DL Team at SemEval-2018 Task 1: Tweet Affect Detection using Sentiment Lexicons and Embeddings

Dmitry Kravchenko
Ben-Gurion University of the Negev / Israel
to.dmitry.kravchenko@gmail.com

Lidia Pivovarova
University of Helsinki / Finland
lidia.pivovarova@cs.helsinki.fi

Abstract

The paper describes our approach for SemEval-2018 Task 1: Affect Detection in Tweets. We perform experiments with manually compelled sentiment lexicons and word embeddings. We test their performance on twitter affect detection task to determine which features produce the most informative representation of a sentence. We demonstrate that general-purpose word embeddings produces more informative sentence representation than lexicon features. However, combining lexicon features with embeddings yields higher performance than embeddings alone.

1 Introduction

The paper describes our approach for SemEval-2018 Task 1: Affect Detection in Tweets (Mohammad et al., 2018).

The research question we address in this paper is what are the best features for tweet affect detection. Our solution uses two types of features: *lexicon features* obtained from manually compiled emotion lexicons, and *word embeddings* built unsupervisedly from large corpora. We use well established lexicons, namely DepecheMood and Vader Sentiment, and most popular Word embeddings, namely GloVe and Google News. We systematically compare all features on two subtasks and demonstrate that even though lexicon features produce unsatisfactory results in isolation, they significantly improve an algorithm performance when combined with more general embeddings.

In addition, we demonstrate that special treatment of Twitter hash-tags also improves the algorithm performance.

2 Tasks and Data

The paper addresses three subtasks:

Task		**Train**	**Dev**	**Test**
EI-reg	all emotions	7102	1464	71816
	anger	1701	388	17939
	fear	2252	389	17923
	joy	1616	290	18042
	sadness	1533	397	17912
V-reg		1181	449	17874
E-c		6838	886	3259

Table 1: Training, development and test set split for three subtasks

- **EI-reg**—an emotion intensity regression task: Given a tweet and an emotion E, determine the intensity of E that best represents the mental state of the tweeter—a real-valued score between 0 (no E at all) and 1 (the highest magnitude of E); separate datasets are provided for fear, sadness, anger, and joy.
- **V-reg**—a sentiment intensity regression task: Given a tweet, determine the intensity of sentiment or valence (V) that best represents the mental state of the tweeter—a real-valued score between 0 (most negative) and 1 (most positive).
- **E-c**—an emotion classification task: Given a tweet, classify it as 'neutral or no emotion' or as one, or more, of eleven given emotions that best represent the mental state of the tweeter: trust, sadness, disgust, fear, optimism, love, joy, pessimism, anticipation, surprise, and anger.

We use English data for all three subtasks. The train, development and test set sizes are shown in Table 1. More details on the data can be found in the task organizers' paper (Mohammad and Kiritchenko, 2018).

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 172–176
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

3 Approach

3.1 Baseline

As a baseline we use the Text-Processing API[1]. The API uses a Naive Bayes model trained using movie reviews and NLTK. The model returns probabilities for negative, positive and neutral labels. Negative and positive probabilities sum to 1 while neutral probability stands alone.

3.2 Lexicon Features

3.2.1 DepecheMood

DepecheMood (Staiano and Guerini, 2014) is an emotion lexicon collected using crowdsourcing. The respondents annotated news articles with eight predefined emotions: afraid, amused, angry, annoyed, dont_care, happy, inspired, sad. Document annotations were then used in a dimensionality reduction algorithm to obtain word emotional scores. The lexicon contains approximately 37 thousand entry. Each entry consists of a word and eight values between 0 and 1, one value for each emotion.

3.2.2 Vader

Vader (Valence Aware Dictionary and sEntiment Reasoner) is a rule-based sentiment analysis tool and a lexicon specifically attuned to sentiments expressed in social media, such as Twitte (Hutto and Gilbert, 2014). The lexicon consists of more than 7000 term, which were compelled from other lexicons and then manualy annotated. Git repository[2] of Vader Sentiment toolkit provides function *polarity_scores* which takes as an input a text and returns 4-dimensional feature vector, which contains negative, positive, neutral and compound scores.

3.3 Embeddings

3.3.1 GloVe

GloVe (Pennington et al., 2014) is an unsupervised algorithm that constructs embeddings from large corpora. The GloVe project [3] provides a number of models trained on various collections. We use the following two models:

1. Common Crawl: 300-dimensional vectors trained on huge Internet corpus of 840 billion tokens and 2.2 million distinct words.
2. Twitter Crawl: 200-dimensional vectors trained on 2 billion tweets with 27 billion tokens and 1.2 million distinct words.

3.3.2 Google News

We use word2vecs (Mikolov et al., 2013) embedding trained on Google News collection[4], which have become almost standard embeddings since they are most frequently used in various research tasks. These embeddings are 300-dimensional vectors built using Google News dataset of 100 billion tokens and 3 million distinct words and phrases.

3.4 Method

We use various combinations of baseline, lexicon and embedding features, described above. Text-processing API and Vader return text-level features. For other sources a tweet representation is built by averaging the word vectors. Concatenation is used to combine features obtained from various sources.

We run several preliminary experiments with V-reg task to compare several algorithms, namely Gradient Boosting Regressor and Random Forest. We use sklearn implementations [5]. Gradient Boosting Regressor yields the best performance for all feature combinations (Table 2). In our official submission we apply Gradient Boosting Regressor for tasks EI-reg and V-reg, and Gradient Boosting Classifier for task E-c.

Hash-tags are special types of tokens in Twitter used to specify a topic or a context for a given message. They frequently contain emotional words. Here are several examples from the dataset:

- *@leesyatt you are a cruel, cruel man. #therewillbeblood* ***#revenge****.*
- *can't believe Achilles killed me!* ***#angry****.*
- *Worst juror ever? Michelle. You were Nicole's biggest threat.* ***#bitter*** *#bb18.*
- *All hell is breaking loose in Charlotte. #CharlotteProtest* ***#anger*** *#looting.*
- *straight people are canoodling on the quad and I'm* ***#offended*** *.*

Thus, we try two different setting: first, processing hash-tags similar to all other words in the text;

[1] http://text-processing.com/api/sentiment/

[2] https://github.com/cjhutto/vaderSentiment

[3] https://nlp.stanford.edu/projects/glove/

[4] https://code.google.com/archive/p/word2vec/

[5] http://scikit-learn.org

Feature set	**Task**					
	EI-reg				**V-reg**	
	anger	*fear*	*joy*	*sadness*	Boost	RF
Baseline	30.83	30.76	43.07	31.67	50.02	40.66
Lexicon						
DepecheMood	16.08	19.00	27.69	10.15	24.57	18.34
Vader	39.89	42.07	**46.58**	**34.39**	52.51	45.40
All Lexicons	**42.91**	**42.31**	45.20	33.56	**54.02**	50.43
Embeddings						
Glove Twitter	54.55	51.38	43.44	52.21	65.97	56.90
GloVe Common Crawl	46.93	53.98	43.66	56.31	66.38	59.26
Google News	51.32	54.45	42.24	54.10	64.54	54.30
Glove Twitter + #	58.15	60.32	54.60	57.20	69.92	59.19
GloVe Common Crawl + #	54.92	61.33	53.73	59.00	70.43	64.05
Google News + #	53.09	59.42	55.77	57.15	67.44	56.38
All Embeddings	**59.01**	**62.97**	**56.42**	**60.33**	**70.48**	60.38
Combined features						
Lexicons + Baseline	44.93	48.63	50.40	41.70	60.12	56.03
Lexicons + Embeddings	**65.89**	65.82	59.90	65.64	73.00	64.96
Lexicons + Embeddings + Baseline	64.09	**66.95**	**63.80**	**65.73**	**72.35**	65.93

Table 2: Experimental results for on development set for two subtasks. Pearson correlation. Gradient Boosting Regressor is used for the EI-reg subtask. Gradient Boosting Regressor (Boost) and Random Forest (RF) is used for V-reg. # means that hash-tags are used separately as additional features.

second processing hash-tags separately to preserve authors' encoding of their emotions. The second strategy consistently yields better results as can be seen from Table 2.

4 Discussion

Comparisons of feature sets and algorithms are presented in Table 2. As can be seen from the table, results are consistent: emeddings yield higher performance than lexicon features for all tasks. DepechMode, even though it has five times more entries than Vader, seems to be less suitable for tweet emotion prediction and yields performance much lower than the baseline. Moreover, using both lexicons in combination not always improves performance and in some cases works even worse than Vader alone.

There is no significant difference between different embeddings. Various embeddings achieve better performance depending on the task, though the best results obtained by using all three in combination.

It can also be seen from Table 2 that separate treatment of hash tags improves model performance. For example, for joy detection task the difference is about 10%, which means that joy is frequently expressed explicitly in hash tags.

The best results for all tasks obtained by using all feature sets in combination (with the only exception of *angry* intensity detection subtask). This makes an improvement in 5.5% for *anger* detection subtask, 4% for *fear*, 7.5% for *joy*, 5.4% for *sadness*, and about 2% for sentiment intensity detection subtask. This means that even though lexicons cannot be used by themselves to detect emotions, they provide important features that cannot be extracted from embeddings. We hypothesize that the main reason for that is low *coverage*, meaning that many tweets have few lexicon features or no such features at all.

The coverage of the task corpora by various feature sets is presented in Table 3. It can be seen from the table that embeddings have much higher coverage than DepecheMood lexicon. Another interesting observation is that *GloVe Twitter* does not have a higher coverage than *GloVe Common Crawl* though *GloVe Twitter* has higher coverage of hash-tags.

Feature set	**Task**									
	EI-reg								**V-reg**	
	anger		*fear*		*joy*		*sadness*			
	≺	#	≺	#	≺	#	≺	#	≺	#
DepecheMood	53.4		52.5		53.6		54.74		53.1	
GloVe Common Crawl	**86.0**	6.2	**85.1**	6.2	**85.2**	4.9	**87.4**	4.8	**85.3**	4.5
GloVe Twitter	80.1	**6.5**	80.1	**6.5**	82.0	**5.2**	82.7	**5.1**	81.7	**4.7**
Google News	74.6	5.7	74.6	5.8	75.1	4.5	76.9	4.5	75.5	4.2

Table 3: Data coverage for various feature sets, percent of word usages. Legend: # - coverage of hash tags, ≺ - coverage of all other words.

Baseline	3
Lexicon features	
DepecheMood	8
Vader	4
Embeddings	
GloVe Common Crawl	300
Glove Twitter	200
Google News	300

Table 4: Feature sets and their dimensionality.

	All instances	**Gold in 0.5-1**
EI-reg		
Anger	65.4 / 82.7	52.6 / 70.8
Fear	67.2 / 77.9	49.7 / 60.8
Joy	64.8 / 79.2	42.0 / 56.8
Sadness	63.5 / 79.8	51.7 / 66.6
Macro-avg	**65.3** / 79.9	**49.0** / 63.8
V-reg		
	78.2 / 87.3	**62.1** / 69.7

Table 5: Official results for EI-reg (emotion intensity regression) and V-reg (valence intensity regression). Scores are given in the format X / Y , where X is our result, and Y is the best official result on the task. Pearson correlation.

Accuracy	**micro-avg F1**	**macro-avg F1**
47.7 / 58.8	**61.0** / 70.1	**41.6** / 52.8

Table 6: Official results for E-c (emotion classification) task. Scores are given in the format X / Y , where X is our result, and Y is the best official result on the task.

5 Results

The best model, used in our officially submitted solution, exploits all six feature sets plus separate embedding vectors for hash-tags. The list of feature sets and their dimensionality is presented in Table 4.

The official results for EI-reg and V-reg tasks are presented in Table 5. We report results for all instances and for instance with highest emotion intensity. The numerical values are similar to what we obtained on the development set. The official results for E-c classification task are presented in Table 6.

6 Conclusion

In this paper we presented our approach for SemEval Affect Detection in Tweets Task. We compare manually collected lexical features with embeddings automatically extracted from huge corpora. We demonstrated that even though lexicons are less suitable for affect detection in tweets due to low coverage they can improve model performance when lexical features are used together with more general embeddings.

In addition, we demonstrated that hash tags are important features for tweet affection detection, since they frequently include emotional words.

In this paper we used rather simplistic methods to combine various features, i.e., vector concatenation. In the future we plan to try another approach: to build a separate classifier for each feature set and then use a meta classifier on top of their results.

Repository

Repository with the code is located on the following URL link: https://github.com/dmikrav/SemEval2018AffectsTweets
The web-site to this project is on the following URL link: https://dmikrav.github.io/SemEval2018AffectsTweets/

References

Clayton J. Hutto and Eric Gilbert. 2014. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 Task 1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Saif M. Mohammad and Svetlana Kiritchenko. 2018. Understanding emotions: A dataset of tweets to study interactions between affect categories. In *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference*, Miyazaki, Japan.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Jacopo Staiano and Marco Guerini. 2014. Depeche mood: a lexicon for emotion analysis from crowd annotated news. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 427–433.