From Unlabelled Tweets to Twitter-specific Opinion Words

Felipe Bravo-Marquez, Eibe Frank Bernhard Pfahringer

Department of Computer Science, University of Waikato

Problem

- The dialect used in **Twitter** contains expressions such as abbreviations, slang words, and misspelled words, that are not observed in traditional media, e.g., **omg**, **looove**, **#hatemyboss**.
- **Opinion lexicons** are resources that associate words with positive and negative sentiment polarities, e.g., **good, awesome** and **ugly, disappointing**, that play a **central** role in sentiment analysis applications.
- Most existing opinion lexicons focus on **formal** English expressions, and are **unsuitable** for Twitter sentiment analysis.
- Previous approaches for Twitter-specific lexicon generation rely on tweets labelled to sentiment classes using **emoticons** [1, 2].
- These approaches are not suitable for domains such as politics, in which emoticons are **infrequently** used to express negative and positive opinions.

Proposed solution

- We propose a supervised model 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 from the corpus are represented by **vectors** based on the **centroids** of the tweet vectors in which a word occurs.
- 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, **generating** new opinion words.
- The model assumes 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.

Tweet-Centroid Model

- Let be $\mathcal{C}$ a **corpus** of 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 an 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:

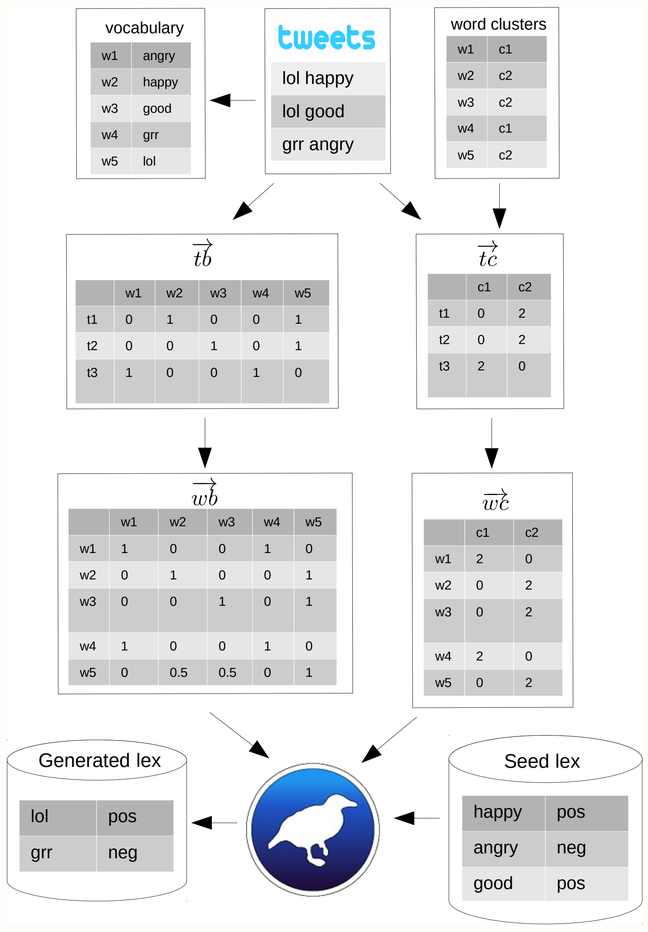
$$\mathcal{W}(w) = \{t : w \in t\} \quad (1)$$

- We define the bag-of-words vector $\overrightarrow{wb}$ as as the **centroid** of all tweet vectors in which w is used. Each dimension of the vector wb_j is calculated as follows:

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

- We also study another word vector representation with **lower dimensionality** based on the interaction of **word clusters**.
- Let c be a clustering function trained with the **Brown clustering algorithm** that maps the m words from $\mathcal{V}$ to a partition $\mathcal{S}$ containing k classes, with $k \ll m$.

- The cluster-based word vectors $\overrightarrow{wc}$ are calculated **analogously** to the bag-of-words vectors in the first approach, producing k-dimensional vectors for each word.



Intrinsic Evaluation

- We use two different collections of tweets: the **Edinburgh corpus** (ED) and the **Stanford Twitter Sentiment** corpus (STS).
- We **tokenise** the tweets from both collections and create the vectors $\overrightarrow{wb}$ and $\overrightarrow{wc}$.
- 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 use a **seed lexicon** of 3769 **positive**, 6414 **negative**, and 7088 **neutral** words to **label** the training words.

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

- All the classification experiments are conducted training an **L2-regularised logistic regression** model with the regularisation C parameter set to 1.0 using LibLINEAR.
- We compare three different instance spaces: **bag-of-words** vectors, **cluster vectors**, and the **concatenation** of both: $[wb_1, \ldots, wb_m, wc_1, \ldots, wc_k]$.
- We use 10 times 10-folds cross-validation and compare results using a corrected resampled paired t-student test with an α level of 0.05.

2-class 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 ○
2-class 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 ○
3-class 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 ○
3-class 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 ○

- 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

- A sample of the generated words with the estimated probabilities for **negative**, **neutral**, and **positive** classes.

Extrinsic Evaluation

- We also study the **usefulness** of the generated words for classifying the overall polarity of Twitter **messages**.
- The baseline of this experiment is a logistic regression model trained using the number of positive and negative words from the **seed lexicon**.
- For each expanded lexicon, we train a logistic regression model using the baseline attributes together with a positive and a negative attribute calculated from the **expanded words**.

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 ○

- The generated lexicons produce **significant improvements** over the baseline.

Conclusion

- We present a supervised model for Twitter opinion lexicon generation that depends on resources that are **cheap to obtain**: a seed lexicon, and a collection of unlabelled tweets
- The model can be used to identify **domain-specific** opinion words by collecting tweets from the target domain.
- The tweet-centroid word representations could be extended to include **any kind of feature** used for message-level sentiment classification, e.g., bigrams, part-of-speech tags, negations.
- The model could be extended to produce a more fine-grained word-level categorisation based on **emotion categories**, e.g., anger, fear, surprise, and joy by relying on an emotion-associated lexicon and **multi-label** classification techniques.

References

[1] 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.

[2] 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.