. In this paper, we explore the role of text pre-processing in sentiment analysis, and report on experimental results that demonstrate that with appropriate feature selection and representation, sentiment analysis accuracies using support vector machines (SVM) in this area may be significantly improved. The level of accuracy achieved is shown to be comparable to the ones achieved in topic categorisation although sentiment analysis is considered to be a much harder problem in the literature.

Selection and/or peer-review under responsibility of the organizers of the 2013 International Conference on Computational Science

Keywords: Sentiment Analysis; Text Pre-processing; Feature Selection; Chi Squared; SVM.

1. Introduction

Sentiment analysis in reviews is the process of exploring product reviews on the internet to determine the overall opinion or feeling about a product. Reviews represent the so called user-generated content, and this is of growing attention and a rich resource for marketing teams, sociologists and psychologists and others who might be concerned with opinions, views, public mood and general or personal attitudes [1].

The huge number of reviews on the web represents the current form of user's feedback. It is hard for humans or companies to get the latest trends and summarise the state or general opinions about products due to the big diversity and size of social media data, and this creates the need of automated and real time opinion extraction and mining. Deciding about the sentiment of opinion is a challenging problem due to the subjectivity factor which is essentially what people think.

Sentiment analysis is treated as a classification task as it classifies the orientation of a text into either positive or negative. Machine learning is one of the widely used approaches towards sentiment classification in addition to lexicon based methods and linguistic methods [2]. It has been claimed that these techniques do not perform as well in sentiment classification as they do in topic categorisation due to the nature of an opinionated text which requires more understanding of the text while the occurrence of some keywords could be the key for an accurate classification [3]. Machine learning classifiers such as naive Bayes, maximum entropy and support vector machine (SVM) are used in [3] for sentiment classification to achieve accuracies that range from 75% to 83%, in comparison to a 90% accuracy or higher in topic based categorisation.

In [4], SVM classifiers are used for sentiment analysis with several univariate and multivariate methods for feature selection, reaching 85-88% accuracies after using chi-squared for selecting the relevant attributes in the texts. A network-based feature selection method that is feature relation networks (FRN) helped improve the performance of the classifier to

88-90% accuracies [4], which is the highest accuracy achieved in document level sentiment analysis to the best of our knowledge.

In this paper, we explore the role of text pre-processing in sentiment analysis, and report on experimental results that demonstrate that with appropriate feature selection and representation, sentiment analysis accuracies using SVM in this area may be improved up to the level achieved in topic categorisation, often considered to be an easier problem.

2. Background

There exist many studies that explore sentiment analysis which deal with different levels of the analysed texts, including word or phrase [5-6], sentence [7-8], and document level [9-10-4], in addition to some studies that are carried out on a user level [11-12]. Word level sentiment analysis explore the orientation of the words or phrases in the text and their effect on the overall sentiment, while sentence level considers sentences which express a single opinion and try to define its orientation. The document level opinion mining is looking at the overall sentiment of the whole document, and user level sentiment searches for the possibility that connected users on the social network could have the same opinion [12].

There exist three approaches towards sentiment analysis; machine learning based methods, lexicon based methods and linguistic analysis [2]. Machine learning methods are based on training an algorithm, mostly classification on a set of selected features for a specific mission and then test on another set whether it is able to detect the right features and give the right classification. A lexicon based method depends on a predefined list or corpus of words with a certain polarity. An algorithm is then searching for those words, counting them or estimating their weight and measuring the overall polarity of the text [13-11]. Lastly the linguistic approach uses the syntactic characteristics of the words or phrases, the negation, and the structure of the text to determine the text orientation. This approach is usually combined with a lexicon based method [8-2].

Pre-processing

Pre-processing the data is the process of cleaning and preparing the text for classification. Online texts contain usually lots of noise and uninformative parts such as HTML tags, scripts and advertisements. In addition, on words level, many words in the text do not have an impact on the general orientation of it.

Keeping those words makes the dimensionality of the problem high and hence the classification more difficult since each word in the text is treated as one dimension. Here is the hypothesis of having the data properly pre-processed: to reduce the noise in the text should help improve the performance of the classifier and speed up the classification process, thus aiding in real time sentiment analysis.

The whole process involves several steps: online text cleaning, white space removal, expanding abbreviation, stemming, stop words removal, negation handling and finally feature selection. All of the steps but the last are called *transformations*, while the last step applying some functions to select the required patterns is called *filtering* [14].

Features in the context of opinion mining are the words, terms or phrases that strongly express the opinion as positive or negative. This means that they have a higher impact on the orientation of the text than other words in the same text. There are several methods that are used in feature selection, where some are syntactic, based on the syntactic position of the word such as adjectives, and some are univariate, based on each feature's relation to a specific category such as chi squared (χ^2) and information gain, and some are multivariate using genetic algorithms and decision trees based on features subsets [4].

There are several ways to assess the importance of each feature by attaching a certain weight in the text. The most popular ones are: feature frequency (FF), Term Frequency Inverse Document Frequency (TF-IDF), and feature presence (FP). FF is the number of occurrences in the document. TF-IDF is given by

$$TF - IDF = FF * Log(N/DF) \tag{1}$$

where N indicates the number of documents, and DF is the number of documents that contains this feature [15]. FP takes the value 0 or 1 based on the feature absent or presence in the document.

Support Vector Machine

SVM [16] has become a popular method of classification and regression for linear and non linear problems [17]. This method tries to find the optimal linear separator between the data with a maximum margin that allows positive values above the margin and negative values below it. This problem is described as a "quadratic programming optimisation problem" [18].

Let $\{(x_{11},y_1),(x_{12},y_2),\ldots,(x_{mn},y_m)\}$ denote the set of training data, where x_{ij} denotes the occurrences of the events *j* in time *i*, and $y_i \in \{-1,1\}$. A support vector machine algorithm is solving the following quadratic problem:

$$\min_{w,b} \frac{1}{2} w^2 + C * \sum_{i=1}^{n} \varepsilon_i \;\; st\, \forall\, i: y_i\big(\langle w, x_{ij}\rangle + b\big) \geq 1 - \varepsilon_i \quad \varepsilon \geq 0 \tag{2}$$

where ε_i are the slack variables in which there are non-separable case and C>0 is the soft margin which controls the differences between margin b and the sum of errors. In other words, it performs a penalty for the data in the incorrect side of classification (misclassified), this penalty rises as the distance to the margin rises. w is the slope of the hyper plane which separates the data [19].

The speciality of SVM comes from the ability to apply a linear separation on the high dimension non-linear input data, and this is gained by using an appropriate kernel function [20]. SVM effectiveness is often affected by the types of kernel function that are chosen and tuned based on the characteristics of the data.

3. Framework

We suggest a computational frame for sentiment analysis that consists of three key stages. First, most relevant features will be extracted by employing extensive data transformation, and filtering. Second, the classifiers will be developed using SVM on each of the feature matrices constructed in the first step and the accuracies resulting from the prediction will be computed, and third the classifier's performance will be evaluated against other approaches.

The most challenging part of the framework is feature selection and here we discuss it in some depth. We will start by applying transformation on the data, which includes HTML tags clean up, abbreviation expansion, stopwords removal, negation handling, and stemming, in which we use natural language processing techniques to perform them. Three different feature matrices are computed based on different feature weighting methods (FF, TF-IDF and FP). We then move to the filtering process where we compute the chi-squared statistics for each feature within each document and choose a certain criterion to select the relevant features, followed by the construction of other features matrices based on the same previous weighting methods.

The data consist of two data sets of movie reviews, where one was first used in [3] containing 1400 documents (700 positive and 700 negative)(Dat-1400), and the other was constructed in [21-4] with 2000 documents (1000 positive, 1000 negative)(Dat-2000). Both sets are publicly available. Although the first set is included in the second set they were treated separately because the set of features that could influence the text are different. Furthermore this separation allows a fair comparison with different studies that used them separately. The features type used in this study is unigrams. We process the data as follows.

3.1. Data Transformation

The text was already cleaned from any HTML tags. The abbreviations were expanded using pattern recognition and regular expression techniques, and then the text was cleaned from non-alphabetic signs. As for stopwords, we constructed a stoplist from several available standard stoplists, with some changes related to the specific characteristics of the data. For example the words *film, movie, actor, actress, scene* are non-informative in movie reviews data. They were considered as stop words because they are movie domain specific words.

As for negation, first we followed [3] by tagging the negation word with the following words till the first punctuation mark occurrence. This tag was used as a unigram in the classifier. By comparing the results before and after adding the tagged negation to the classifier there was not much of a difference in the results. This conclusion is consistent with the findings of [22]. The reason is that it is hard to find a match between the tagged negation phrases among the whole set of documents. For that reason, we reduced the tagged words after the negation to three and then to two words taking in account the syntactic position, and this allowed more negation phrases to be included as unigrams in the final set of reduced features.

In addition, stemming was performed on the documents to reduce redundancy. In Dat-1400 the number of features was reduced from 10450 to 7614, and in Dat-2000 it was reduced from 12860 to 9058 features.

After that three feature matrices were constructed for each of the datasets based on three different types of features weighting: TF-IDF, FF, and FP. To make clear, in the FF matrix, the (i,j)-th entry is the FF weight of feature i in document j. Sets of experiments were carried out on the feature matrices of Dat-1400, which will be shown in Section 4.

3.2. Filtering

The method we are using for filtering is the univariate method chi-squared. It is a statistical analysis method used in text categorisation to measure the dependency between the word and the category of the document it is mentioned in. If the word is frequent in many categories, chi-squared value is low, while if the word is frequent in few categories then chi-squared value is high.

In this stage the value of chi-squared test was computed for each feature of the resulted features from the first stage. After that, based on a 95% significance level of the value of chi-squared statistics, a final set of features was selected in both datasets, resulting in 776 out of 7614 features in Dat-1400, and 1222 out of 9058 features in Dat-2000. The two sets were used to construct the features matrices on which the classification was conducted. At this stage each data set has three feature matrices: FF, TF-IDF, and FP.

3.3. Classification Process

After constructing the above mentioned matrices we apply SVM classifier on each stage. We chose the Gaussian radial basis kernel function which has the parameter γ that controls for the area in which the support vector has an effect in the data space. SVM was applied by using the machine learning package 'e1071' in R. We applied the SVM with different combination of the C and γ, due to the sensitivity of SVM performance to their values. For the classification process, each set was divided into two parts one for training and the other for testing, by ratio 4:1, that is 4/5 parts were used for training and 1/5 for testing. Then training was performed with 10 folds cross validation for classification.

3.4. Performance Evaluation

The performance metrics used to evaluate the classification results are precision, recall and F-measure. Those metrics are computed based on the values of true positive (*tp*), false positive (*fp*), true negative (*tn*) and false negative (*fn*) assigned classes. Precision is the number of true positive out of all positively assigned documents, and it is given by

$$\boldsymbol{precision} = \frac{\boldsymbol{tp}}{\boldsymbol{tp+fp}} \tag{3}$$

Recall is the number of true positive out of the actual positive documents, and it is given by

$$recall = \frac{tp}{tp+fn} \tag{4}$$

Finally F-measure is a weighted method of precision and recall, and it is computed as

$$F - measure = \frac{2 * precision * recall}{precision + recall} \tag{5}$$

where its value ranges from 0 to 1 and indicates better results the closer it is to 1.

4. Experiments and Results

In this section we report the results of several experiment to assess the performance of the classifier. We run the classifier on each of the features matrices resulting from each data transformation and filtering and compare the performance to the one achieved by running the classifier on non-processed data based on accuracies and Equation 5. Furthermore we compare those results to the reported results in [3-4] based on the accuracies and features type.

It is argued in [21] that "standard machine learning classification techniques, such as support vector machines (SVMs), can be applied to the entire documents themselves" and this is why [3-21] apply the classifier on the entire texts with no pre-processing or feature selection methods. Therefore, to allow a fair comparison with other results based on the tuned kernel parameters we are using in this stage, γ=0.001 and C=10, we classified the documents without any pre-processing. Then we applied the classifier on the Dat-1400 features matrix resulting from the first stage of pre-processing.

Table 1 compares the classifier performances resulting from the classification on both not pre-processed and pre-processed data for each of the features matrices (TF-IDF, FF, FP). Furthermore it compares these results to those that are achieved in [3] for both TF-IDF and FF matrices. The comparison is based on the accomplished accuracies and the metrics calculated in Equations 3,4,5 .

Table 1: The classification accuracies in percentages on Dat-1400, the column no pre-proc refers to the results reported in [3], no pre-proc2 refers to our results with no pre-processing, and pre-proc refers to the results after pre-processing, with optimal parameters $\gamma=10^{-3}$, and C=10

	TF-IDF		*FF*			*FP*		
	no pre-proc	pre-proc	no pre-proc1	no pre-proc2	pre-proc	no pre-proc1	no pre-proc2	pre-proc
Accuracy	78.33	81.5	72.7	76.33	83	82.7	82.33	83
Precision	76.66	83	NA	77.33	80	NA	80	82
Recall	79.31	80.58	NA	76.31	85.86	NA	83.9	83.67
F-Measure	77.96	81.77	NA	76.82	82.83	NA	81.9	82.82

Table 1 shows the that for the data that was not a subject to pre-processing, a good improvement occurred on the accuracies of the FF matrix, from 72.8% reported in [3] to 76.33%, while the accuracies of the FP matrix were slightly different, we achieved 82.33% while [3] reported 82.7%. In addition we obtained 78.33% accuracy in TF-IDF matrix where [3] did not use TF-IDF. By investigating further in the results we notice the increase in the accuracies when applying the classifier on the pre-processed data after the data transformation, with a highest accuracy of 83% for both matrices FF and FP.

Table 1 shows that although the accuracy accomplished in the FP matrix is close to the one achieved before and in [3], there is a big amendment in the classifier performance on the TF-IDF and FF matrices, and this shows the importance of stemming and removing stopwords in achieving higher accuracy in sentiment classification. We emphasise that to be able to use the SVM classifier on the entire document, one should design and use a kernel for that particular problem [23].

After that we classify the three different matrices that were constructed after the filtering (chi-squared feature selection). The accomplishments (see Table 2) of the classifier were high comparing to what was achieved in previous experiment and in [3]. Selecting the features based on their chi squared statistics value helped reducing the dimensionality and the noise in the text, allowing a high performance of the classifier that could be comparable to topic categorisation.

Table 2 presents the accuracies and evaluation metrics of the classifier performance before and after chi squared was applied.

Table 2: The classification accuracies in percentages before and after using chi-squared on Dat-1400, with optimal parameters $\gamma=10^{-5}$, and C=10

	TF-IDF		*FF*		*FP*	
	no chi	chi	no chi	Chi	no chi	Chi
Accuracy	81.5	92.3	83	90	83	93
Precision	83	93.3	80	92	82	94
Recall	80.58	91.5	85.86	88.5	83.67	92.16
F-Measure	81.77	92.4	82.83	90.2	82.82	93.06

Table 2 shows a significant increase in the quality of the classification, with the highest accuracy of 93% achieved in the FP matrix, followed by 92.3% in TF-IDF and 90.% in FF matrices, likewise the F-measure results is very close to 1, and that indicates a high performance of the classification. To the best of our knowledge, those results were not reported in document level sentiment analysis using chi-squared in previous studies. Hence, the use of transformation and then filtering on the texts data reduces the noise in the texts and improves the performance of the classification. Figure 1 shows how the prediction accuracies of SVM gets higher the fewer the number of features is.

A feature relation networks selection based method (FRN) was proposed in [4] to select relative features from Dat-2000 and improve the sentiment prediction using SVM. The accuracy achieved using FRN 89.65% , in comparison to an accuracy of 85.5% they achieved by using chi-squared method among some other univariate and multivariate feature selection methods.

We pre-processed Dat-2000, then ran SVM classifier, and we deliver a high accuracy of 93.5% in TF-IDF matrix followed by 93% in FP and 90.5% in FF (see Table 3), and that is as well higher than what was found in [4].

Table 3: Best accuracies in percentages resulted from using chi-squared on 2000 documents, with optimal parameters $\gamma=10^{-6}$, and $C=10$

	TF-IDF	*FF*	*FP*
Accuracy	93.5	90.5	93
Precision	94	89.5	91
Recall	93.06	91.3	94.79
F-Measure	93.53	90.4	92.87

The features that were used in [4] are of different types including different N-grams categories such as words, POS tags, Legomena and so on, while we are using unigrams only. We have demonstrated that using unigrams in the classification has a better effect on the classification results in comparison to other feature types, and this is consistent with the findings of [3].

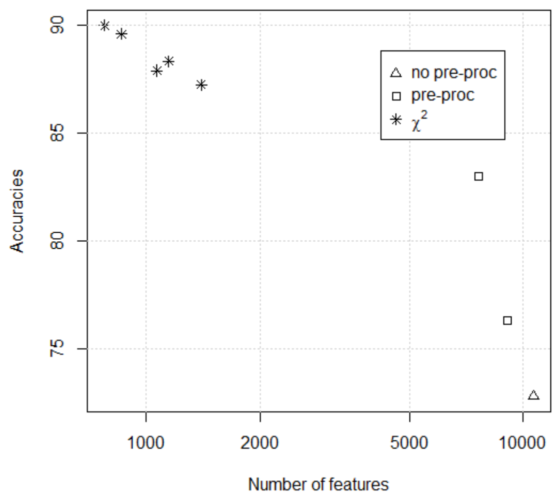

Figure 1: The correlation between accuracies and the number of features, no pre-proc refers to the results in [3], pre-proc and $\chi2$ refers to our results

5. Conclusion and Future Work

Sentiment analysis emerges as a challenging field with lots of obstacles as it involves natural language processing. It has a wide variety of applications that could benefit from its results, such as news analytics, marketing, question answering, knowledge bases and so on. The challenge of this field is to develop the machine's ability to understand texts as human readers do. Getting important insights from opinions expressed on the internet especially from social media blogs is vital for many companies and institutions, whether it is in terms of product feedback, public mood, or investors' opinions.

In this paper we investigated the sentiment of online movie reviews. We used a combination of different pre-processing methods to reduce the noise in the text in addition to using chi-squared method to remove irrelevant features that do not affect its orientation. We have reported extensive experimental results, showing that, appropriate text pre-processing methods including data transformation and filtering can significantly enhance the classifier's performance. The level of accuracy achieved on the two data sets is comparable to the sort of accuracy that can be achieved in topic categorisation, a much easier problem.

Financial blogs are a source of a different type of text data in the huge pool of social media. How investors' sentiments are correlated to stock prices fluctuation and how can the investor opinion be translated into a signal for buying or selling are the type of issues we will address in the future in the form of sentiment analysis of investors' opinions.

References

[1] H. Tang, S. Tan, X. Cheng, A survey on sentiment detection of reviews, Expert Systems with Applications 36 (7) (2009) 10760–10773.

[2] M. Thelwall, K. Buckley, G. Paltoglou, Sentiment in twitter events, Journal of the American Society for Information Science and Technology 62 (2) (2011) 406–418.

[3] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? sentiment classification using machine learning techniques, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002.

[4] A. Abbasi, S. France, Z. Zhang, H. Chen, Selecting attributes for sentiment classification using feature relation networks, IEEE Transactions on Knowledge and Data Engineering, 23 (3) (2011) 447–462.

[5] P. Tetlock, M. Saar-Tsechansky, S. Macskassy, More than words: Quantifying language to measure firms' fundamentals, The Journal of Finance 63 (3) (2008) 1437–1467.

[6] T. Wilson, J. Wiebe, P. Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, in: Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), 2005, pp. 347–354.

[7] H. Yu, V. Hatzivassiloglou, Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences, in: Proceedings of the conference on Empirical methods in natural language processing, EMNLP-2003, 2003, pp. 129–136.

[8] L. Tan, J. Na, Y. Theng, K. Chang, Sentence-level sentiment polarity classification using a linguistic approach, Digital Libraries: For Cultural Heritage, Knowledge Dissemination, and Future Creation (2011) 77–87.

[9] S. R. Das, News Analytics: Framework, Techniques and Metrics, Wiley Finance, 2010, Ch. 2, the Handbook of News Analytics in Finance.

[10] B. Pang, L. Lee, Seeing stars: Exploiting class relationship for sentiment categorization with respect to the rating scales, in: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL), 2005, pp. 115-124.

[11] P. Melville, W. Gryc, R. Lawrence, Sentiment analysis of blogs by combining lexical knowledge with text classification, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 1275–1284.

[12] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, P. Li, User-level sentiment analysis incorporating social networks, Arxiv preprint arXiv:1109.6018.

[13] X. Ding, B. Liu, P. Yu, A holistic lexicon-based approach to opinion mining, in: Proceedings of the international conference on Web search and web data mining, ACM, 2008, pp. 231–240.

[14] I. Feinerer, K. Hornik, D. Meyer, Text mining infrastructure in r, Journal of Statistical Software 25 (5) (2008) 1–54.

[15] J.-C. Na, H. Sui, C. Khoo, S. Chan, Y. Zhou, Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews, in: Conference of the International Society for Knowledge Organization (ISKO), 2004, pp. 49–54.

[16] V. Vapnik, The nature of statistical learning theory, springer, 1999.

[17] C. Lee, G. Lee, Information gain and divergence-based feature selection for machine learning-based text categorization, Information processing & management 42 (1) (2006) 155–165.

[18] S. Russell, P. Norving, Artificial Intelligence: A Modern Approach, second edidtion Edition, Prentice Hall Artificial Intelligence Series, Pearson Education Inc., 2003.

[19] J. Wang, P. Neskovic, L. N. Cooper, Training data selection for support vector machines, in: ICNC 2005. LNCS, International Conference on Neural Computation, 2005, pp. 554–564.

[20] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, R. Williamson, Estimating the support of a high-dimensional distribution, Neural computation 13 (7) (2001) 1443–1471.

[21] B. Pang, L. Lee, A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, in: Proceedings of the ACL, 2004.

[22] K. Dave, S. Lawrence, D. M. Pennock, Mining the peanut gallery: Opinion extraction and semantic classification of product reviews, in: Proceedings of WWW, 2003, p. 519–528.

[23] B. Scholkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, V. Vapnik, Comparing support vector machines with gaussian kernels to radial basis function classifiers, Signal Processing, IEEE Transactions on 45 (11) (1997) 2758–2765.