Journal of Machine Learning Research 20 (2019) 1-6 Submitted 7/18; Revised 5/19; Published 5/19

AffectiveTweets: a Weka Package for Analyzing Affect in Tweets

Felipe Bravo-Marquez FBRAVO@DCC.UCHILE.CL
Department of Computer Science, University of Chile & IMFD, Santiago, Chile

Eibe Frank EIBE@WAIKATO.AC.NZ
Bernhard Pfahringer BERNHARD@WAIKATO.AC.NZ
Department of Computer Science, University of Waikato, Hamilton, New Zealand

Saif M. Mohammad SAIF.MOHAMMAD@NRC-CNRC.GC.CA
National Research Council Canada, Ottawa, ON, Canada

Editor: Antti Honkela

Abstract

AffectiveTweets is a set of programs for analyzing emotion and sentiment of social media messages such as tweets. It is implemented as a package for the Weka machine learning workbench and provides methods for calculating state-of-the-art affect analysis features from tweets that can be fed into machine learning algorithms implemented in Weka. It also implements methods for building affective lexicons and distant supervision methods for training affective models from unlabeled tweets. The package was used by several teams in the shared tasks: EmoInt 2017 and Affect in Tweets SemEval 2018 Task 1.

Keywords: Twitter, Sentiment Analysis, Emotion Analysis, Affective Computing, Lexicon Induction, Distant Supervison

1. Introduction

Social media platforms such as Twitter are widely used to express affective states, often in relation to particular entities, topics or events. We use the term "affect" to encompass sentiment, emotion, mood, and other related mental states.

The automatic detection of affect from tweets[1] has become a popular task due to its importance in commerce, politics, and sports (Liu, 2012). Most successful solutions to this problem, including its many variations, are based on supervised machine learning models trained on manually annotated examples (Mohammad et al., 2018).

The process of building an affect detection system for tweets comprises three main components: 1) the training dataset on which the system is trained, 2) the function used for mapping tweets, provided as strings, into fixed size vectors of feature-value pairs, and 3) the machine learning algorithm used for training the model for detecting affect.

Replicating an affect detection system reported in a scientific publication can be a challenging endeavor. These systems usually include complex hand-engineered features and specific parameter setups for the learning algorithm (Mohammad et al., 2018). Unfortunately, these aspects are not always described in sufficient detail.

[1] We use the terms "social media post" and "tweet" interchangeably.

We present AffectiveTweets, a software for easily prototyping affect analysis systems for social media posts using Weka (Hall et al., 2009), a widely-used open source machine learning workbench implemented in Java.

2. The AffectiveTweets Package

The software is implemented as a Weka[2] package that can be installed with the Weka package manager. It can be accessed through Weka's GUIs or the command line interface. It is licensed under the GNU General Public License, Version 3 and hosted on Github.[3] Instructions for installing, using, testing, and contributing to the software are given on the project's website.[4]

The main functionalities of the package are implemented as Weka filters: tools enabling data transformations for the underlying machine learning task. AffectiveTweets provides three types of filters: 1) tweet-level filters, 2) word-level filters, and 3) distant supervision filters.

Other resources provided by the package are: training datasets, affective lexicons (lists of words annotated by affect), pre-trained word embeddings, and wrappers to the Twitter-specific tokenizer and POS tagger implemented in the *TweetNLP* library (Gimpel et al., 2011).

The package was made available as the official baseline system for the WASSA-2017 Shared Task on Emotion Intensity (EmoInt) (Mohammad and Bravo-Marquez, 2017) and for SemEval 2018 task 1 (Affect in Tweets) (Mohammad et al., 2018). It was used by 5 teams in the former task and by 15 teams in the latter.

2.1. Tweet-level Filters

Tweet-level filters in the package act on string attributes containing the text of tweets and calculate features suitable for further processing with the learning algorithms implemented in Weka. Many of these features have been used by state-of-the-art systems (Kiritchenko et al., 2014; Mohammad et al., 2018). The filters are described in the following.

The *TweetToSparseFeatureVector* filter calculates several sparse features for every tweet: word n-grams (adding a NEG prefix to words occurring in negated contexts), character n-grams, POS tags, and Brown word clusters. The scope of a negation is determined by a simple heuristic: from the occurrence of a negator word up until a punctuation mark or the end of the sentence. A list of negator words such as *no, not, won't* and *never* is provided with the package.

The *TweetToLexiconFeatureVector* filter calculates features by aggregating word-level affect scores taken from a fixed list of well-known affect lexicons. This list includes AFINN (Årup Nielsen, 2011), the Sentiment140 lexicon (Kiritchenko et al., 2014), and others. Users are encouraged to cite the publications describing any of the lexicons or resources used with the package.[5]

[2] `https://www.cs.waikato.ac.nz/ml/weka/`

[3] `https://github.com/felipebravom/AffectiveTweets`

[4] `https://affectivetweets.cms.waikato.ac.nz/`

[5] The BibTex entries of these publications are given at the package's website.

The *TweetToInputLexiconFeatureVector* filter allows users to calculate features from a tweet using their own affective lexicons in ARFF format.[6] The features are calculated by adding or counting the affect associations of the words matching the given lexicons. All numeric and nominal attributes from each lexicon are considered. Numeric scores are added and nominal labels are counted.

The *TweetToSentiStrengthFeatureVector* filter calculates positive and negative sentiment intensities for a tweet using the *SentiStrength* lexicon-based method (Thelwall et al., 2012).

The *TweetToEmbeddingsFeatureVector* filter calculates a tweet-level feature representation using pre-trained word embeddings (word vectors) by applying one of the following aggregation schemes: average of word embeddings; addition of word embeddings; or concatenation of the first k word embeddings in the tweet. The concatenation scheme is suitable for training deep neural networks with the *WekaDeeplearning4j* package (Lang et al., 2019).

2.2. Word-level Filters

The word-level filters allow users to build their own affective lexicons, which can later be used to calculate tweet-level features via the *TweetToInputLexiconFeatureVector* filter.

The *PMILexiconExpander* filter calculates the Pointwise Mutual Information (PMI) semantic orientation score for each word in a corpus of tweets annotated by sentiment (Turney, 2002). The score is calculated by subtracting the PMI of the target word with a negative sentiment from the PMI of the target word with a positive sentiment.

The *TweetCentroid* filter calculates word distributional vectors from a corpus of unlabeled tweets by treating words as the centroid of the tweet vectors in which they appear (Bravo-Marquez et al., 2015). The resulting word vectors can be labeled with an affective lexicon in ARFF format using the *LabelWordVectors* filter, making it possible to train a word-level affective classifier on these words. This classifier can be used to expand the original affective lexicon to the words from the provided corpus that are not covered by the lexicon. Word embeddings trained with neural networks can also be used in a similar way using filters provided by the *WekaDeepLearning4j* package.

2.3. Distant Supervision Filters

Distant supervision models are heuristic labeling functions used for automatically creating training data from unlabeled corpora. These models have been widely adopted for training affective models for tweets because large amounts of unlabeled tweets can easily be obtained through the use of the Twitter API. The filters described below take a collection of unlabeled tweets and a polarity lexicon of positive and negative words in ARFF format as input.

The *LexiconDistantSupervision* filter implements the most popular distant supervision approach for Twitter affective analysis: If a word from the lexicon is found, the tweet is labeled with the word's polarity. Tweets with both positive and negative words are discarded. The filter allows the removal of the lexicon word from the tweet's content. A list of positive and negative emoticons is used as the default lexicon (Read, 2005).

Two additional distant supervision methods that generate positive and negative instances by averaging tweet vectors annotated according to a given polarity lexicon are

[6] `https://waikato.github.io/weka-wiki/arff_stable/`

also provided. These methods have shown to be superior to the emoticon-based approach (Bravo-Marquez et al., 2016a),(Bravo-Marquez et al., 2016b).

The *ASA* filter implements the Annotate-Sample-Average (ASA) (Bravo-Marquez et al., 2016a) distant supervision method. It generates synthetic labeled instances by sampling a number of tweets containing at least one word from the lexicon with the desired polarity, and averaging the feature vectors of the sampled tweets.

The *PTCM* filter implements the Partitioned Tweet Centroid Model (PTCM) (Bravo-Marquez et al., 2016b). This method uses the tweet centroid model for calculating word vectors and a polarity lexicon for training a word-level classifier. The classifier is deployed on tweets represented by the original tweet-level representation used to calculate the word vectors. The model increases the number of labeled instances by partitioning the tweets associated with each word into smaller, disjoint subsets and calculating one centroid per partition, which is labeled according to the lexicon.

3. Related Tools and Benchmark

There is no existing tool for building affective models for tweets implementing all the functionalities provided by AffectiveTweets, but some related tools exist. The Stanford *CoreNLP* software allows training recursive deep models on a sentiment treebank (Socher et al., 2013). *SentiStrength* (Thelwall et al., 2012) and *Vader* (Hutto and Gilbert, 2014) are both lexicon-based systems that calculate sentiment intensity scores for social media messages. The *R* package *TM* (Meyer et al., 2008) for text mining has a plugin for sentiment classification based on an affective lexicon. *The Natural Language Toolkit* (NLTK) (Bird and Loper, 2004) provides a sentiment analysis module implementing the Vader method and the Bing Liu's lexicon.[7]

For demonstration, we benchmark *AffectiveTweets* against similar and equivalent models built using the NLTK sentiment analysis module and *Scikit-learn* (Pedregosa et al., 2011) on the dataset from the "SemEval 2013 Sentiment Analysis in Twitter Message Polarity Classification" task (Nakov et al., 2013). Classification results on the testing partition and execution times are shown in Table 1. Instructions for reproducing these experiments are given on the project's website.

Features	Implementation	Kappa	F_1	Time (seconds)
Word n-grams	Scikitlearn + NLTK	0.42	0.64	30.7
Word n-grams	AffectiveTweets	0.45	0.66	13.0
Word n-grams + Liu Lexicon	Scikitlearn + NLTK	0.48	0.68	13.4
Word n-grams + Liu Lexicon	AffectiveTweets	0.48	0.68	27.4
Liu Lexicon + Vader	Scikitlearn + NLTK	0.41	0.63	8.9
Liu Lexicon + SentiStrength	AffectiveTweets	0.40	0.63	31.9
Word n-grams + Liu Lexicon + Vader	Scikitlearn + NLTK	0.51	0.70	16
Word n-grams + Liu Lexicon + SentiStrength	AffectiveTweets	0.49	0.69	68.5
Word n-grams + All lexicons + SentiStrength	AffectiveTweets	0.52	0.71	74.6

Table 1: Benchmark results. The word n-grams include features for $n = 1, 2, 3$ and 4. Each model consists of a logistic regression trained on the corresponding features. Execution time is averaged over 10 repetitions.

[7] `http://www.nltk.org/api/nltk.sentiment.html`

4. Acknowledgments

Felipe Bravo-Marquez was funded by Millennium Institute for Foundational Research on Data.

References

Finn Årup Nielsen. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the 1st Workshop on Making Sense of Microposts (#MSM2011)*, pages 93–98, May 2011.

Steven Bird and Edward Loper. NLTK: The natural language toolkit. In *The Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics*, pages 214–217, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Felipe Bravo-Marquez, Eibe Frank, and Bernhard Pfahringer. From unlabelled tweets to Twitter-specific opinion words. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 743–746, 2015.

Felipe Bravo-Marquez, Eibe Frank, and Bernhard Pfahringer. Annotate-sample-average (ASA): A new distant supervision approach for Twitter sentiment analysis. In *Proceedings of 22nd European Conference on Artificial Intelligence*, pages 498–506, 2016a.

Felipe Bravo-Marquez, Eibe Frank, and Bernhard Pfahringer. From opinion lexicons to sentiment classification of tweets and vice versa: A transfer learning approach. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 145–152, 2016b.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah 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*, pages 42–47, 2011.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

Clayton J. Hutto and Eric Gilbert. 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*, pages 216–225, 2014.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762, 2014.

Steven Lang, Felipe Bravo-Marquez, Christopher Beckham, Mark Hall, and Eibe Frank. Wekadeeplearning4j: A deep learning package for weka based on deeplearning4j. *Knowledge-Based Systems*, 2019.

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

David Meyer, Kurt Hornik, and Ingo Feinerer. Text mining infrastructure in R. *Journal of Statistical Software*, 25(5):1–54, 2008.

Saif M. Mohammad and Felipe Bravo-Marquez. WASSA-2017 shared task on emotion intensity. In *Proceedings of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*, 2017.

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

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises*, pages 312–320, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

Jonathon Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, pages 43–48, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642. Association for Computational Linguistics, 2013.

Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012.

Peter 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*, pages 417–424. Association for Computational Linguistics, 2002.