DIEGOLab: An Approach for Message-level Sentiment Classification in Twitter

Abeed Sarker, Azadeh Nikfarjam, Davy Weissenbacher, Graciela Gonzalez
Department of Biomedical Informatics
Arizona State University
Scottsdale, AZ 85281, USA
{abeed.sarker,anikfarj,dweissen,graciela.gonzalez}@asu.edu

Abstract

We present our supervised sentiment classification system which competed in SemEval-2015 Task 10B: Sentiment Classification in Twitter— Message Polarity Classification. Our system employs a Support Vector Machine classifier trained using a number of features including n-grams, dependency parses, synset expansions, word prior polarities, and embedding clusters. Using weighted Support Vector Machines, to address the issue of class imbalance, our system obtains positive class F-scores of 0.701 and 0.656, and negative class F-scores of 0.515 and 0.478 over the training and test sets, respectively.

1 Introduction

Social media has seen unprecedented growth in recent years. Twitter, for example, has over 645,750,000 users and grows by an estimated 135,000 users every day, generating 9,100 tweets per second[1]). Users often express their views and emotions regarding a range of topics on social media platforms. As such, social media has become a crucial resource for obtaining information directly from end-users, and data from social media has been utilized for a variety of tasks ranging from personalized marketing to public health monitoring. While the benefits of using a resource such as Twitter include large volumes of data and direct access to end-user sentiments, there are several obstacles associated with the use of social media data. These include the use of non-standard terminologies, misspellings, short and ambiguous posts, and data imbalance, to name a few.

In this paper, we present a supervised learning approach, using Support Vector Machines (SVMs) for the task of automatic sentiment classification of Twitter posts. Our system participated in the SemEval-2015 task *Sentiment Classification in Twitter— Message Polarity Classification*. The goal of the task was to automatically classify the polarity of a Twitter post into one of three predefined categories— positive, negative and neutral. In our approach, we apply a small set of carefully extracted lexical, semantic, and distributional features. The features are used to train a SVM learner, and the issue of data imbalance is addressed by using distinct weights for each of the three classes. The results of our system are promising, with positive class F-scores of 0.701 and 0.656, and negative class F-scores of 0.515 and 0.478 over the training and test sets, respectively.

2 Related Work

Following the pioneering work on sentiment analysis by Pang *et. al.* (2002), similar research has been carried out under various umbrella terms such as: semantic orientation (Turney, 2002), opinion mining (Pang and Lee, 2008), polarity classification (Sarker et al., 2013), and many more. Pang *et al.* (2002) utilized machine learning models to predict sentiments in text, and their approach showed that SVM classifiers trained using bag-of-words features produced promising results. Similar approaches have been applied to texts of various granularities— documents,

[1] http://www.statisticbrain.com/twitter-statistics/. Accessed on: 26th August, 2014.

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 510–514,
Denver, Colorado, June 4-5, 2015. ©2015 Association for Computational Linguistics

sentences, and phrases.

Due to the availability of vast amounts of data, there has been growing interest in utilizing social media mining for obtaining information directly from users (Liu and Zhang, 2012). However, social media sources, such as Twitter posts, present various natural language processing (NLP) and machine learning challenges. The NLP challenges arise from factors, such as, the use of informal language, frequent misspellings, creative phrases and words, abbreviations, short text lengths and others. From the perspective of machine learning, some of the key challenges include data imbalance, noise, and feature sparseness. In recent research, these challenges have received significant attention (Jansen et al., 2009; Barbosa and Feng, 2010; Davidov et al., 2010; Kouloumpis et al., 2011; Sarker and Gonzalez, 2014).

3 Methods

3.1 Data

Our training and test data consists of the data made available for SemEval 2015 task 10 (A–D). Each instance of the data set made available consisted of a tweet ID, a user ID, and a sentiment category for the tweet. For training, we downloaded all the annotated tweets that were publicly available at the time of development of the system. We were able to obtain, from the training and development sets released by the organizers, a total of 9,289 tweets for which the annotations were available. Of these, 4,445 (48%) were annotated as neutral, 1,416 (15%) as negative, and 3,428 (37%) as positive. The data is heavily imbalanced with particularly small number of negative instances.

3.2 Features

We derive a set of lexical, semantic, and distributional features from the training data. A brief description of each feature and preprocessing technique is described below.

3.2.1 Preprocessing

We perform standard preprocessing such as tokenization, lowercasing and stemming of all the terms using the Porter stemmer[2] (Porter, 1980). Our preliminary investigations suggested that stop words can play a positive effect on classifier performances by their presence in word 2-grams and 3-grams; so, we do not remove stop words from the texts.

3.2.2 N-grams

Our first feature set consists of word n-grams of the tweets. A word n-gram is a sequence of contiguous *n* words in a text segment, and this feature enables us to represent a document using the union of its terms. We use 1-, 2-, and 3-grams as features.

3.2.3 Synset

It has been shown in past research that certain terms, because of their prior polarities, play important roles in determining the polarities of sentences (Sarker et al., 2013). Certain adjectives, and sometimes nouns and verbs, or their synonyms, are almost invariably associated with positive or non-positive polarities. For each adjective, noun or verb in a tweet, we use WordNet[3] to identify the synonyms of that term and add the synonymous terms as features.

3.2.4 Average Sentiment Score

For this feature, we incorporate a score that attempts to represent the general sentiment of a tweet using the prior polarities of its terms. Each word-POS pair in a comment is assigned a score and the overall score assigned to the comment is equal to the sum of all the individual term-POS sentiment scores divided by the length of the sentence in words. For term-POS pairs with multiple senses, the score for the most common sense is chosen. To obtain a score for each term, we use the lexicon proposed by Guerini *et al.* . The lexicon contains approximately 155,000 English words associated with a sentiment score between -1 and 1. The overall score a sentence receives is therefore a floating point number with the range [-1:1]. One problem faced, when using such a lexicon on tweets, is words are frequently misspelled and, thus, missed by the lexicon matching process. We, therefore, used a fast, moderately accurate, and publicly available spelling correction sys-

[2] We use the implementation provided by the NLTK toolkit `http://www.nltk.org/`.

[3] `http://wordnet.princeton.edu/`. Accessed on October 13, 2014.

tem[4] to process each tweet before performing lexicon matches.

3.2.5 Grammatical Dependencies

Stanford grammatical dependencies have been designed with a view to provide a simple and usable analysis of the grammatical structure of a sentence by people who are not (computational) linguists (de Marneffe et al., 2006). In this schema, each relation between words of a sentence are encoded as binary predicates between two words. A semantic interpretation which uses the notions of traditional grammar are attached to the relations to facilitate their comprehension. For example, from the sentence *I love the banner*, we expect in the analysis the relations *nsubj(love, I), det(banner, the), dobj(love, banner)* denoting subject, determinant and direct object roles, respectively. Based on previous research (Nikfarjam et al., 2012), our intuition is that dependency relationships maybe useful for polarity classification. We used the Stanford parser integrated in the Stanford CoreNLP 3.4 suite,[5] and computed collapsed and propagated dependency trees for each tweet.

3.2.6 Embedding Cluster Features

Considering the nature of the user posts in Twitter, it is common to observe rarely occurring or unseen tokens in the test data. In order to address this issue, we use embedding cluster features introduced in (Nikfarjam et al., 2014). We categorize the similar tokens into clusters, and as a result, each token in the corpus has an associated cluster number. Therefore, every tweet is represented with a set of cluster numbers, with similar tokens having the same cluster number. The word clusters are generated based on K-means clustering of the token representative vectors (known as embeddings). The embeddings are meaningful real-valued vectors of configurable dimensions (usually, 150 to 500 dimensions) learned from large volumes of unlabeled sentences. We generate 150-dimensional vectors using the word2vec tool.[6]. Our corpus includes a large number of unlabeled sentences from the provided train/test tweets plus an additional 860,000 in-house set of collected tweets about user opinions on medications. The vector and cluster dimensions are selected based on extrinsic evaluation of different configurations for the embedding clusters, generated from the same in-house Twitter corpus in our previous study. Word2vec learns the embeddings by training a neural network-based language model, and mapping tokens from similar contexts into vectors that can then be clustered using vector similarity techniques. More information about generating the embeddings can be found in the related papers (Bengio et al., 2003; Turian et al., 2010; Mikolov et al., 2013).

3.2.7 Other Features

In addition to the abovementioned features, we used the post lengths, in number of characters, as a feature.

3.3 Classification

Using the abovementioned features, we trained SVM classifiers for the classification task. The performance of SVMs can vary significantly based on the kernel and specific parameter values. For our work, based on some preliminary experimentation on the training set, we used the RBF kernel. We computed optimal values for the *cost* and γ parameters via grid-search and 10-fold cross validation over the training set. To address the problem of data imbalance, we utilized the weighted SVM feature of the LibSVM library (Chang and Lin, 2011), and we attempted to find optimal values for the weights in the same way using 10-fold cross validation over the training set. We found that $cost = 8.0$, $\gamma = 0.0$, $\omega_1 = 3.5$, and $\omega_2 = 2.2$ to produce the best results, where ω_1 and ω_2 are the weights for the positive and negative classes, respectively.

4 Results

Table 1 presents the performance of our system on the training and test data sets. The table presents the positive and negative class F-scores for the system, and the average of the two scores— the metric that is used for ranking systems in the SemEval evaluations for this task. For the training set, the results are those obtained via 10-fold cross validation. The test set

[4] `http://norvig.com/spell-correct.html`. Accessed on January 7, 2015.

[5] `http://nlp.stanford.edu/software/corenlp.shtml`. Accessed on January 8, 2015

[6] Available at: `https://code.google.com/p/word2vec/`. Accessed on 13 January, 2015

consists of 2,390 instances and the full training set is used when performing classification on this set.

Data set	Positive F-score (P)	Negative F-score (N)	$\frac{P+N}{2}$
Training	0.701	0.515	0.608
Test	0.656	0.478	0.567

Table 1: Classification results for the DIEGOLab system over the training and test sets.

4.1 Feature Analysis

To assess the contribution of each feature towards the final score, we performed leave-one-out feature and single feature experiments. Tables 3 and 2 show the $\frac{P+N}{2}$ values for the training and the test sets for the two set of experiments. The first row of the tables present the results when all the features are used, and the following rows show the results when a specific feature is removed or when a single feature is used. The tables illustrate that the most important feature set is n-grams, and there is a large drop in the evaluation score when that feature is removed (in Table 2). For all the other feature sets, the drops in the evaluation scores shown in Table 3 are very low, meaning that their contribution to the final evaluation score is quite limited. Table 3 suggests that the sentiment score feature is the second most useful feature after n-grams. The experiments suggest that the classifier settings (*i.e.*, the parameter values and the class weights) play a more important role in our final approach, as greater deviations from the scores presented can be achieved by fine tuning the parameter values than by adding, removing, or modifying the feature sets. Further experimentation is required to identify useful features and to configure existing features to be more effective.

Feature removed	Training set average	Test set average
None	0.608	0.567
N-grams	0.575	0.527
Synset	0.606	0.565
Sentiment Score	0.608	0.561
Grammatical Dependencies	0.601	0.562
Embedding Clusters	0.602	0.566
Other	0.608	0.565

Table 2: Leave-one-out $\frac{P+N}{2}$ feature scores for the training and test sets.

Feature	Training set average	Test set average
All	0.608	0.567
N-grams	0.587	0.560
Synset	0.507	0.478
Sentiment Score	0.561	0.489
Grammatical Dependencies	0.435	0.436
Embedding Clusters	0.482	0.461
Other	0.303	0.272

Table 3: Single feature $\frac{P+N}{2}$ scores for the training and test sets.

5 Conclusions and Future Work

Our system achieved moderate performance on the SemEval sentiment analysis task utilizing very basic settings. The F-scores were particularly low for the negative class, which can be attributed to the class imbalance. Considering that the performance of our system was achieved by very basic settings, there is promise of better performance via the utilization of various feature generation and engineering techniques.

We have several planned future tasks to improve the classification performance on this data set, and for social media based sentiment analysis in general. Following on from our past work on social media data (Patki et al., 2014; Sarker and Gonzalez, 2014), a significant portion of our future work will focus on the application of more informative features for automatic classification of social media text, including sentiment analysis. We are also keen to explore the use of text normalization techniques, at various granularities, to improve classification performance over social media data.

Acknowledgments

This work was supported by NIH National Library of Medicine under grant number NIH NLM 1R01LM011176. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NLM or NIH.

References

Luciano Barbosa and Junlan Feng. 2010. Robust Sentiment Detection on Twitter from Biased and Noisy Data. In *Proceedings of COLING*, pages 36–44.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Enhanced Sentiment Learning Using Twitter Hashtags and Smileys. In *Proceedings of COLING*, pages 241–249.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D Manning. 2006. Generating Typed Dependency Parsers from Phrase Structure Parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, pages 449–454.

Marco Guerini, Lorenzo Gatti, and Marco Turchi. 2013. Sentiment Analysis: How to Derive Prior Polarities from SentiWordNet. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 1259–1269.

Bernard J Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. 2009. Twitter Power: Tweets as Electronic Word of Mouth. *Journal of the American Society for Information Science and Technology*, 60(11):2169–2188.

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter Sentiment Analysis: The Good the Bad and the OMG! In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*.

Bing Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 415–463.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781v3.

Azadeh Nikfarjam, Ehsan Emadzadeh, and Graciela Gonzalez. 2012. A Hybrid System for Emotion Extraction from Suicide Notes. *Biomedical Informatics Insights*, 5. Suppl 1:165–174.

Azadeh Nikfarjam, Abeed Sarker, Karen O'Connor, Rachel Ginn, and Graciela Gonzalez. 2014. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association (JAMIA)*.

Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1):1–135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the ACL conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86.

Apurv Patki, Abeed Sarker, Pranoti Pimpalkhute, Azadeh Nikfarjam, Rachel Ginn, Karen O'Connor, Karen Smith, and Graciela Gonzalez. 2014. Mining Adverse Drug Reaction Signals from Social Media: Going Beyond Extraction. In *Proceedings of BioLinkSig 2014*.

Martin F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Abeed Sarker and Graciela Gonzalez. 2014. Portable Automatic Text Classification for Adverse Drug Reaction Detection via Multi-corpus Training. *Journal of Biomedical Informatics*.

Abeed Sarker, Diego Molla, and Cecile Paris. 2013. Automatic Prediction of Evidence-based Recommendations via Sentence-level Polarity Classification. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, pages 712–718.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394.

Peter Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL-02, 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424.