Prior Polarity Dictionary Derived from SentiWordNet based on Random Forest Algorithm

Xiaobin Li[1, a], YongQuan Dong[1, b], Gai-Ge Wang[1, c], Mo Hou[1, d]

[1]School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, 221116, China

[a]email: wb0817002@163.com, [b]email: tomdyq@163.com

[c]email: gaigewang@gmail.com, [d]email: 469954990@qq.com

Keywords: Sentiment Analysis; Sentiment Strength; Support Vector Regression; Random Forest

Abstract. The prior polarity of words, as a challenging problem, can make great contribution to the sentiment analysis task. In this paper, we propose a method to generate the prior polarity dictionary based on Random Forest (RF) learning algorithm. We compare the proposed approach with the previous methods. The experimental results show that it is better than the state-of-art Support Vector Regression (SVR) method and it can gain better performance.

Introduction

With the great growth of the amount of data, sentiment analysis with manual labor of these massive emotional data is a very expensive and almost impossible task. Using the technology of computer aided sentiment analysis came into being. At present, according to the sentiment analysis granularity of user comments, the sentiment analysis techniques can be divided into several levels, such as words, sentences, documents and so on. Sentiment dictionary can directly judge the polarity of words. Sentences and documents sentiment analysis performance can also be improved with the help of sentiment dictionary [1]. At the same time several sentiment dictionaries are widely applied in the area of sentiment analysis. These sentiment dictionaries include SentiWordNet, General Inquirer, Opinion Lexicon and so on [2, 3].

The prior polarity of a word can give the word a sentiment score where users do not need consider the various meanings or the context of the word. There are two basic strategies for the acquisition of the polarity of a word, one is a manual annotation. The other is to acquire knowledge from the existing dictionary. Although the precision performance of the latter one is reduced, the annotation cost is lower than the previous one. Also the prior polarity annotation method based dictionary can produce a large number of words sentiment score efficiently.

Inspired by the paper [4], we put forward a novel method to obtain prior polarity dictionary with Random Forest algorithm. Combined with various sentiment calculation formulas the algorithm can get a sentiment dictionary with higher performance. The experimental results demonstrate that there are two distinct merits: (I) the method we proposed can obtain a higher precision than the previous methods; (II) the proposed method can get a higher classification performance in micro blogs than the previous method.

The paper is structured as follows: in Section 2 we briefly describe the related work. Then, in Section 3 we present our proposed approach. In Section 4 we present a series of experiments to verify our approach. Finally, in Section 5 we give our conclusion.

Related Work

The sentiment strength of a word is the emotion degree, which is more accurate than the sentiment polarity. For example, sentiment of the word "good" is weaker than the word “excellent”. This is particularly useful in the field of context-free sentiment analysis. In [5], the authors proposed an approach to build a sentiment strength dictionary with semantic graph in WordNet where the semantic distance is calculated between the target word and the reference word. In [6], the search engine and recursive rules are used to construct the progressive relationship among

words. The sentiment strength of each word can be obtained based on the connection analysis method in these words. In [7], the authors build a sentiment pool with some seed words and calculate the sentiment strength of all words in WordNet with the semantic relation in WordNet. In [8], the authors construct a sentiment dictionary based on SentiWordNet. The proposed research focused on raising the performance of SentiWordNet3.0 by using it as a labeled corpus to build another sentiment lexicon, named Senti-CS. The part of speech information, usage based ranks, and sentiment scores are used to calculate Chi-Square-based feature weight for each unique subjective term/part-of-speech pair extracted from SentiWordNet3.0. This weight is then normalized in a range between -1 and +1 using min-max normalization. In [9], the authors proposed a semi-supervised sentiment analysis approach that incorporates lexicon-based methodology with machine learning in order to improve sentiment analysis performance. Mathematical models such as information gain and cosine similarity are employed to revise the sentiment scores defined in SentiWordNet. This research also emphasizes the importance of nouns and employs them as semantic features with other parts of speech.

In [10], the authors compare 14 formulas that appear in the previous literatures, and assess which one best approximates the human judgment of prior polarities, with both regression and classification models. The experiments demonstrate that weighted average over word senses is the strategy that best approximates human judgment. Some authors follow the previous work and put forward the method based on support vector machine to obtain the sentiment strength of words [4, 11]. The experiments demonstrate a better performance than the weighted average method in [10].

Random Forest to Generate Prior Polarity

This paper will recalculate 14 kinds of sentiment strength calculation methods proposed in [10]. During the training phase, the 14 kinds of sentiment strength will be input to the training model. In order to evaluate the model, we will test these models in some public available sentiment annotation dataset and micro blog datasets.

Features.

The format of each lemma in SentiWordNet is[# POS,ID,posScore,negScore,SynsetTerms,Gloss]. The aim of the paper is to give the sentiment strength of each word without consideration of many meanings of the word or the context of the word.

Firstly two equations (1) and (2) are defined as follows.

$$f_d = |\text{posScore}| - |\text{negScore}| \quad (1)$$

$$\text{f}_\text{m} = \text{Sign}(f_d)\max(|\text{posScore}|, |\text{negScore}|) \quad (2)$$

There are 8 basic sentiment strength calculation strategies for a lemma#pos [10], where all strategies are listed in Table 1.

For the calculation methods of the above features, in addition to the calculation methods of Rnd and Uni, the other 6 kinds of sentiment calculation methods in PosScore and NegScore were calculated respectively by the Eqs. (1)-(2). They can be extended to 12 distinct features. At last, there will be 14 different sentiment strength score which will be input as the features to the training model during learning phrase.

Table1 Sentiment strength calculation strategies

Feature	Calculation strategies	Description
Rnd	$\text{Rnd} = \text{random}(-1,1)$	Feature Rnd is a random value between -1 and 1 is given to specific lemma#pos
Uni	(a) posScore $= \frac{\sum_{i=1}^{n} posScore_i}{numPos}$ (b) negScore $= \frac{\sum_{i=1}^{n} negScore_i}{numNeg}$ (c) $s = Sign(numPos - numNeg)$ Uni $= s * f_m(posScore, negScore)$	numPos denotes the number of $posScore_i$ not equal to 0; numNeg denotes the number of $negScore_i$ not equal to 0; n denotes the number of different meanings of lemma#pos; Feature Uni is the maximum value of the average sentiment score.
Swrnd	(a) posScore $= posScore_i$ (b) negScore $= negScore_i$	$i = random(1, n)$, n denotes the number of different meanings of lemma#pos;
Fs	(c) posScore $= posScore_1$ (d) negScore $= negScore_1$	The first meaning of lemma#pos is appointed. posScore and negScore are given according to the meaning.
Mean	(a) posScore $= \frac{\sum_{i=1}^{n} posScore_i}{n}$ (b) negScore $= \frac{\sum_{i=1}^{n} negScore_i}{n}$	Two average sentiment score are calculated on lemma#pos. Then the Eqs. (1)-(2) are used to compute two Mean features.
Senti	(a) posScore $= \frac{\sum_{i=1}^{n} posScore_i}{numPos}$ (b) negScore $= \frac{\sum_{i=1}^{n} negScore_i}{numNeg}$	numPos denotes the number of $posScore_i$ not equal to 0; numNeg denotes the number of $negScore_i$ not equal to 0;
W1	(a) posScore $= \frac{\sum_{i=1}^{n} \frac{1}{2^{i-1}} posScore_i}{n}$ (b) negScore $= \frac{\sum_{i=1}^{n} \frac{1}{2^{i-1}} negScore_i}{n}$	n denotes the number of different meanings of lemma#pos; The sentiment score of each meaning of lemma#pos is weight average as the geometric series. The coefficients are assigned with 1/2.
W2	(a) posScore $= \frac{\sum_{i=1}^{n} \frac{1}{i} posScore_i}{n}$ (b) negScore $= \frac{\sum_{i=1}^{n} \frac{1}{i} negScore_i}{n}$	n denotes the number of different meanings of lemma#pos; The sentiment score of each meaning of lemma#pos is weight average as the harmonic series. The coefficients are assigned with 1/2.

Learning algorithm.

The above 14 calculated feature values can be used as the sentiment strength score. At the same time, in [4], these features are input to support vector machine model to obtain the best prior polarity dictionary. The next section of our paper will apply Random Forest model to obtain a priori polarity dictionary which has superior performance than the dictionary obtained in [4].

Random Forest is an ensemble learning tool, which integrates the results of multiple decision trees [12]. The algorithm is depicted in ALOGORITHM 1.

ALGORITHM 1: RF

Input:

T $= \{x_i, y_i\}_{i=1}^{\ell}$:$\ell$ labeled instances;

U $= \{x_i\}_{\ell+1}^{\ell+u}$:u unlabeled instances;

Process:

For b=1,…,B:

$\{x_{bi}, y_{bi}\}_{i=1}^{\ell} = \text{BootStrapSample}(\{x_i, y_i\}_{i=1}^{\ell})$

$DecisionTree_b = CART(\{x_{bi}, y_{bi}\}_{i=1}^{\ell})$

End

Output:

$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} DecisionTree_b(U)$

In Algorithm 1, the BootStrapSample sampling mainly focuses on two dimensions of sampling including random selection of sub samples and random selection of sub properties.

Sentiment classification for Micro blogs.

The main purpose of this paper is to evaluate the performance of Random Forest algorithm in generating sentiment dictionary, so there is no optimization work to improve the performance in sentiment classification. Here we only use a mean sentiment strength score of all words in micro blog to denote as the sentiment strength of the micro blog. The sentiment classification algorithm for micro blogs is shown in ALGORITHM 2.

ALGORITHM 2: SA

Input:

T : one Micro-blog with all lower character

D=$\{lemma_i, pos_i, score_i\}_{i=1}^{n}$: sentiment score lexica for lemma#pos

Process:

Score=0

Ts=Stemmer(T)

$\{w_i, pos_i\}_{i=1}^{m}$=Tokenizer(Ts)

For i=1,…,m

If $\{w_i, pos_i\}$ in D

Score=Score+$score_i$

End

End

Output:

$\hat{f} = Score/m$

The function of Stemmer is to get the trunk of the micro-blog text. The function of Tokenizer is to segment and give part of speech tagging for the micro-blog text. The algorithm will eventually return a score for a micro-blog. If the returned score is greater than 0, then the micro-blog is judged as positive one. Otherwise the micro-blog is judged as negative one.

Experiments

Datasets.

The sentiment dictionary used in the paper is SentiWordNet3.0 [2, 3] . SentiWordNet is a widely used sentiment dictionary in the field of sentiment analysis in which each entry lemma format is [# POS, ID, posScore, negScore, SynsetTerms, Gloss]. #POS column is part of speech; ID column is the identified code of the entry lemma. PosScore and NegScore columns are the positive and negative sentiment accordingly. The two value is in the range between 0 and 1. SynsetTerms#n columns are the meanings of number entry; Gloss column is the meaning of the word. The detail content is omitted in the table because of its length. There are totally 117659 records in SentiWordNet3.0.

In order to evaluate the performance of the previously generated sentiment strength dictionary an manual annotation dictionary Anew [13] is used in the experiments. In Anew the researchers require students to give a first look impression when reading a word. Valence, arousal, and dominance are the three dimensions of a word needed to appointed respectively. Finally, several statistical methods are applied to calculate the mean and standard deviation of the corresponding dimension. Description column is the word name. Word No. column is the identified number of the word. Valence (Arousal, Dominance) Mean (SD) column list the mean (standard deviation) values of Valence (Arousal, Dominance). The values of the 3 dimensions are between 1 and 9; Word Frequency column gives the number of words by voting. There are totally 1034 entries in Anew.

Metrics.

In order to evaluate the score of the corresponding entries, the MAE and Success metrics proposed in [10] are listed in Eqs. (3) and (4).

$$\text{MAE} = \frac{\sum_{i=1}^{n} |f(w_i) - Anew_{mean}(w_i)|}{n} \quad (3)$$

$$\text{Sucess} = \frac{\sum_{i=1}^{n} [|f(w_i) - Anew_{mean}(w_i)| < 0.5 Anew_{sd}(w_i)]}{n} \quad (4)$$

The MAE metric measures the error of the calculated prior sentiment strength score $f(w_i)$. The Success metric is mainly to determine whether the sentiment score $f(w_i)$ will fall within 0.5 times the standard deviation. From a statistical point of view, fall within the range of 0.5 times the difference measurement in the standard has larger probability coverage of real value. Obviously we wish to get a smaller MAE and bigger Success metric sentiment strength score. Then in the next section the Success/MAE will be used as usability metric for sentiment strength dictionary assert.

Discussion and Experimental Results.

In the experiment, the decision tree number B in Random Forest (RF) algorithm is set to 10. At the same time, we will set part of the data in the Anew dictionary as the input feature and the model is tested on all Anew data. The percentage of train data is represented by data per. Table 2 gives the comparisons result between RF and SVR. From the results listed in Table 2, we can make a conclusion that the RF algorithm is significantly better than the SVR algorithm according to the MAE, Success/MAE, the correct rate (ACC) and F1 metric respectively.

Table2 Sentiment strength comparison result between SVR and RF for Anew

	MAE	Success/MAE	ACC	F1
SVR(per=0.1)	0.3492	1.0289	0.6946	0.5155
RF(per=0.1)	**0.3425**	**1.1896**	**0.7089**	**0.5836**
SVR(per=0.5)	0.3421	1.1035	0.7083	0.5645
RF(per=0.5)	**0.2999**	**1.5665**	**0.7505**	**0.6547**
SVR(per=0.9)	0.3450	1.0697	0.7018	0.5487
RF(per=0.9)	**0.2629**	**1.9672**	**0.7836**	**0.6873**

In order to evaluate the sentiment classification performance of the dictionary generated in this paper, we apply the ALGORITHM 2 on the micro-blog STS-Test dataset and the micro-blog STS-Gold dataset [14].

Fig.1 shows the classification accuracy in the STS-Test dataset with the sentiment strength dictionary generated by using SVR and RF respectively. From the Fig.1, we can find that RF has higher classification accuracy than SVR for all data percentage.

Fig.2 shows the classification accuracy in the STS-Gold dataset with SVR and RF algorithm. There is mostly the same phenomenon as the classification accuracy in the STS-Test dataset.

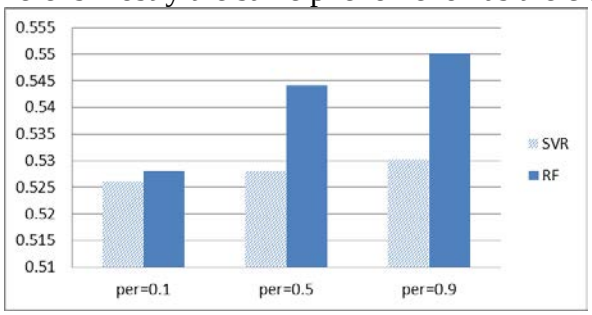

Fig.1 Classification Accuracy on STS-Test

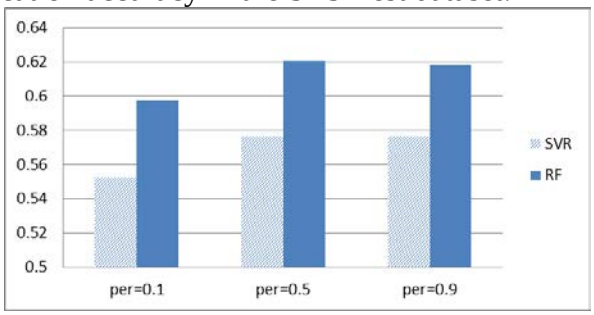

Fig.2 Classification Accuracy on STS-Gold

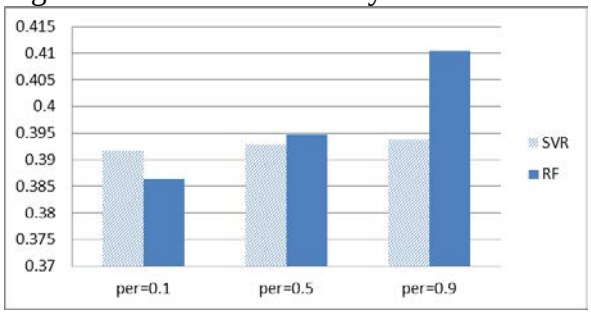

Fig.3 Classification F1 on STS-Test

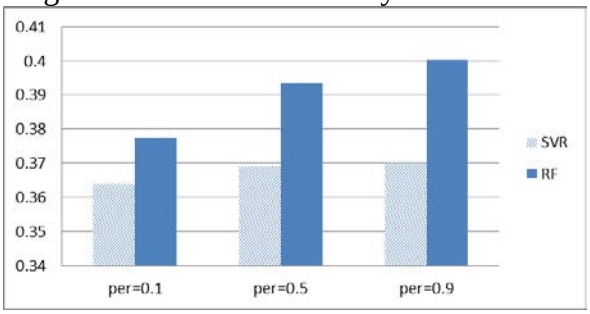

Fig.4 Classification F1 on STS-Gold

Fig.3 and Fig.4 show the classification F1 metric comparison for the STS-Test and the STS-Gold micro-blog data sets based on the two sentiment strength dictionary generated from the two algorithms respectively. From Fig.3 and 4, we can see that the SVR algorithm is slightly superior to STS-Test with a small amount of data (per=0.1). While more training data is fed into algorithm RF algorithm will have higher classification F1 metric than SVR algorithm. At the same time the performance gap between the two algorithms will become more lager with more training data is fed into the algorithms.

Conclusion

The prior polarity judgment of a word is a challenging task in the field of sentiment analysis. This paper proposed a novel prior polarity dictionary generated with Random Forest model. The method set some manual sentiment strength score as the input feature to the Random Forest model. Then the model will produce sentiment strength score for all words. The experimental results show that it is better than Support Vector Regression method and it can gain better performance.

Acknowledgements

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions. This work was supported by Research Fund for the Doctoral Program of Jiangsu Normal University (No. 14XLR036).

References

[1] Ke W, Rui X, Khan FH, Qamar U, Bashir S. A Survey on Automatical Construction Methods of Sentiment Lexicons[J]. ACTA AUTOMATICA SINICA. 2016(04):495-511.

[2] A E, F S. Sentiwordnet: a publicly available lexical resource for opinion mining[C]. Proceedings of the 2006 Language Resources and Evaluation. 2006.

[3] Baccianella S, Esuli A, Sebastiani F. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining[C]. International Conference on Language Resources & Evaluation. 2010:2200-4.

[4] Gatti L, Guerini M, Turchi M. SentiWords: Deriving a High Precision and High Coverage Lexicon for Sentiment Analysis[J]. IEEE Transactions on Affective Computing. 2016;7(4):409-21.

[5] Gbolahan KW, Sarabjot SA. Predicting the Polarity Strength of Adjectives Using WordNet[J]. International AAAI Conference on Web and Social Media; Third International AAAI Conference on Weblogs and Social Media. 2009.

[6] Lu Y, Kong X, Quan X, Liu W, Xu Y. Exploring the Sentiment Strength of User Reviews[C]. Web-Age Information Management: 11th International Conference, WAIM 2010, Jiuzhaigou, China, July 15-17, 2010 Proceedings. 2010:471-82.

[7] Kumar A, Sebastian TM. Sentiment Analysis on Twitter[J]. International Journal of Computer Science Issues. 2012;9(3):372-8.

[8] Khan FH, Qamar U, Bashir S. Senti-CS: Building a lexical resource for sentiment analysis using subjective feature selection and normalized Chi-Square-based feature weight generation[J]. EXPERT SYSTEMS. 2016;33(5):489-500.

[9] Khan FH, Qamar U, Bashir S. A semi-supervised approach to sentiment analysis using revised sentiment strength based on SentiWordNet[J]. Knowledge & Information Systems. 2016:1-22.

[10] Gatti L, Guerini M. Assessing Sentiment Strength in Words Prior Polarities[J]. Computer Science. 2012.

[11] Guerini M, Gatti L, Turchi M. Sentiment Analysis: How to Derive Prior Polarities from SentiWordNet[J]. Breast Cancer Immunodiagnosis & Immunotherapy. 2013:3-11.

[12] Breiman L. Random Forests[J]. Machine Learning. 2001;45(1):5-32.

[13] Bradley MM, Lang PJ. Affective Norms for English Words (ANEW): Instruction Manual and Affective Ratings[J]. Journal Royal Microscopical Society. 1999;88(1):630-4.

[14] Saif H, Fernandez M, He Y, Alani H, Fernandez M. Evaluation Datasets for Twitter Sentiment Analysis. A survey and a new dataset, the STS-Gold[C]. Workshop: Emotion and Sentiment in Social and Expressive Media: Approaches and Perspectives From Ai. 2013.