From Unlabelled Tweets to Twitter-specific Opinion Words

Felipe Bravo-Marquez, Eibe Frank, and Bernhard Pfahringer
Department of Computer Science, University of Waikato
Hamilton, New Zealand
fjb11@students.waikato.ac.nz, eibe@waikato.ac.nz, bernhard@waikato.ac.nz

ABSTRACT

In this article, we propose a word-level classification model for automatically generating a Twitter-specific opinion lexicon from a corpus of unlabelled tweets. The tweets from the corpus are represented by two vectors: a bag-of-words vector and a semantic vector based on word-clusters. We propose a distributional representation for words by treating them as the centroids of the tweet vectors in which they appear. The lexicon generation is conducted by training a word-level classifier using these centroids to form the instance space and a seed lexicon to label the training instances. Experimental results show that the two types of tweet vectors complement each other in a statistically significant manner and that our generated lexicon produces significant improvements for tweet-level polarity classification.

Categories and Subject Descriptors

I.2.7.7 [**Artificial Intelligence**]: Natural Language Processing—*Text Analysis*

General Terms

Experimentation, Measurement

Keywords

Lexicon Generation; Sentiment Analysis; Twitter

1. INTRODUCTION

Twitter[1] is a massive microblogging service in which users post short messages limited to 140 characters referred to as tweets. The large amount of personal opinions that is constantly generated on this platform has drawn increasing attention among the sentiment analysis research community.

The main challenge in analysing Twitter opinions is how to deal with the informal dialect used on this plattform, because it contains expressions such as acronyms, abbreviations, slang words, and misspelled words, that are not observed in traditional media [5].

Opinion lexicons, which are resources that associate words with sentiment polarities, play a central role in sentiment analysis applications [11]. However, most existing opinion lexicons focus on formal English expressions, and are unsuitable for Twitter sentiment analysis.

In this article, we propose a method for automatically generating a Twitter-oriented opinion lexicon from a collection of unlabelled tweets. We classify each word from a corpus into one of three different polarity classes: positive, negative, or neutral. The words are represented by vectors of attributes that are based on the context in which the words occur. We use a seed lexicon to label a sample of the words and train a linear classifier on the labelled instances. The fitted model is then used to classify the remaining unlabelled words.

Our approach based on word-level vectors takes the Distributional Hypothesis [7] as inspiration, which states that words occurring in the same contexts tend to have similar meanings. We exploit the short nature of Twitter messages to treat a whole tweet as a word's context, and we model tweets as vectors calculated from the textual content. We calculate word-level vectors based on the centroids of the tweet vectors in which a word occurs. In essence, we are assuming that words exhibiting a certain polarity are more likely to be used in contexts expressing the same polarity than in contexts exhibiting a different one.

We study and compare two different vector space models for tweet-level representation. The first is a high-dimensional bag-of-words model using word frequencies as dimension values. The second, is a semantic representation based on word-clusters. We rely on the Brown clustering algorithm [2] to tag a tweet according to a sequence of word clusters and create cluster frequency vectors.

Previous approaches for Twitter-specific lexicon generation rely on collections of tweets that were previously labelled to sentiment classes using distant supervision [13, 20] or pre-trained classifiers [1]. In contrast, our approach takes a raw collection of tweets and a seed lexicon to perform the generation. To the best of our knowledge, this is the first lexicon generation model for tweets in which a word-level classifier is trained using features calculated from unlabelled corpora.

The remainder of this article is organised as follows. In Section 2, we review some previous work on opinion lexicon generation. In Section 3, we formalise our word-level vector space models. Our main experiments and results are presented in Section 4. The conclusions are discussed in Section 5.

2. RELATED WORK

Lexicon generation techniques normally rely on a small seed lexicon which is expanded by exploiting word relations from two type of resources: a lexical database such as WordNet, or a corpus of

[1] `http://www.twitter.com`

SIGIR'15, August 09 - 13, 2015, Santiago, Chile.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3621-5/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2766462.2767770 .

documents. Methods based on WordNet consider semantic relations such as synonyms, antonyms, [9, 10] or dictionary definitions [3, 4] to perform the expansion. As semantic databases cover a fixed vocabulary, they are not suitable for the Twitter dialect. On the other hand, corpus approaches exploit statistical patterns observed in document corpora. Thus, they can potentially be applied to any domain. Statistical patterns can be computed using different types of methods, such as conjunction relations between adjectives [8], latent semantic analysis [16], and pointwise mutual information (PMI) [16, 17]. Previous work on Twitter lexicon generation computes the PMI between words and tweet-level sentiment labels. The tweets are automatically labelled to polarity classes using either distant supervision [13, 20] or self-training [1]. Distant supervision methods rely on strong sentiment clues found in a message such as emoticons [13, 20] or hashtags [13] to label the messages. Tweets where these clues are not observed are discarded. In the self-training approach [1], a message-level polarity classifier is trained from a corpus of manually labelled tweets and used to tag a large corpus of unlabelled tweets.

3. TWEET-CENTROID WORD VECTORS

In this section, we describe the word vectors for lexicon generation. These vectors are distributional representations [18] in which words are described according to their context. We assume that a word's context is the entire tweet in which it occurs. The first model we discuss is the bag-of-words (BOW) tweet-centroid model, which represents words according to the other words that co-occur with it.

Suppose we have a corpus $\mathcal{C}$ formed by n tweets $t_1, \ldots, t_n$, where each tweet t is a sequence of words. Let $\mathcal{V}$ be the vocabulary formed by the m different words $w_1, \ldots, w_m$ found in $\mathcal{C}$. The tweet-level bag-of-words model represents each tweet t as a m-dimensional vector $\overrightarrow{tb}$ where each dimension j has a numerical value $f_j(t)$ that corresponds to the frequency of the word w_j within the sequence of words of t.

For each word w, we define the word-tweet set $\mathcal{W}(w)$ as the set of tweets in which w is observed:

$$\mathcal{W}(w) = \{t : w \in t\} \tag{1}$$

We define the bag-of-words vector $\overrightarrow{wb}$ as as the centroid of all tweet vectors in which w is used. In other words, $\overrightarrow{wb}$ is an m-dimensional vector in which each dimension wb_j is calculated as follows:

$$wb_j = \sum_{t \in \mathcal{W}(w)} \frac{f_j(t)}{|\mathcal{W}(w)|} \tag{2}$$

However, because bag-of-word models tend to produce high-dimensional sparse vectors, we also study another word vector representation with lower dimensionality based on the interaction of word clusters.

Let c be a clustering function that maps the m words from $\mathcal{V}$ to a partition $\mathcal{S}$ containing k classes, with $k \ll m$. In our experiments, this function is trained in an unsupervised fashion from a corpus of tweets using the Brown clustering algorithm [2], which produces hierarchical clusters by maximising the mutual information of bigrams. These clusters have shown to be useful for tagging tweets according to part-of-speech classes [5].

We tag the word sequences of the tweets from $\mathcal{C}$ with the clustering function c. Afterwards, we create a new tweet-level vector $\overrightarrow{tc}$ of k dimensions based on the frequency of occurrence of a cluster s in the tweet. The cluster-based word vectors $\overrightarrow{wc}$ are calculated analogously to the bag-of-words vectors in the first approach. We take the centroids of the cluster-based vectors $\overrightarrow{tc}$ from the tweets of $\mathcal{W}(w)$, producing k-dimensional vectors for each word.

4. EXPERIMENTS

We evaluate the proposed vectors for lexicon generation using two different collections of tweets: the Edinburgh corpus (ED) [15], and the Stanford Twitter Sentiment corpus (STS)[2] [6].

The ED corpus is a collection of 97 million tweets acquired from the Twitter Streaming API covering multiple topics and languages. We take a random sample of 2.5 million English tweets from this collection. The STS corpus is a collection of 1.6 million English tweets collected by submitting queries with positive and negative emoticons to the Twitter search API. The emoticons are removed from the content. The ED corpus represents a realistic sample from a stream of tweets, whereas STS was intentionally manipulated to over-represent subjective tweets. We study these datasets to observe the effects of manipulating the collection of tweets for lexicon generation.

We tokenise the tweets from both collections and create the vectors $\overrightarrow{wb}$ and $\overrightarrow{wc}$ described in Section 3. The clustering function c was taken from the **TweetNLP** project[3]. This function was trained to produce 1000 different word clusters from a collection of around 56 million tweets using the Brown-clustering algorithm.

The two vectors $\overrightarrow{wb}, \overrightarrow{wc}$ are used as attributes to train a word-level classifier for lexicon generation. To avoid learning spurious relationships from infrequent words, vectors of words that occur in less than 10 tweets are discarded ($|\mathcal{W}(w)| < 10$). We also discard the dimensions from $\overrightarrow{wb}$ corresponding to those unfrequent words. Analogously, we remove all dimensions from $\overrightarrow{wc}$ associated with clusters appearing in less than 10 tweets.

We label the words that match a seed lexicon formed by words categorised into three sentiment categories: positive, negative, and neutral. The seed lexicon is built from the union of four existing hand-made lexicons, and a list of 87 positive and negative emoticons: *MPQA* [19], *Bing Liu* [11], *Afinn* [14], and *NRC-emotion lexicon* [12]. We discard all words labelled with conflicting polarities by different lexicons. The resulting seed lexicon has 3769 positive, 6414 negative, and 7088 neutral words. The main properties of the ED and STS datasets are summarised in Table 1.

Dataset	STS	ED
#tweets	$1,600,000$	$2,500,000$
#positive words	$2,015$	$2,639$
#negative words	$2,621$	$3,642$
#neutral words	$3,935$	$5,085$
#unlabelled words	$36,451$	$67,692$
#bag-of-words attributes	$45,022$	$79,058$
#cluster-vector attributes	993	999

Table 1: Dataset properties.

We first study the problem of classifying words into positive and negative classes. We train an L2-regularised logistic regression model with the regularisation C parameter set to 1.0 using LibLINEAR[4]. For performance estimation, we apply 10 times 10-folds cross-validation on the positive and negative labelled words from the two datasets. We compare three different instance spaces: bag-of-words vectors $\overrightarrow{wb}$, cluster vectors $\overrightarrow{wc}$, and the concatenation of both: $[wb_1, \ldots, wb_m, wc_1, \ldots, wc_k]$. We compare classification

[2] `http://help.sentiment140.com/`

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

[4] `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`

accuracy and the weighted area under the ROC curve (AUC) obtained by the different instance spaces using a corrected resampled paired *t*-student test with an α level of 0.05. Results are displayed in Table 2. Statistically significant improvements over the bag-of-words approach are denoted with the symbol ○.

Accuracy			
Dataset	BOW	CLUSTER	CONCAT
STS	75.52 ± 1.81	77.2 ± 1.9 ○	**77.85** ± 1.94 ○
ED	77.75 ± 1.54	77.62 ± 1.37	**79.15** ± 1.39 ○
AUC			
Dataset	BOW	CLUSTER	CONCAT
STS	0.83 ± 0.02	0.84 ± 0.02 ○	**0.85** ± 0.02 ○
ED	0.85 ± 0.01	0.85 ± 0.01	**0.86** ± 0.01 ○

Table 2: Word-level 2-class polarity classification performance.

We can observe that the classification results are slightly better for ED than STS. The cluster-based representation is better than the bag-of-words representation in STS. However, this pattern is not observed in ED. The concatenation of both vector models produces significant improvements in accuracy and AUC over the baseline in both datasets.

Accuracy			
Dataset	BOW	CLUSTER	CONCAT
STS	61.84 ± 1.46	64.42 ± 1.54 ○	**64.57** ± 1.44 ○
ED	62.93 ± 1.31	64.5 ± 1.16 ○	**65.5** ± 1.19 ○
AUC			
Dataset	BOW	CLUSTER	CONCAT
STS	0.77 ± 0.01	**0.79** ± 0.01 ○	**0.79** ± 0.01 ○
ED	0.78 ± 0.01	0.79 ± 0.01 ○	**0.8** ± 0.01 ○

Table 3: Word-level three-class polarity classification performance.

The detection of neutral words is an important task in sentiment analysis because it enables removal of non-opinion words from a passage of text. In the next experiment, we include neutral words to train a three-class polarity classifier. The classification results are given in Table 3. We can see that the classification performance is lower than in the previous experiment. The cluster-based vectors are significantly better than the bag-of-words vectors in both datasets. This suggests that word clusters are especially helpful in distinguishing neutral and non-neutral words. The concatenation of the two vectors achieves the best performance among all the experiments.

word	label	negative	neutral	positive
#recession	negative	0.603	0.355	0.042
#silicon_valley	neutral	0.043	0.609	0.348
bestfriends	positive	0.225	0.298	0.477
christamas	positive	0.003	0.245	0.751
comercials	negative	0.678	0.317	0.005
hhahaha	positive	0.112	0.409	0.479
powerpoint	neutral	0.068	0.802	0.13
psychotic	negative	0.838	0.138	0.024
widows	negative	0.464	0.261	0.275
yassss	positive	0.396	0.08	0.524

Table 4: Generated words example.

We use the three-class classifiers trained using both vectors to label the unlabelled words from the two collection of tweets. A sample of the generated words from the ED corpus with the estimated probabilities for negative, neutral, and positive classes is shown in Table 4.

As an additional validation for the generated words, we study their usefulness for classifying the overall polarity of Twitter messages. To do this, we compare the classification performance obtained by a simple classifier that uses attributes calculated from the seed lexicon, with the performance obtained by a classifier with attributes derived from both the seed lexicon and the generated words. The evaluation is done on three collections of tweets that were manually annotated to positive and negative classes: *6HumanCoded*[5], *Sanders*[6], and *SemEval*[7]. The number of positive and negative tweets of these datasets is given in Table 5.

	Positive	Negative	Total
6Coded	1340	949	2289
Sanders	570	654	1224
SemEval	5232	2067	7299

Table 5: Message-level polarity classification datasets.

The baseline of this experiment is a logistic regression model trained using the number of positive and negative words from the seed lexicon that are found within the tweet's content as attributes. For each expanded lexicon, we train a logistic regression model using the baseline attributes together with a positive and a negative score calculated as the weighted sum of the corresponding probabilities of words classified as positive or negative, respectively.

Accuracy			
Dataset	Baseline	STS	ED
Sanders	73.25 ± 3.51	74.76 ± 4.21	**76.58** ± 3.8 ○
6-human	72.84 ± 2.57	75.08 ± 2.31 ○	**76.42** ± 2.34 ○
SemEval	77.72 ± 1.24	78.97 ± 1.31 ○	**79.18** ± 1.22 ○
AUC			
Dataset	Baseline	STS	ED
Sanders	0.78 ± 0.04	0.8 ± 0.04 ○	**0.83** ± 0.04 ○
6-human	0.79 ± 0.03	0.82 ± 0.03 ○	**0.83** ± 0.02 ○
SemEval	0.78 ± 0.02	0.82 ± 0.02 ○	**0.84** ± 0.02 ○

Table 6: Message-level classification performance.

The classification results obtained for message-level classification in the three datasets are shown in Table 6. We observe from the table that with the exception of the accuracy obtained by the STS-based lexicon on the Sanders dataset, the generated lexicons produce significant improvements over the baseline. Furthermore, the lexicon generated from the ED corpus outperforms the performance of the STS lexicon in accuracy and AUC score respectively. These results indicate that collections of tweets manipulated to over-represent subjective tweets such as STS, are not necessarily better for lexicon generation than random collections of tweets such as ED.

5. CONCLUSIONS

In this paper, we studied two distributional representations for classifying Twitter opinion words in a supervised fashion. Our experimental results show the usefulness of the generated words for message-level polarity classification. The main advantage of the

[5] http://sentistrength.wlv.ac.uk/documentation/6humanCodedDataSets.zip

[6] http://www.sananalytics.com/lab/twitter-sentiment/

[7] http://www.cs.york.ac.uk/semeval-2013/task2/

proposed technique is that it depends on resources that are relatively cheap to obtain: a seed lexicon, and a collection of unlabelled tweets. The former can be obtained from publicly available resources such as the ones used in this work, and the latter can be freely collected from the Twitter API. The source code and generated lexicons are released to the research community[8].

The proposed method does not depend on labelled tweets or tweets with emoticons, in contrast to previous approaches [1, 13, 20]. Thus, our model can be used to identify domain-specific opinion words by collecting tweets from the target domain. This could be useful in domains such as politics, in which emoticons are not frequently used to express negative and positive opinions.

Considering that our model represents words by the centroid of tweet-level vectors, we could extend it to include any kind of feature used for message-level sentiment classification. These features could include textual properties such as bigrams, part-of-speech tags, negations, among others. In future work, we will also study how to include attributes provided by low-dimensional distributed representations or word embeddings such as the neural language models implemented in the *Word2vec* library[9].

Finally, it would be possible to extend the model to produce a more fine-grained word-level categorisation based on emotion categories, e.g., anger, fear, surprise, and joy. This could be achieved by relying on the labels provided by an emotion-associated lexicon [12] and multi-label classification techniques.

6. REFERENCES

[1] L. Becker, G. Erhart, D. Skiba, and V. Matula. Avaya: Sentiment analysis on twitter with self-training and polarity lexicon expansion. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises*, SemEval'13, pages 333–340, 2013.

[2] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

[3] A. Esuli and F. Sebastiani. Determining the semantic orientation of terms through gloss classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 617–624, New York, NY, USA, 2005. ACM.

[4] A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation*, LREC'06, pages 417–422, 2006.

[5] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. 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*, pages 42–47. Association for Computational Linguistics, 2011.

[6] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 2009.

[7] Z. Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

[8] V. Hatzivassiloglou and K. R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98, pages 174–181, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.

[9] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM.

[10] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, pages 1367–1373, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[11] B. Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.

[12] S. Mohammad and P. D. Turney. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[13] S. M. Mohammad, S. Kiritchenko, and X. Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises*, SemEval'13, pages 321–327, 2013.

[14] F. Nielsen. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*, #MSM2011, pages 93–98, 2011.

[15] S. Petrović, M. Osborne, and V. Lavrenko. The edinburgh twitter corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, WSA '10, pages 25–26, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[16] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[17] P. D. Turney and M. L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346, 2003.

[18] P. D. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, Jan. 2010.

[19] T. Wilson, J. Wiebe, and P. 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*, HLT '05, pages 347–354, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[20] Z. Zhou, X. Zhang, and M. Sanderson. Sentiment analysis on twitter through topic-based lexicon expansion. In H. Wang and M. Sharaf, editors, *Databases Theory and Applications*, volume 8506 of *Lecture Notes in Computer Science*, pages 98–109. Springer International Publishing, 2014.

[8] http://www.cs.waikato.ac.nz/ml/sa/lex.html#sigir15

[9] https://code.google.com/p/word2vec/