SemEval classification: Twitter Sentiment Analysis using different classifiers

Roel van der Burg, 4066243

November 2017

Abstract

Micro-blogging platforms like Twitter, established in 2006, have shown enormous growth in popularity.This proves a valuable opportunity for sentiment analysis. Natural language processing (NLP) studies have already extensively performed such analysis on longer pieces of text like documents and movie reviews. Short sentence classification, like tweets, however provide different NLP challenges. Sentiment analysis on sentences prove to be particularly sensitive to contextual polarity keywords and informal language use. Support vector machines (SVM) and nave Bayes (NB) has proven to be strong baselines for these types of tasks formerly, but more recently deep neural network models has shown strong improvements in performance on these tasks. This paper aims to build a deep neural network for 3-scale sentiment classification as described in SemEval task 4 [1].

1 Introduction

Extensive natural language processing (NLP) studies already performed analysis on longer pieces of text like documents and reviews. [2] [3] [4] [5]. Twitter however with its 140 character count limit (280 recently), imposes several different challenges. Twitter being a very informal blogging service, thus has very specialised terms and slang in its messages [2]. This paper aims to build a deep neural model to classify tweets to a three point sentiment score.

2 Literature Sentiment Analysis

As discussed, tweets differ from other NLP tasks in the sense that they are even more prone to very informal language, spelling, punctuation, misspellings and so forth. While this is fundamental to NLP, we still can expect wine reviews or legal writing to have a more structured text format. This poses an additional difficulty in the processing of tweets. A thesaurus of positive and negative keywords is often times used as a start for sentiment classification. This negative or positive sentiment connotation can be seen as an a priori polarity score thus belonging to the word without

context [6]. In practise however most words differ strongly in their a-priori and contextual polarity. Wilson et al report a simple prior polarity classifier classifying sentences to a neutral, positive, negative or both class having an accuracy of 48 percent [6].

Though an analysis of sentiment based on a single words proves another difficulty in the polarity reversal of a sentences. The impact of sentimental keywords can be modified, weakened, strengthened or reversed given certain contextual operators [7]. Nakagawa illustrates this with the example "the medicine kills cancer cells", while cancer cells generally infers a negative sentiment, the polarity is inversed by the addition of the word kills [8].

2.1 Existing Approaches

The top performing system from the SemEval 2013 classification competition used diverse sentiment lexicons together with a variety of manually extracted features to build a SVM classifier [9]. Additionally the Naive Bayes (NB) and Support Vector Machines (SVM) often times seem to provide strong baseline models for text sentiment classification. The performance of these algorithms naturally, is strongly dependent on the choice of features and data sets used. Wang reports that for short sentence tasks NB tend to outperform SVMs [10]. Moreover bag of features models in general tend to outperform more sophisticated structure-sensitive models [10]. However these manually crafted features recently have been outperformed by new approaches using (deep) neural networks. These approaches to sentiment classification use machine learning algorithms to focus on the design of more effective features [9]. Current state-of-the-art of the Sem-Eval-2017 task uses a Long Short-Term Memory or LSTM network together with word embedding pre-trained on large twitter dataset [11]. Other approaches consist of ensemble based gradient tree boost [12] or convolutional networks [13].

3 Methods and Data

This paper focuses on task 4, subtask A of the message sentiment polarity classification task of the SemEval-2017 International Workshop on Semantic Evaluation [1]. Task 4 focuses on Sentiment analysis on Twitter on a three-point (positive, negative, neutral) scale. For this year's task all previous data sets have been made available, totalling up to 51621 annotated three-point tweets, after removing duplicates. Existence of duplicate training data proves very detrimental to fitting the model. Moreover a public dataset containing 160 thousand tweets is used [14]. This dataset is annotated using distant supervision with emoticons serving as noisy labels for the sentiment of the tweet. After scraping, these emoticons where removed from the tweet. When used with proper pre-processing this large automatically acquired dataset is able to get up to eighty percent accuracy on a two-point scale [14].

Tweet	Sentiment
Who are you tomorrow? Will you make me smile or just bring me sorrow? #HottieOfTheWeek Demi Lovato	Neutral
The maestro ... the legend Roger Federer king of the backhand game one of his best shots	Positive

Table 1: Example tweets from the SemEval 2017 DataSet

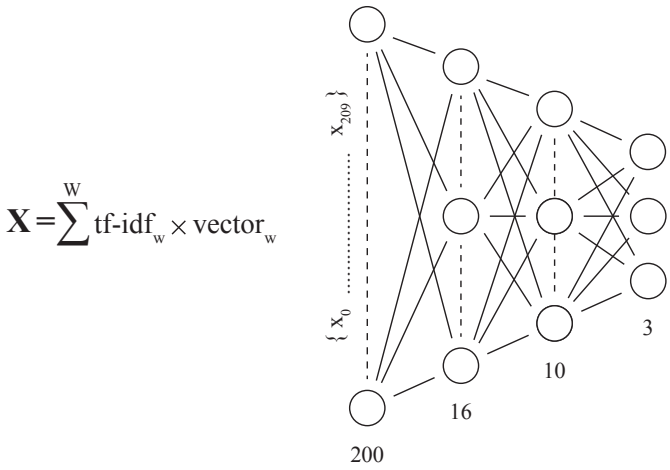

Figure 1: Schematic overview of Sequential Network

3.1 Word2Vec

To capture the dependencies of text in a word embedding a word2vec model is trained on the whole dataset [15]. The word2vec model is a fast and efficient way to train high dimensional vector representations of words and capture complex word dependencies that are able to deliver strong performance on several syntactic and semantic language tasks. For sentiment analysis on tweets however, the model trained requires a single vector representation for the whole tweet. For this a term frequency (tf), inverse document frequency (idf) weighting scheme is used. From here each term is smoothed using euclidean normalisation. The final weighting for each vector then is:

$$weight_{tf,idf} = \log \frac{1 + n_d}{1 + df(d,t)} * tf(t,d)$$

Weighting the final word embedding vector by a tf-idf gives several advantages over simply taking the average of all the words. Even after filtering the stop words in the pre-processing stage an argument is to be made that certain words do have greater explanatory power than other ones. The tf-idf score enables that by providing a dampening effect by the idf score that accounts for the frequency of a word in the corpus but increases proportionally to the number of occurrences of that particular word. This weighted feature vector calculated per tweet is then used as the input for the (deep) neural network shown in fig 1.

Table 2: Results A: Message Polarity Classification 3 point scale

Subtask A: Message Polarity Classification 3 point scale	AvgRec	$F1^{PN}$	Accuracy
SequentialModel Dev	0.674	0.771	0.742
SequentialModel Test Performance	0.583	0.544	0.584
SemEval-2016 Datastories Test performance	0.669	0.648	0.648
Baseline SemEval	0.245	0.33	0.483

Using a 10-fold stratified testing with the training set a two-layer neural network model proved to acquire better results over the single layer framework. The weighted word embedding vectors show to capture intricacy that is better captured by an extra layer in the network.

3.2 Evaluation

Testing will be done given a set of three evaluation measures. The first one will be the average recall, which is the recall averaged across the three classes.

$$AvgRec = \frac{1}{3}(R^P + R^N + R^U) \quad (1)$$

Here R^P, R^N, R^U refer to the recall of respective positive, negative and neutral class. Thus AvgRec will range between [0,1] where one represents the perfect classifier. The value 0.333 will represent a trivial classifier where all tweets are classified according to the same class. The advantage of an average recall over other measures as the accuracy is that it is more robust against class imbalances. The accuracy of a dataset with an imbalances majority set is skewed towards performance of that majority class. For the same reason the F_1 score is also sensitive to class imbalance. The second evaluation metric is the F_1^{PN} measure. [1].

$$F_1^{PN} = \frac{1}{2}(F_1^P + F_1^N) \quad (2)$$

Here the F_1^P, F_1^N refer to the F_1 scores with respect to the positive and negative classes. Together with the accuracy score these three measures are used for evaluation on the test set.

4 Results

Finally the data is test using the sequential neural network set up presented in the paper. Testing is been done on separate hold-out training set from the SemEval [1]. Results are presented in table 2, together with the top-performing classifier of the 2017 SemEval Task4 subtask A contestant. When ranked on average recall score current set up would rank 25th out of the 37 contestants.

5 Discussion and Conclusion

The model discussed in this paper showed average performance ranked across contenders and could be improved in several ways. Due to necessity of a word embedding as input for the network, compromises have to be made when weighting individual word embeddings into a single

[1] http://alt.qcri.org/semeval2017/task4/index.php?id=results

vector. Tf-idf is a strong method to amplify or dampen the effect of certain words, but do miss the intricacies of polarity reversal of words. Moreover the strength of the network is dependent on the information contained in vector X (Fig.1). The word2vec algorithm has to vectorize tweets and thus is presented with very informal and often times infrequent words in the dataset. To capture all (semantic) information into a high dimensional vector space, words ideally, need to be seen repeatedly by the algorithm. To mitigate these effects a bigger dataset could be used for training, or a minimal threshold can be set to eliminate certain words from the word2vec corpus.

References

[1] Sara Rosenthal, Noura Farra, and Preslav Nakov. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, 2017.

[2] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, 2010.

[3] Alistair Kennedy and Diana Inkpen. Sentiment classification of movie reviews using contextual valence shifters. *Computational intelligence*, 22(2):110–125, 2006.

[4] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[5] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM, 2003.

[6] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.

[7] Karo Moilanen and Stephen Pulman. Sentiment composition. In *Proceedings of RANLP*, volume 7, pages 378–382, 2007.

[8] Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. Dependency tree-based sentiment classification using crfs with hidden variables. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 786–794. Association for Computational Linguistics, 2010.

[9] Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*, 2013.

[10] Sida Wang and Christopher D Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics, 2012.

[11] Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, 2017.

[12] Naveen Kumar Laskari and Suresh Kumar Sanampudi. Twina at semeval-2017 task 4: Twitter sentiment analysis with ensemble gradient boost tree classifier. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 659–663, 2017.

[13] Yufei Xie, Maoquan Wang, Jing Ma, Jian Jiang, and Zhao Lu. Eica team at semeval-2017 task 3: Semantic and metadata-based features for community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 292–298, 2017.

[14] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(2009):12, 2009.

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.