TweetSenti: Target-dependent Tweet Sentiment Analysis

Quanzhi Li
Machine Intelligence Technology
Alibaba Group, Bellevue, WA,
USA, quanzhi.li@alibaba-inc.com

Qiong Zhang
Machine Intelligence Technology
Alibaba Group, Bellevue, WA,
USA, qz.zhang@alibaba-inc.com

Luo Si
Machine Intelligence Technology
Alibaba Group, Bellevue, WA,
USA, luo.si@alibaba-inc.com

ABSTRACT

TweetSenti is a system for analyzing the sentiment of an entity in tweets. A sentence or tweet may contain multiple entities, and they do not always have the same sentiment polarity. Therefore, it is necessary to detect the sentiment for a specific target entity. This type of target-dependent (entity level) sentiment analysis has become attractive and has been used in many applications, but it is still a challenging task. TweetSenti employs a new approach for detecting the entity level sentiment. Our model splits a sentence into a left context and a right context according to the target entity, and it also exploits two different types of word embeddings to represent a word, the general word embedding and the sentiment specific word embedding. A hybrid neural network is used to capture both the sequence and structure information of the two sides of the target entity. The sequence information is learned by attention-based bi-directional LSTM models. The structure information is captured by multi-context CNN models. Based on this algorithm, we built a web-based application that users can interact with and analyze an entity's sentiment in Twitter at real-time.

CCS CONCEPTS

• Computing methodologies~Natural language processing • Computing methodologies~Neural networks

KEYWORDS

Target-dependent sentiment, entity level sentiment classification, Twitter, social media

ACM Reference format:

Quanzhi Li, Qiong Zhang and Luo Si. 2019. TweetSenti: Target-dependent Tweet Sentiment Analysis. In *Proceedings of WWW '19: The Web Conference (WWW '19), May 13, 2019, San Francisco, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3308558.3314141

1 Introduction

With the permeation of social media in our lives, people are expressing their opinions about basically everything on social media platforms, such as Twitter, Facebook and Reddit. Sentiment analysis for entities, such as people, organizations, and products, has applications in various areas. Companies want to find out what consumers think of their products, brands, services or competitors. In public actions, it can be used to analyze online reactions to social and cultural phenomena. In politics, we can use it to keep track of society's opinions on politicians, the government, policy changes, or to predict election results.

There have been many studies on message or sentence level sentiment classification. Traditional sentiment classification approaches use sentiment lexicons to generate various features. Pang et al. [11] treat sentiment classification as a special case of text categorization. Many studies follow Pang's approach by designing features and applying different learning algorithms on them [4, 6, 10]. Feature engineering plays an important role in sentence sentiment classification [9]. Deep learning has also been used in sentence level sentiment analysis, mainly by exploiting word embedding [1, 8].

An entity in a sentence (post or tweet) does not necessarily have the same polarity type as the sentence, and different entities in the same sentence may have different polarities. For example, in the sentence of "*Farmers like Trump more than Hillary*", the two entities, *Trump* and *Hillary*, will have different sentiment polarities. There are some studies on entity level sentiment prediction [5, 2, 3, 14, 15, 18, 17] in the last several years. Jiang et al. [5] use both entity dependent and independent features to assign sentiment polarity to entities. By using POS features and the CRF algorithm, Mitchell et al. [7] identify polarities for people and organizations in tweets. Dong et al. [2] apply adaptive recursive neural network on the entity sentiment classification. The above two approaches use syntax parsers to parse tweets to generate features. Vo and Zhang [15] use target dependent and independent features, and also features based on lexicons.

Compared to the basic LSTM algorithm, TD-LSTM from [13] improves the sentiment classifier performance by treating an entity as a target. TC-LSTM from [14] extended TD-LSTM by adding target representations, which are word vectors, into the input of the LSTM cell unit. Several studies also explored applying attention mechanism with LSTM, and it has shown performance improvement [3, 16, 17, 18]. Wang et al [16] propose an attention-based LSTM model, ATAE-LSTM, and their experiment shows that attention mechanism is effective. The approach from [3] tries to enhance the attention-based LSTM by exploiting syntactic information to construct a syntax-based attention model.

In our TweetSenti system, we employed a new approach to address this problem. The main difference between our approach and previous studies are as follow: First, previous studies exploit only LSTM (with or without word attention), which captures mainly the sequence information of a sentence. Our approach uses a hybrid approach, which captures both the sequence and structure information of a sentence by integrating the attention-based bi-directional LSTM with the multi-context CNN model. Second, The context of an entity will affect its polarity, and usually an entity has a left context and also a right one. In our approach, we split a sentence into a left context and a right context according to the target entity, and we capture both contexts and also the whole sentence which captures the interaction between the left and the right context, through the multi-context CNN network structure. Third, previous studies use only the general word embedding (WE), such as embedding from word2vec [8] or C&W [1], to represent a word. Our approach uses not only WE but also the sentiment specific word embedding (SSWE) to represent a word.

In the demonstration, we will provide a highly interactive session, in which the audience can interact with the web application of our system. Several scenarios will be demonstrated.

2 TweetSenti System

In the following two subsections, we first describe the high-level workflow of TweetSenti, and then introduce how we build the entity sentiment classification module that supports the system.

2.1 TweetSenti Workflow

Figure 1 shows the high-level workflow of this application. We briefly introduce some of its components below.

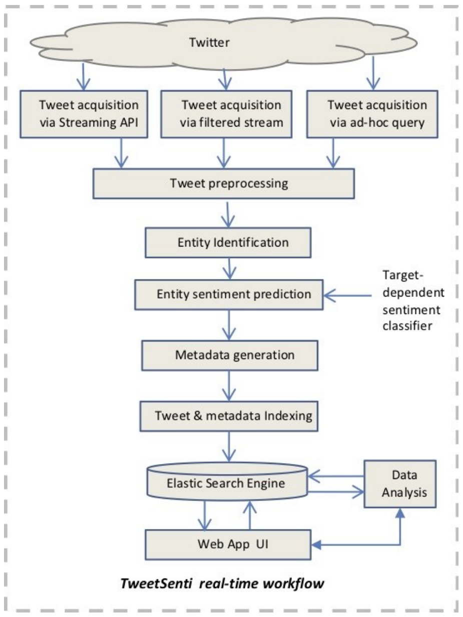

Twitter
Tweet acquisition via Streaming API
Tweet acquisition via filtered stream
Tweet acquisition via ad-hoc query
Tweet preprocessing
Entity Identification
Entity sentiment prediction
Target-dependent sentiment classifier
Metadata generation
Tweet & metadata Indexing
Elastic Search Engine
Data Analysis
Web App UI
TweetSenti real-time workflow

Figure 1: The high-level system real-time workflow

Tweet Acquisition. This system has three channels to acquire tweets from Twitter, all at real time. The main approach is through the Twitter streaming API, and this is done by our backend streaming tool at real-time. The second approach is via the filtered stream, which acquires tweets by monitoring a list of keywords, such as names of companies and political figures. These keywords can be modified at any time, and the system will make changes accordingly. Finally, users can submit queries via the system interface, and TweetSenti will retrieve related tweets on the fly from Twitter.

Tweet Preprocessing. In order for the sentiment classifier to predict the sentiment polarity for an entity, some preprocessing steps are necessary, such as normalizing numbers, and removing URLs and retweet mentions.

Entity Identification. To recognize entities from a tweet, we use TweetNLP package [20]. Some necessary preprocessing steps for entity identification is done by this package itself.

Entity Sentiment Prediction. For each entity extracted by the entity identification module, we use our entity sentiment classifier to predict its sentiment polarity. The polarity value and other metadata will be analyzed and presented to users by TweetSenti.

Metadata Generation. Some metadata are extracted or created in this step, and they will be displayed on UI or used for other analysis. For example, the tweet creation time is extracted and used for displaying the creation time, and generating the sentiment trending chart as well.

Data Indexing. A retrieved tweet and the generated metadata from it by the above components are ingested into an Elastic Search engine cluster (ES) [21] and indexed. We chose ES because it provides all the functions our application needs, and it has good scalability and reliability.

Data Analysis. This module is to analyze the data indexed and stored in the ES engine, and it also generates related information for TweetSenti to present them on the UI. For example, given a set of tweets talking about Trump, this model calculates how the sentiment for *Trump* have changed over a given period of time, and it also identifies a list of terms associated with the positive tweets and negative tweets. These information will be presented to users.

2.2 Entity Sentiment Classification Module

Figure 2 shows the construction process of the entity sentiment classification model. The following three subsections describe the three main components in the classification model construction process: the word embeddings, the framework of the classifier, and how the classifier is the trained.

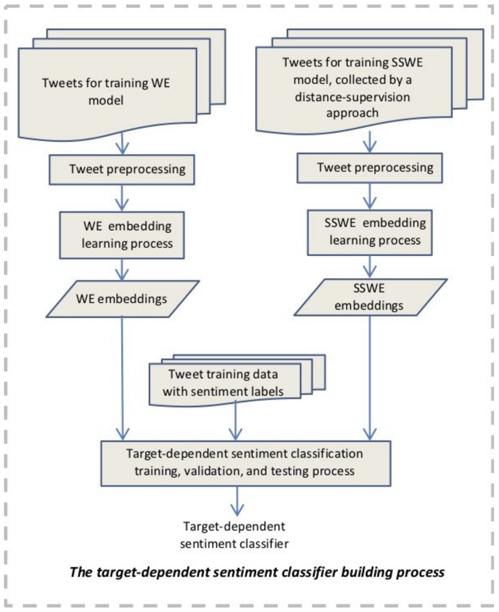

Tweets for training WE model
Tweets for training SSWE model, collected by a distance-supervision approach
Tweet preprocessing
Tweet preprocessing
WE embedding learning process
SSWE embedding learning process
WE embeddings
SSWE embeddings
Tweet training data with sentiment labels
Target-dependent sentiment classification training, validation, and testing process
Target-dependent sentiment classifier
The target-dependent sentiment classifier building process

Figure 2: The classification module construction process

2.2.1 Word Embedding.:As mentioned before, we concatenate the WE embedding and the SSWE embedding together to represent a word.

WE model. The general embedding of a word captures both the syntactic structure and semantics of the word. Traditional bag-of-words and bag-of-n-grams hardly capture the semantics of words. Word embedding has been used in many NLP tasks. word2vec [8] is used in our algorithm to generate the WE embedding. About 200 million tweets were collected from Twitter to build the embedding model. A total of 3 billion words were processed, and the embeddings were generated for 3.5 million unique terms. The embedding size, window size and word count threshold were empirically set as 300, 8 and 5, respectively. Appropriate preprocessing steps are conducted on each tweet before it is fed into the learning process, such as removing URLs, normalizing numbers, etc.

SSWE model. SSWE extends the C&W model [1] by incorporating the sentiment information into the neural network to learn the embedding; it captures the sentiment information of sentences as well as the syntactic contexts of words [12]. Given an original (or corrupted) n-gram and the sentiment polarity of a tweet as input, it predicts a two-dimensional vector *(f0, f1)*, for each input n-gram, where *(f0, f1)* are the language model score and sentiment score of the input n-gram, respectively. The training objectives are twofold: the original n-gram should get a higher language model score than the corrupted n-gram, and the polarity score of the original n-gram should be more aligned to the polarity label of the tweet than the corrupted one. The SSWE model was built from tweets using a distance-supervision approach, which uses emotion symbols to determine if a tweet is positive or negative. If the tweet contains one of the following positive symbols, it is treated as a positive tweet: *:) :-) =) :D ;)*. And it is negative tweet if it contains one of these symbols: *:-(:(=(;(*. Eventually, a total of ten million tweets were collected from Twitter, where 5 million of them contain positive emotions and the other 5 million contain negative ones. Mentions and URLs were removed from tweets. The embedding size, window size and word count threshold were empirically set as 50, 3 and 2, respectively [12].

2.2.2 The Classification Model To exploit both the sequence information and structure information of a sentence, we use a hybrid neural network, which combines the attention-based bi-directional LSTM and multi-context CNN models. The high-level network structure is shown in Figure 3. In Figure 3, R_F and R_B are the two attention-based LSTM vectors, learned from the forward and backward information, respectively. Given a target entity in a sentence, we split the sentence into two contexts: its left context and its right context. The target term is also included in each context . This is based on the assumption that the sentiment toward the target is decided by both contexts, and treating them separately will give us more information undiscovered before. C_{left} is the vector learned from the left context by a CNN model, and C_{right} is the vector learned from the right context also through a CNN model. C_{all} is generated from the whole sentence, which reflects the interaction between the left and right contexts of the entity. We concatenate these five vectors together, forming a vector V: $V = [R_F, R_B, C_{left}, C_{right}, C_{all}]$. Then, this vector is fed to the fully connected hidden layers, and a softmax layer is used to predict the sentiment polarity.

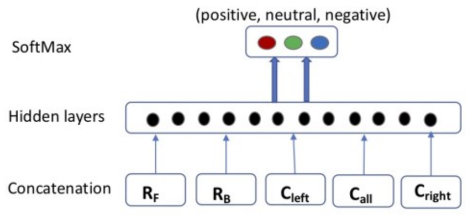


Figure 3: The high-level model structure

2.2.3 Model Training. Because our system works on Twitter, we use tweets to train and build our model. We combine two data sets together. The first data set is from Dong et al. [2014]. This data set was manually annotated with three sentiment polarity labels (positive, neutral and negative) toward target entities. It has 6248 training tweets and 692 testing tweets, with a balanced number of positive, negative, and neutral tweets (25%, 25%, and 50%, respectively). Dong et al. [2014] report that the data set was annotated with 82.5% inter-subject agreement among annotators. This data set has been used in many previous studies. The second data set is from SemEval 2017. It has a total of 50,333 tweets, with 19,902 positive tweets, 22,591 neutral, and 7,840 negative ones [19]. Combining two data sets gave us more training data.

Like previous studies, we use three-fold validation to tune our model and hyper-parameters. Stochastic gradient descent, AdaDelta update, shuffled mini-batch, back-propagation and dropout are used. The WE and SSWE embedding were fine-tuned during the training process of our hybrid neural network. Based on the validation result, we set the feature map (filter) width of the three CNN layers as 2 and 3, the number of feature map as 300, and batch size as 30.

To see how our model works compared to other methods, we conducted an experiment and compared it to two state-of-the-art entity level sentiment prediction algorithms, AB-LSTM2 from Yang et al. [18], which is an attention-based LSTM approach, and LSTM+SynATT+TarRep from He et al. [3], which is also a variant of the attention-based LSTM model. The result shows that our model slightly outperformed these two methods, with about 3% performance increase, using F1 measure that has been used in previous work [2, 5, 14, 17].

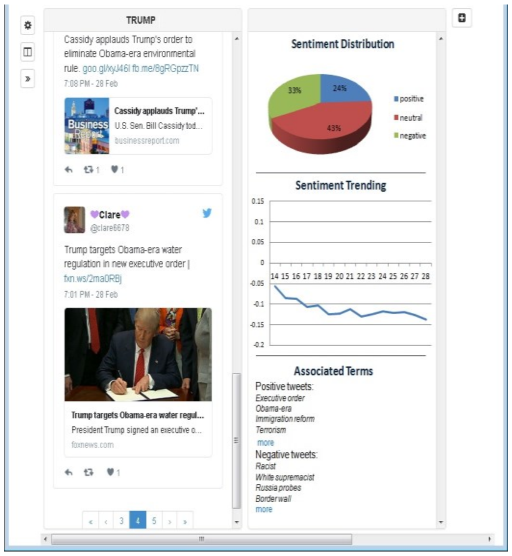
3 System Demonstration

We will provide a highly interactive demonstration, in which the audience can interact with the web application of our system. The following three scenarios will be demonstrated:

#1. A user can input a tweet, or just a sentence, and the system will extract all the entities from the text, analyze their sentiment, and display their sentiment scores. It will also show the sentiment difference between a specific entity and the tweet, i.e. entity level vs. tweet level sentiment.

#2. Our system is connected to Twitter's real-time streaming data. We will demonstrate the real-time sentiment change for some popular people and entities in some hot events that will be happening during the conference period, to make the demonstration more interesting.

#3. Users can also provide an entity (e.g. person, organization), and the system will retrieve the related tweets from the data sources described above. These tweets will be analyzed and the sentiment analysis is conducted for the provided entity at real-time. It will also show the sentiment trending change.



Cassidy applauds Trump's order to eliminate Obama-era environmental rule. goo.gl/xyJ46l fb.me/8gRGpzzTN
7:08 PM - 28 Feb

Cassidy applauds Trump'...
U.S. Sen. Bill Cassidy tod...
businessreport.com

Clare
@clare6678

Trump targets Obama-era water regulation in new executive order | fxn.ws/2ma0RBj
7:01 PM - 28 Feb

Trump targets Obama-era water regul...
President Trump signed an executive o...
foxnews.com

Positive tweets:
Executive order
Obama-era
Immigration reform
Terrorism

Negative tweets:
Racist
White supremacist
Russia probes
Borderwall

Figure 4: A screenshot showing the sentiment analysis results for "Trump". The panel on the left displays the related tweets, and the one on the right presents the sentiment analysis.

An example of scenario #3: Figure 4 shows how it works for scenario #3 mentioned above. In this example, "*Trump*" is the entity name input by a user. Our system gets the tweets related to Trump from the data sources described before, analyzes their sentiments toward *Trump*, and presents both the tweets and the sentiment analysis result in this window. This screenshot has two panels. The one on the left displays the tweets containing term Trump, and the one on the right presents three types of sentiment analysis results. The top one has a pie chart showing the sentiment distribution of all the tweets collected for *Trump* during the specified time period. Sentiment trending is displayed in the middle of this panel. It shows how the sentiment for *Trump* changed over this period of time. Our sentiment prediction algorithm can output a probability value for each polarity, which is a real number. The value for each day is the average sentiment value of all the related tweets from that day for *Trump*. At the bottom of the panel, it shows a list of terms associated with positive tweets and negative tweets. They are ordered by the strength of their association with the polarity type. By clicking the + icon at the top right corner of this window, one can add more panels to show sentiment analysis for multiple entities in the same window. If we change the time range to minutes or hours, and the "Auto Update" is set as On, we can see the sentiment information displayed in the right panel changing dynamically, as new tweets are being processed at real-time. To get the demo scenario #1 mode, users can click the setting icon at the top left corner and make corresponding changes.

REFERENCES

[1] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. Natural language processing (almost) from scratch. J. Mach. Learn. Res., 12:2493–2537.
[2] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, 2014. adaptive recursive neural network for target-dependent twitter sentiment classification. ACL.
[3] R. He, W. S. Leey, H. T. Ngy, and D. Dahlmeier, 2018. Effective Attention Modeling for Aspect-Level Sentiment Classification, COLING
[4] X. Hu, J. Tang, H. Gao, and H. Liu. 2013. Unsupervised sentiment analysis with emotional signals. In WWW, pages 607–618.
[5] L. Jiang, M. Yu, M. Zhou, X. Liu, T. Zhao, 2011. target-dependent twitter sentiment classification, ACL
[6] B. Liu. 2012. Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1):1–167
[7] M. Mitchell, J. Aguilar, T. Wilson, and B. V. Durme. 2013. Open domain targeted sentiment. EMNLP.
[8] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. NIPS.
[9] S. M. Mohammad, S. Kiritchenko, and X. Zhu. 2013. Nrc-canada: Building the state-of-the- art in sentiment analysis of tweets. SemEval 2013.
[10] Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. LREC.
[11] B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. EMNLP, pages 79–86
[12] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin. 2014. Learning sentiment specific word embedding for twitter sentiment classification. ACL, pages 1555–1565.
[13] D. Tang, B. Qin, X. Feng, and T. Liu. 2015. Target-Dependent Sentiment Classification with Long Short Term Memory. ArXiv preprint arXiv: 1512.01100 .
[14] D. Tang, B. Qin, X. Feng, T. Liu, 2016. Effective LSTMs for Target-Dependent Sentiment Classification, COLING
[15] D. Vo and Y. Zhang, 2015. Target-dependent twitter sentiment classification with rich automatic features, IJCAI
[16] S. Wang and C. D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. ACL, pages 90–94
[17] Y. Wang and M. Huang and L. Zhao and X. Zhu, 2016. Attention-based LSTM for Aspect-level Sentiment Classification, EMNLP

[18] M. Yang, W. Tu, J. Wang, F. Xu, X. Chen, 2017. Attention-Based LSTM for Target-Dependent Sentiment Classification, AAAI
[19] Sara Rosenthal, Noura Farra, Preslav Nakov, SemEval-2017 Task 4: Sentiment Analysis in Twitter. SemEval 2017.
[20] O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, et al.. Improved part-of-speech tagging for online conversational text with word clusters. NAACL2013.
[21] ElasticSearch, https://www.elastic.co/, 2016