UT-DB: An Experimental Study on Sentiment Analysis in Twitter

Zhemin Zhu   Djoerd Hiemstra   Peter Apers   Andreas Wombacher
CTIT Database Group, University of Twente
Drienerlolaan 5, 7500 AE, Enschede, The Netherlands
{z.zhu, d.hiemstra, p.m.g.apers, A.Wombacher}@utwente.nl

Abstract

This paper describes our system for participating SemEval2013 Task2-B (Kozareva et al., 2013): Sentiment Analysis in Twitter. Given a message, our system classifies whether the message is *positive*, *negative* or *neutral* sentiment. It uses a co-occurrence rate model. The training data are constrained to the data provided by the task organizers (No other tweet data are used). We consider 9 types of features and use a subset of them in our submitted system. To see the contribution of each type of features, we do experimental study on features by leaving one type of features out each time. Results suggest that unigrams are the most important features, bigrams and POS tags seem not helpful, and stopwords should be retained to achieve the best results. The overall results of our system are promising regarding the constrained features and data we use.

1 Introduction

The past years have witnessed the emergence and popularity of short messages such as tweets and SMS messages. Comparing with the traditional genres such as newswire data, tweets are very short and use informal grammar and expressions. The shortness and informality make them a new genre and bring new challenges to sentiment analysis (Pang et al., 2002) as well as other NLP applications such named entity recognition (Habib et al., 2013).

Recently a wide range of *methods* and *features* have been applied to sentimental analysis over tweets. Go et al. (2009) train sentiment classifiers using machine learning methods, such as Naive Bayes, Maximum Entropy and SVMs, with different combinations of features such as unigrams, bigrams and Part-of-Speech (POS) tags. Microblogging features such as hashtags, emoticons, abbreviations, all-caps and character repetitions are also found helpful (Kouloumpis et al., 2011). Saif et al. (2012) train Naive Bayes models with semantic features. Also the lexicon prior polarities have been proved very useful (Agarwal et al., 2011). Davidov et al. (2010) utilize hashtags and smileys to build a large-scale annotated tweet dataset automatically. This avoids the need for labour intensive manual annotation. Due to the fact that tweets are generated constantly, sentiment analysis over tweets has some interesting applications, such as predicting stock market movement (Bollen et al., 2011) and predicting election results (Tumasjan et al., 2010; O'Connor et al., 2010).

But there are still some unclear parts in the literature. For example, it is unclear whether using POS tags improves the sentiment analysis performance or not. Conflicting results are reported (Pak and Paroubek, 2010; Go et al., 2009). It is also a little surprising that *not* removing stopwords increases performance (Saif et al., 2012). In this paper, we build a system based on the concept of co-occurrence rate. 9 different types of features are considered. We find that using a subset of these features achieves the best results in our system, so we use this subset of features rather than all the 9 types of features in our submitted system. To see the contribution of each type of features, we perform experiments by leaving one type of features out each time. Results show that unigrams are the most important

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 384–389, Atlanta, Georgia, June 14-15, 2013. ©2013 Association for Computational Linguistics

features, bigrams and POS tags seem not helpful, and retaining stopwords makes the results better. The overall results of our system are also promising regarding the constrained features and data we use.

2 System Description

2.1 Method

We use a supervised method which is similar to the Naive Bayes classifier. The score of a tweet, denoted by t, and a sentiment category, denoted by c, is calculated according to the following formula:

$$Score(t, c) = [\sum_{i=1}^{n} \log CR(f_i, c)] + \log P(c),$$

where f_i is a feature extracted from t. The sentiment category c can be *positive*, *negative* or *neutral*. And $CR(f_i, c)$ is Co-occurrence Rate (CR) of f_i and c which can be obtained as follows:

$$CR(f, c) = \frac{P(f_i, c)}{P(f_i)P(c)} \propto \frac{\#(f_i, c)}{\#(f_i)\#(c)},$$

where $\#(*)$ is the number of times that the pattern $*$ appears in the training dataset. Then the category of the highest score $\arg\max_c Score(t, c)$ is the prediction.

This method assumes all the features are independent which is also the assumption of the Naive Bayes model. But our model excludes $P(f_i)$ because they are observations. Hence comparing with Naive Bayes, our model saves the effort to model feature distributions $P(f_i)$. Also this method can be trained efficiently because it only depends on the empirical distributions.

2.2 Features

To make our system general, we constrain to the text features. That is we do not use the features outside the tweet texts such as features related to the user profiles, discourse information or links. The following 9 types of features are considered:

1. Unigrams. We use lemmas as the form of unigrams. The lemmas are obtained by the Stanford CoreNLP[1] (Toutanova et al., 2003). Hashtags and emoticons are also considered as unigrams. Some of the unigrams are stopwords which will be discussed in the next section.

2. Bigrams. We consider two adjacent lemmas as bigrams.

3. Named entities. We use the CMU Twitter Tagger (Gimpel et al., 2011; Owoputi et al., 2013)[2] to recognize named entities. The tokens covered by a named entity are not considered as unigrams any more. Instead a named entity as a whole is treated as a single feature.

4. Dependency relations. Dependency relations are helpful to the sentiment prediction. Here we give an example to explain this type of features. In the tweet "I may not be able to vote from Britain but I COMPLETLEY support you!!!!", the dependency relation between the word 'not' and 'able' is 'NEG' which stands for negation, and the dependency relation between the word 'COMPLETELY' and 'support' is 'ADVMOD' which means adverb modifier. For this example, we add 'NEG able' and 'completely support' as dependency features to our system. We use Stanford CoreNLP (Klein and Manning, 2003a; Klein and Manning, 2003b) to obtain dependencies. And we only consider two types of dependencies 'NEG' and 'ADVMOD'. Other dependency relations are not helpful.

5. Lexicon prior polarity. The prior polarity of lexicons have been proved very useful to sentiment analysis. Many lexicon resources have been developed. But for a single lexicon resource, the coverage is limited. To achieve better coverage, we merge three lexicon resources. The first one is SentiStrength[3] (Kucuktunc et al., 2012). SentiStrength provides a fine-granularity system for grading lexicon polarity which ranges from -5 (most negative) to $+5$ (most positive). Our grading system consists of three categories: *negative*, *neutral* and *positive*. So we map the words ranging from -5 to -1 in SentiStrength to *negative* in our grading system, and the words ranging from

[1] http://nlp.stanford.edu/software/corenlp.shtml

[2] http://www.ark.cs.cmu.edu/TweetNLP/

[3] http://sentistrength.wlv.ac.uk/

$+1$ to $+5$ to *positive*. The rest are mapped to *neutral*. We do the same for the other two lexicon resources: OpinionFinder[4] (Wiebe et al., 2005) and SentiWordNet[5] (Esuli and Sebastiani, 2006; Baccianella and Sebastiani, 2010).

6. Intensifiers. The tweets containing intensifiers are more likely to be non-neutral. In the submitted system, we merge the boosters in SentiStrength and the intensifiers in OpinionFinder to form a list of intensifiers. Some of these intensifiers strengthen emotion (e.g. 'definitely'), but others weaken emotion (e.g. 'slightly'). They are distinguished and assigned with different labels {`intensifier_strengthen`, `intensifier_weaken`}.

7. All-caps and repeat characters. All-caps[6] and repeat characters are common expressions in tweets to make emphasis on the applied tokens. They can be considered as implicit intensifiers. In our system, we first normalize the repeat characters. For example, `happyyyy` is normalized to `happy` as there are ≥ 3 consequent `y`. Then they are treated in the same way as intensifier features discussed above.

8. Interrogative sentence. Interrogative sentences are more likely to be neutral. So we add if a tweet includes interrogative sentences as a feature to our system. The sentences ending with a question mark '?' are considered as interrogative sentences. We first use the Stanford CoreNLP to find the sentence boundaries in a tweet, then check the ending mark of each sentence.

9. Imperative sentence. Intuitively, imperative sentences are more likely to be negative. So if a tweet contains imperative sentences can be a feature. We consider the sentences start with a verb as imperative sentences. The verbs are identified by the CMU Twitter Tagger.

We further filter out the low-frequency features which have been observed less than 3 times in the training data. Because these features are not stable indicators of sentiment. Our experiments show that removing these low-frequency features increases the accuracy.

2.3 Pre-processing

The pre-processing of our system includes two steps. In the first step, we replace the abbreviations as described in Section 2.3.1. In the second step, we use the CMU Twitter Tagger to extract the features of emoticons (e.g. `:)`), hashtags (e.g. `#Friday`), reciepts (e.g. `@Peter`) and URLs, and remove these symbols from tweet texts for further processing.

2.3.1 Replacing Abbreviations

Abbreviations are replaced by their original expressions. We use the Internet Lingo Dictionary (Wasden, 2010) to obtain the original expressions of abbreviations. This dictionary originally contains 748 acronyms. But we do not use the acronyms in which all characters are digits. Because we find they are more likely to be numbers than acronyms. This results in 735 acronyms.

3 Experiments

Our system is implemented in Java and organized as a pipeline consisting of a sequence of annotators and extractors. This architecture is very similar to the framework of UIMA (Ferrucci and Lally, 2004). With such an architecture, we can easily vary the configurations of our system.

3.1 Datasets

We use the standard dataset provided by SemEval2013 Task2-B (Kozareva et al., 2013) for training and testing. The training and development data provided are merged together to train our model. Originally, the training and development data contain 9,684 and 1,654 instances, respectively. But due to the policy of Twitter, only the tweet IDs can be released publicly. So we need to fetch the actual tweets by their IDs. Some of the tweets are no longer existing after they were downloaded for annotation. So the number of tweets used for training is less than the original tweets provided by the organizers. In our case, we obtained 10,370 tweets for training our model.

[4] https://code.google.com/p/opinionfinder/

[5] http://sentiwordnet.isti.cnr.it/

[6] All characters of a token are in upper case.

Class	Precision	Recall	F-Score
Positive	74.86	60.05	66.64
Negative	47.80	59.73	53.11
Neutral	67.02	73.60	70.15
Avg (Pos & Neg)	61.33	59.89	**59.87**

Table 1: Submitted System on Twitter Data

Class	Precision	Recall	F-Score
Positive	54.81	57.93	56.32
Negative	37.87	67.77	48.59
Neutral	80.78	58.11	67.60
Avg (Pos & Neg)	46.34	62.85	**52.46**

Table 2: Submitted System on SMS Data

There are two test datasets: Twitter and SMS. The first dataset consists of 3,813 twitter messages and the second dataset contains 2,094 SMS messages. The purpose of having a separate test set of SMS messages is to see how well systems trained on twitter data will generalize to other types of data.

3.2 Results of Our Submitted System

We use a subset of features described in Section 2.2 in our submitted system: unigrams, named entities, dependency relations, lexicon prior polarity, intensifiers, all-caps and repeat characters, interrogative and imperative sentences. The official results on the two datasets are given in Table (1, 2). Our system is ranked as #14/51 on the Twitter dataset and #18/44 on the SMS dataset.

3.3 Feature Contribution Analysis

To see the contribution of each type of features, we vary the configuration of our system by leaving one type of features out each time. The results are listed in Table 3.

In Table 3, ‘Y(T)’ means the corresponding feature is used and the test dataset is the Twitter Data, and ‘N(sms)’ means the corresponding feature is left out and the test dataset is SMS Data.

Feature	Y(T)	N(T)	Y(sms)	N(sms)
Stopword	59.87	58.19	52.64	51.00
POS Tag	58.68	59.87	51.87	52.64
Bigram	58.47	59.87	51.94	52.64
Unigram	59.87	41.22	52.64	35.09
$3 \leq$	59.87	57.66	52.64	51.20
Intensifier	59.87	59.47	52.64	52.39
Lexicon	59.87	58.33	52.64	51.26
Named Ent.	59.87	59.71	52.64	51.80
Interrogative	59.87	59.67	52.64	52.93
Imperative	59.87	59.54	52.64	52.14
Dependence	59.87	59.37	52.64	52.08

Table 3: Avg (Pos & Neg) of Leave-one-out Experiments

From Table 3, we can see that unigrams are the most important features. Leaving out unigrams leads to a radical decrease of F-scores. On the Twitter dataset, the F-score drops from 59.87 to 41.44, and on the SMS dataset, the F-score drops from 52.64 to 35.09. And also filtering out the low-frequency features which happens less than 3 times increases the F-scores on Twitter data from 57.66 to 59.87, and on SMS data from 51.20 to 52.64. Removing stopwords decreases the scores by 1.66 percent. This result is consistent with that reported by Saif et al. (2012). By taking a close look at the stopwords we use, we find that some of the stopwords are highly related to the sentiment polarity, such as ‘can’, ‘no’, ‘very’ and ‘want’, but others are not, such as ‘the’, ‘him’ and ‘on’. Removing the stopwords which are related to the sentiment is obviously harmful. This means the stopwords which originally developed for the purpose of information retrieval are not suitable for sentimental analysis. Dependency relations are also helpful features which increase F-scores by about 0.5 percent. The POS tags and bigrams seem not helpful in our experiments, which is consistent with the results reported by (Kouloumpis et al., 2011).

4 Conclusions

We described the method and features used in our system. We also did analysis on feautre contribution. Experiment results suggest that unigrams are the most important features, POS tags and bigrams seem not helpful, filtering out the low-frequency features is helpful and retaining stopwords makes the results better.

Acknowledgements

This work has been supported by the Dutch national program COMMIT.

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andrea Esuli Stefano Baccianella and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May. European Language Resources Association (ELRA).

J. Bollen, H. Mao, and X. Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science*.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 241–249, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06*, pages 417–422.

David Ferrucci and Adam Lally. 2004. Uima: an architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng.*, 10(3-4):327–348, September.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for twitter: annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 42–47, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. Technical report, Stanford University.

M. B. Habib, M. van Keulen, and Z. Zhu. 2013. Concept extraction challenge: University of twente at #msm2013. In *Proceedings of the 3rd workshop on 'Making Sense of Microposts' (#MSM2013), Rio de Janeiro, Brazil*, Brazil, May. CEUR.

Dan Klein and Christopher D. Manning. 2003a. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2003b. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press.

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! In Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts, editors, *ICWSM*. The AAAI Press.

Zornitsa Kozareva, Preslav Nakov, Alan Ritter, Sara Rosenthal, Veselin Stoyonov, and Theresa Wilson. 2013. Sentiment analysis in twitter. In *Proceedings of the 7th International Workshop on Semantic Evaluation*. Association for Computation Linguistics.

Onur Kucuktunc, B. Barla Cambazoglu, Ingmar Weber, and Hakan Ferhatosmanoglu. 2012. A large-scale sentiment analysis for yahoo! answers. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 633–642, New York, NY, USA. ACM.

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*.

Olutobi Owoputi, Brendan OConnor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Hello, who is calling?: Can words reveal the social nature of conversations? In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, June.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 79–

86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hassan Saif, Yulan He, and Harith Alani. 2012. Semantic sentiment analysis of twitter. In *Proceedings of the 11th international conference on The Semantic Web - Volume Part I*, ISWC'12, pages 508–524, Berlin, Heidelberg. Springer-Verlag.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

A. Tumasjan, T.O. Sprenger, P.G. Sandner, and I.M. Welpe. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 178–185.

Lawrence Wasden. 2010. Internet lingo dictionary: A parents guide to codes used in chat rooms, instant messaging, text messaging, and blogs. Technical report, Attorney General.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 1(2):0.