NDMSCS: A Topic-Based Chinese Microblog Polarity Classification System

Yang Wang, Yaqi Wang, Shi Feng, Daling Wang, Yifei Zhang
Northeastern University, Shenyang, China
{wangyangdm,wyqnumber1}@gmail.com,
{fengshi,wangdaling,zhangyifei}@ise.neu.edu.cn

Abstract

In this paper, we focus on topic-based microblog sentiment classification task that classify the microblog's sentiment polarities toward a specific topic. Most of the existing approaches for sentiment analysis usually adopt the target-independent strategy, which may assign irrelevant sentiments to the given topic. In this paper, we leverage the non-negative matrix factorization to get the relevant topic words and then further incorporate target-dependent features for topic-based microblog sentiment classification. According to the experiment results, our system (NDMSCS) has achieved a good performance in the SIGHAN 8 Task 2.

1 Introduction

Nowadays, people are willing to express their feelings and emotions via the microblog services, such as Twitter and Weibo. Therefore, the microblog has aggregated huge amount of sentences that contain people's rich sentiments. Extracting and analyzing the sentiments in microblogs has become a hot research topic for both academic communities and industrial companies.

The microblog usually has a length limitation of 140 characters, which leads to extremely sparse vectors for the learning algorithms. On the other hand, people are used to using a simple sentence, or even a few words to express their attitude or viewpoint toward a specific topic. Most of the existing sentiment analysis methods could classify the microblogs into positive, negative and neutral categories. However, these methods usually adopt the target-independent strategy, which may assign irrelevant sentiments to the given topic.

In this paper we develop a machine learning system for topic-based microblog polarity classification. Given a microblog and a topic, we intend to classify whether the microblog is of positive, negative, or neutral sentiment towards the given topic. For microblogs conveying both a positive and negative sentiment towards the topic, whichever is the stronger sentiment should be chosen.

To tackle challenges, firstly we use non-negative matrix factorization to find the topic relevant words. And then we propose feature selection strategy and construct vectors to convert the raw microblog text into the TFIDF feature values, combined with the linguistic features, which we then use together with the labels to train our sentiment classifier. Our approach includes an extensive usage of Python based NLP and machine learning resources for conducting word segmentation, POS tagging and classifier implementation.

We evaluate our proposed system on the test set of Topic-Based Chinese Message Polarity Classification Task in SIGHAN 8. Our system is ranked 3rd on the task test set for overall F1 value and also achieves good performance in the positive and negative F1 values. The experiment shows the effectiveness of our proposed system.

2 Non-negative Matrix Factorization

Topic based sentiment analysis task need to consider the target that sentiment words described, so we try to find the words related to the specific topic. And the topics of test set are different from the training set, so we want to use the wildcard to replace the topic words to reduce the influence of different topics. We consider using the topic modeling to discovery the hidden topic information in large collections of documents. People usually use the probabilistic methods, such as Latent Dirichlet allocation (LDA) (Blei et al., 2003), to build the topic model. However, an effective alternative is to use Non-negative Matrix Factorization (NMF) (Lee et al., 1999). NMF refers to an unsuper-

Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing (SIGHAN-8), pages 180–184,
Beijing, China, July 30-31, 2015.

vised family of algorithms from linear algebra that simultaneously performs dimension reduction and clustering.

NMF takes non-negative matrix as an input, and factorizes it into two smaller non-negative matrices *W* and *H*, each having k dimensions. When multiplied together, these factors approximate the original matrix X. It finds a decomposition of samples *X* into two matrices *W* and *H* of non-negative elements, by optimizing for the squared Frobenius norm:

$$arg\min_{W,H} \|X - WH\|^2 = \Sigma_{i,j} X_{i,j} - WH_{i,j} \quad (1)$$

We can specify the parameter k to control the number of topics that will be produced. The rows of the matrix *W* provides weights for the input documents relative to the k topics and these values indicate the strength of association between documents and topics. The columns of the matrix *H* provide weights for the terms relative to the topics. By ordering the values in a given column and selecting the top-ranked terms, we can produce a description of the corresponding topic.

NMF implements the Nonnegative Double Singular Value Decomposition(NNDSVD) which is proposed by Boutsidis et al. (2008). NNDSVD is based on two SVD processes, one approximating the data matrix, the other approximating positive sections of the resulting partial SVD factors utilizing an algebraic property of unit rank matrices. The basic NNDSVD algorithm is better fit for sparse factorization.

Once the document-term matrix *X* has been constructed, we apply NMF topic modeling as follows: First we initialize the value of k to 5 for training data and 20 for test data. We generate initial factors using the NNDSVD. Then we apply the NMF algorithm on the document-term matrix *X*, using the initial parameters from first step, for a fixed number of iterations (e.g. 1000) to produce final factors (*W*, *H*). Each row of *H* is a distribution over all terms in a vocabulary, and easily interpreted as the topics. In each topic we choose top-ranked terms as the topic words.

The data preparation and topic modeling described above can be implemented using the Python Scikit-learn[1] toolkit. We use TfidfVectorizer to create document-term matrix of size (d, t), and generate factor *W* of size (d, k) and factor *H* of size (k, t) by using NMF. Here d and t represent the number of documents and terms, and k represents the number of topics. We get the topic words in the training data as show in Table 1.

Topic ID	Topic Words
Topic 1	ssix, 三星, edge, galaxy, mnine,五千块, 给你, 还是
Topic 2	日本, 马桶盖, 中国, 杭州, 游客, 国内, 确系, 热销
Topic 3	降息, 央行, 基准利率, 下调, 百分点, 存款, 一年期, 贷款
Topic 4	油价, 令吉, 国油, 物价, 商家,公司, 调涨, 燃油
Topic 5	雾霾, 柴静, 穹顶, 之下, 调查, 视频, 同呼吸共命运, 完整版

Table 1: The topic words extracted from the document.

Because the documents carry a lot of noise and the NMF algorithm doesn't know anything about the documents, terms, or topics it contains, we manually inspect and remove the unrelated topic words. The words were discarded for various reasons: they were too generic, or irrelevant to the primary topic. In order to convert the problem to the topic independent emotion classification problem, we preprocess the microblog by replacing the topic words with TW and setting their POS tags to noun.

3 System Overview

Figure 1 gives a brief overview of our system that takes the microblogs and the corresponding labels as inputs to learn sentiment classifiers. We build a TFIDF-NMF pipeline to get the topic words after preprocessing. We use three-way classification framework in which we incorporate rich topic-dependent feature representations of the microblog text. The classifier is then used to predict test microblog sentiment labels. The proposed system basically include the module of preprocessing, topic word expansion, feature extraction and classification. In this section we discuss each module in detail.

3.1 Preprocessing

Handle Traditional Chinese Text: Some of the microblogs are written in traditional Chinese, so we first convert the traditional Chinese to the simplified Chinese based on the tool OpenCC[2], which

[1] http://scikit-learn.org/

[2] http://opencc.byvoid.com/

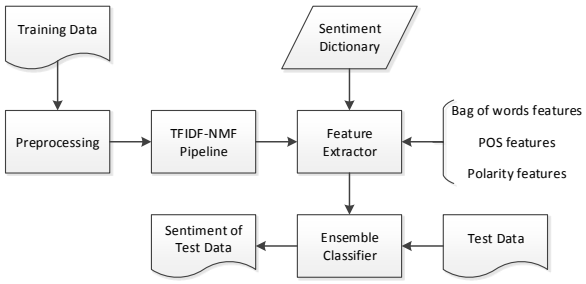

Figure 1: System Overview

is an open source project for character conversion between Traditional and Simplified Chinese.

Replace URLs: The URLs can lead the reader into the new webpages. These URLs do not carry much information about the sentiment. But they might help to identify whether the microblogs contain sentiment information. Thus we use 'http' to replace all the URLs in microblogs.

Remove Retweet Mentions: The retweet mentions in a microblog often start with '@', and are followed by people or organizations. This information is also unhelpful for the sentiment classification of the microblog. Hence they are removed.

Remove Unrelated Punctuations: Some punctuation such as single comma and colon are removed because they are unrelated to sentiment analysis. Some punctuation such as the question and exclamation mark could indicate people's sentiments, so we preserve them for further steps.

Remove numbers: Numbers are usually without any emotional information. Thus, numbers are removed in order to refine the microblog content. But there is a topic Samsung S6 in the training data, and we convert this topic to Samsung Ssix.

Text Segmentation: In the Chinese text analysis task, we need to consider the word as a unit. We use the Jieba[3] Chinese text processing tool to segment the Chinese microblogs into words. The words in sentiment lexicons are added into Jieba default dictionary, which could ensure a higher segmentation accuracy.

Remove Stop Words: Stop words are extremely common words. And stop words do not carry any sentiment information and thus are of no use.

Handle Unbalanced Data: In SIGHAN training dataset, the number of neutral microblogs is about 4 times bigger than that of the microblogs with emotions, which leads to serious unbalanced data. To tackle this problem, we oversample the microblogs with emotions to balance the dataset.

[3]https://github.com/fxsjy/jieba/

3.2 Baseline Model

SIGHAN provided two sentiment lexicon: NTUSD and DLUT Emotion Ontology. We combine the two lexicons, remove the duplicate words, and finally we get 14,828 positive words and 20,366 negative words in the new lexicon.

We first perform the preprocessing steps listed in Section 3.1 and for each sentence we count the number of positive and negative sentiment words. Simple Sentiment Word-Count Method (SSWCM) (Yuan et al., 2013) is an intuitively basic algorithm for sentiment classification. The polarity of text is determined by the number of sentiment words. If the number of positive words is larger than negative words, we will classify the text as the positive polarity. If the number of positive words is less than negative words, we will classify the text as the negative polarity. In other cases, the text is classified as the neutral polarity.

3.3 Feature Extraction

The feature extraction process is a key component for sentiment analysis. The feature vector consists of bag of words features, POS features and polarity features.

Bag of Words Features: We use unigram, bigrams and trigrams as features and the TFIDF as the weighting scheme based on the bag-of-words model. TFIDF is a term weighting scheme developed for information retrieval originally, that has also achieved good performance in document classification and clustering tasks.

Part of Speech Features: We use Jieba Part of Speech Tokenizer, which tags the POS of each word after segmentation. The feature vector uses POS tags to express of how many nouns, verbs, adjectives, hashtags, emoticons, urls and special punctuations like question marks and exclamation marks a microblog consists. These elements are normalized by the length of the microblog text.

Polarity Features: We leverage the given sentiment lexicons to increase the feature set and reflect the sentiment words of the microblog in numerical features. The feature vector consists of the following features for each sentiment lexicon: number of positive and negative sentiments words, sentiment score (number of positive words minus number of negative words), number of positive and negative

emoticons, number of positive and negative sentiments words around the topic words (context 5 words).

3.4 χ^2 Feature Selection

The idea of χ^2 feature selection is similar as mutual information. For each feature and class, there is also a score to measure if the feature and the class are independent to each other. We can use χ^2 test, which is a statistic method to check if two events are independent. It assumes the feature and class are independent and calculates χ^2 value. The large score implies they are not independent. The larger the score is, the higher dependency they have. So we want keep features for each classes with highest χ^2 scores. We use the Scikit library to select features according to the k highest scores.

3.5 Classification

After pre-processing and feature extraction we feed the features into a classifier. We tried various classifiers using the Scikit library, including Linear Support Vector Classification, Logistic Regression and Random Forest.

Linear Support Vector Classification (Linear SVC) similar to SVM with parameter kernel='linear', but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples.

Logistic Regression is a linear model for classification rather than regression. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function.

Random Forest fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting.

These implementations fit a multiclass (one-vs-rest) classification with L2 regularization. After experimentation it was found that Linear SVC gave the best performance. The parameters of the model were computed using grid search. The parameter search uses a 5-fold cross validation to find the maximum F-measure of different parameter values.

We implement a simple ensemble classifier that allows us to combine the different classifiers. It simply takes the majority rule of the predictions by the classifiers. The final classifier is the ensemble of linear SVC, logistic regression, random forest.

4 Experiments and Results

4.1 SIGHAN Dataset

Microblogs labeled as positive, negative or neutral were given by SIGHAN. The organizers provided us with 4,905 microblogs which contain 5 topics for training and 19,469 microblogs for the test data which contain 20 topics.

4.2 Results

We present the score and rank obtained by the system on the test dataset. There were 13 teams participated the task 2 of SIGHAN8. We compare our results with other participators using the F measure and the result is given in Table 2. The AVG and MAX represent the average and max value of the unrestricted result for all the participators. The F1+ and F1- represent the F measure for the positive and negative class respectively.

Model	F1+	F1-	F1
Baseline	0.1451	0.3943	0.3587
POS + Polarity Features	0.1551	0.3607	0.6796
POS + Polarity Features + TFIDF Weighting	0.1625	0.3888	0.7483
MAX	0.6039	0.6938	0.8535
AVG	0.1915	0.3646	0.6978

Table 2: The comparison with other participators for the classification task.

After combining POS features and polarity features with the TFIDF weighting, the model add features about the words, and the experiment result is improved.

5 Conclusion and Future Works

We present results for sentiment analysis on microblog by building a supervised system which combines TFIDF weighting with linguistic features which contain topic based features. We report the overall F-measure for three-way classification tasks: positive, negative and neutral.

At present, this system still has a lot of space to promote. Later, we will consider the following work to enhance the experiment result: Using the word vectors or neural network model for sentiment analysis tasks. More in-depth study of topics related features. For example, consider the coreference resolution technology to deal with the complicated situation refers to introducing syntax analysis.

6 Acknowledgements

This work is supported by the National Basic Research 973 Program of China under Grant No. 2011CB302200-G, the National Natural Science Foundation of China under Grant No.61370074, 61402091.

References

Dalmia A, Gupta M, Varma V. 2015. *SemEval 2015: Twitter Sentiment Analysis The good, the bad and the neutral!* SemEval 2015

Jiang L, Yu M, Zhou M, et al. 2011. *Target-dependent twitter sentiment classification*, volume 1. Association for Computational Linguistics

Blei D M, Ng A Y, Jordan M I. 2003. *Latent dirichlet allocation* the Journal of machine Learning research

Lee D D, Seung H S 1999. *Learning the parts of objects by non-negative matrix factorization* Nature

Boutsidis C, Gallopoulos E. 2008. *SVD based initialization: A head start for nonnegative matrix factorization* Pattern Recognition

Yuan B, Liu Y, Li H, et al. 2013. *Sentiment Classification in Chinese Microblogs: Lexicon-based and Learning-based Approaches* International Proceedings of Economics Development and Research (IPEDR)

Dong L, Wei F, Tan C, et al. 2014. *Adaptive recursive neural network for target-dependent twitter sentiment classification* Association for Computational Linguistics

Pang B, Lee L, Vaithyanathan S. 2002. *Thumbs up?: sentiment classification using machine learning techniques* Association for Computational Linguistics

Wang M, Liu M, Feng S, et al. 2014. *A Novel Calibrated Label Ranking Based Method for Multiple Emotions Detection in Chinese Microblogs* Natural Language Processing and Chinese Computing

Illecker M, Zangerle E. 2015. *Real-time Twitter Sentiment Classification based on Apache Storm*

Go A, Huang L, Bhayani R. 2009. *Twitter sentiment analysis* Entropy

Wasi S B, Neyaz R, Bouamor H, et al. 2014. *CMUQ@ Qatar: Using Rich Lexical Features for Sentiment Analysis on Twitter* SemEval 2014