

How Will Your Tweet Be Received? Predicting the Sentiment Polarity of Tweet Replies

Soroosh Tayebi Arasteh^{*†‡}, Mehrpad Monajem[†], Vincent Christlein[†], Philipp Heinrich[†], Angelos Nicolaou[†], Hamidreza Naderi Boldaji[†], Mahshad Lotfinia[§] and Stefan Evert[†]

[†]Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

[‡]Harvard Medical School, United States

[§]Sharif University of Technology, Iran

soroosh.arasteh@fau.de

Abstract—Twitter sentiment analysis, which often focuses on predicting the polarity of tweets, has attracted increasing attention over the last years, in particular with the rise of deep learning (DL). In this paper, we propose a new task: predicting the predominant sentiment among (first-order) replies to a given tweet. Therefore, we created RETWEET, a large dataset of tweets and replies manually annotated with sentiment labels. As a strong baseline, we propose a two-stage DL-based method: first, we create automatically labeled training data by applying a standard sentiment classifier to tweet replies and aggregating its predictions for each original tweet; our rationale is that individual errors made by the classifier are likely to cancel out in the aggregation step. Second, we use the automatically labeled data for supervised training of a neural network to predict reply sentiment from the original tweets. The resulting classifier is evaluated on the new RETWEET dataset, showing promising results, especially considering that it has been trained without any manually labeled data. Both the dataset and the baseline implementation are publicly available.

I. INTRODUCTION

Twitter is one of the dominant social media platforms and has become a major source for local and global news as well as political argumentation. Sentiment analysis is a research area in Natural Language Processing that aims to identify the opinions, attitudes or emotions expressed in a text document or sentence, often with respect to a particular topic [1]. Twitter sentiment analysis usually involves detecting whether a given tweet expresses a *positive*, *negative*, or *neutral* sentiment polarity, but has also looked at subparts of the tweet (words, phrases) or at aggregated sentiment in tweet collections, as well as sentiment towards a specific topic, e. g., a person, product, or event [2].

Nakov et al. [3] first introduced the task of message- and expression-level sentiment analysis on tweets; since then many further challenges have emerged. Rosenthal et al. [4] introduced the task of sentiment towards a topic. Variations of the task include more fine-grained classification (e. g., a five-point sentiment classification with labels *highly positive*, *positive*, *neutral*, *negative*, *highly negative*) and quantification of the overall distribution towards a given topic [5]. There is also work on sentiment analysis of figurative language [6] and detecting stance in tweets [7]. Other tasks such as out-of-context

sentiment intensity of words and phrases [8] and implicit event polarity [9] are closely related, but do not target tweets.

In this paper, we introduce a new challenge in Twitter sentiment analysis: given only an original tweet, predict the predominant polarity among all first-order replies this source tweet will receive, i. e., whether the audience will overall react positively, negatively, or neutrally to the tweet. It is challenging to collect large amounts of training data for this task because, for each training item, annotators need to read all first-order replies in order to label the source tweet. Therefore, we propose to generate automatically labeled training data by applying a standard Twitter sentiment classifier to sets of replies and aggregating the results. These data can then be used for standard supervised training of the final classifier. For evaluation purposes, we have created RETWEET, a publicly available¹ test set with manual annotation, and use it for validating our proposed method².

II. STANDARD TWEET SENTIMENT CLASSIFICATION

For our method presented in Sec. III, we need a state-of-the-art sentiment classifier for tweets. As a baseline, we evaluate models for task 10.B of SemEval 2015 [4], which is the task of three-way classification according to *positive*, *negative*, or *neutral* sentiment for English tweets. We use 51 875 manually labeled tweets from the SemEval datasets [2]–[5], [10] as training data and 9155 tweets as validation data. We combined individual training and test sets to ensure there is no overlap.

Our own classifier is based on a recurrent neural network. The tweets are tokenized using SpaCy [11]. The embedding layer uses a Vocabulary of the 50K the most frequent words in the training data; out-of-vocabulary words are replaced with a special “unknown” token (<unk>). The embeddings are initialised with a 200-dimensional GloVe [12] model trained on 27 billion tweets.

The main building-blocks of our model are two Bi-directional Long-Short Term Memory Units (BiLSTM) [13] with the hidden and cell states of length 256. First of all, a dropout layer is applied to the input embedding vectors. Then the result is

^{*} Corresponding author.

A major part of the research was carried out by the first two authors.

¹ Accessible under this link: <https://www.kaggle.com/soroosharasteh/retweet/>.

² The source code is accessible below: <https://github.com/starasteh/retweet/>.

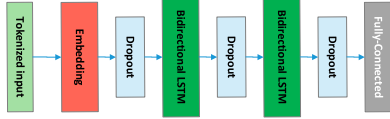


Fig. 1: The network architecture.

TABLE I: Evaluation of sentiment classifiers on the SemEval 2014 and 2015 official test sets. The scores present the average F1 scores of the positive and negative classes according to Equ. (1). Numbers taken from [4], [10], [14].

System	2014	2015
Logistic regression on 1-3 grams baseline	0.629	0.586
The 9th place of the original task	0.674	0.620
The winner of the original task	0.709	0.648
The state-of-the-art	0.748	0.688
<i>Our sentiment polarity classifier</i> (cf. Sec. II)	0.652	0.624

fed into a BiLSTM followed by another dropout layer and then the second BiLSTM followed by another dropout layer. Finally, a fully connected layer (followed by a SoftMax) is used for the classification. Fig. 1 shows our selected architecture. All dropout layers use a drop probability of 0.5. As loss, we chose the weighted cross-entropy with inverted class frequencies of the training data as loss weights to counteract the imbalanced dataset. The model is optimized using the Adam optimizer, learning rate of 10^{-4} and weight decay of 10^{-5} .

To compare our tweet classifier with other state-of-the-art models, we test our trained model on the 2014 and 2015 test sets of SemEval, which consist of 1852 and 2389 manually labeled tweets, respectively. In order to be consistent with historical editions of this competition, we use the average F_1 score of the positive and negative classes as the metric of interest, see Equ. (1). Note that, except accuracy, which is calculated on the whole set, the rest of the evaluation scores are all calculated according to Equ. (1) throughout this paper, i. e., discarding the neutral class results:

$$\text{score} = \frac{\text{score}_{\text{pos}} + \text{score}_{\text{neg}}}{2} \quad (1)$$

Tab. I shows that our tweet classifier is in the range of state-of-the-art and conventional machine learning models and we can proceed to utilize it for automatic labeling.

III. METHODOLOGY

Having a trained sentiment classifier, we can start solving the main problem of predicting overall sentiment polarity of the replies. In the automatic labeling process, given a source tweet and a set of corresponding replies, the trained sentiment classifier is first applied to all the replies as inputs, while ignoring the source tweet. Consequently, the network predicts a sentiment category for each reply, which are collected in a vector. A heuristic algorithm (cf. algorithm 1) is used to derive a final label for the overall sentiment of replies to the source tweet based on this label vector. Some justifications are needed to support our method, as there is no theoretical motivation or empirical evidence behind the chosen ratios.

Algorithm 1: Automatic label assignment strategy.

Result: One label for every tweet based on its replies.

for all tweets do

if #total neutral replies > 85% of the total replies

then

 Label = neutral ;

else

if #total positives replies > #1.5× of total negative replies **then**

 Label = positive ;

else if #total negative replies > #1.6× of total positives replies **then**

 Label = negative ;

else

 Label = neutral ;

Firstly, we generally expect most of the polarity predictions in the vector to be neutral as not all replies can be expected to carry positive/negative sentiment. Therefore, the overall reply sentiment is only considered to be neutral if the proportion of neutral replies exceeds a fairly high threshold, here set to 85%. Secondly, we observe that usually every tweet triggers both positive and negative replies as there are different people with different mindsets commenting on the tweet. In other words: there is usually no exclusively positive or negative reply vector for a tweet, even for seemingly clear-cut cases such as, e. g., birthday wishes. We therefore assign positive overall sentiment if there are at least 1.5 times as many positive replies as negative replies, and vice versa. Note that the exact values were fine-tuned based on various observations on the behavior of the network in different situations. Finally, we assign *neutral* to a tweet when the total numbers of positive and negative replies were close to each other, as none of them is dominant and there are two sets of contradicting replies, making it difficult to opt for one side.³

IV. RETWEET: DATASET OF TWEETS & OVERALL PREDOMINANT SENTIMENT OF THEIR REPLIES

As there is no publicly available dataset for our purposes, we downloaded data using the Twitter API. We collected 35 072 training tweets together with a total of 1 519 504 replies, and used 10% of them for validation. To download all of the replies to a tweet, the Search API could be used. However, it is limited to 75 000 requests per hour, which causes the mining and downloading process to be slow. Furthermore, using the Twitter API, there is no possibility of obtaining completely random data. Therefore, we tried to make the procedure as random as possible by utilizing two different strategies for data selection, using the collected data in an intermixed manner.

Our first strategy is based on a sample of English tweets obtained by filtering the Twitter stream via a list of cultural

³Note that despite the name the *neutral* class thus also contains tweets that trigger both positive and negative replies.

keywords [15]. This list consists of 147 words that are deemed to play a “pivotal role in discussions of culture and society” [15, xvii], covering diverse words such as *aesthetics*, *environment*, *feminism*, *power*, or *tourism*. We extracted all tweets in 2019 having a minimum of 20 first-order replies in the dataset. The data come with an obvious caveat: both the source tweet and replies must contain at least one word from the keyword list.⁴

As our second approach, we use the GetOldTweets3 [16] library to download all the replies corresponding to every tweet. To increase randomness, instead of using the same list of keywords as in the first strategy, we manually selected keywords that are likely to trigger long discussions, such as *Coronavirus* and *football*, or strong opinions, such as *birthday*, *war*, or *racism*. In addition to the constraint that every tweet must have at least 20 first-order replies, we only accepted tweets (both source tweets and replies) consisting of at least 20 tokens. This is because our tweet classifier (cf. Sec. II) is optimized based on the message-level classification paradigm and thus relies on a sufficient number of words in the message.

A total of 5015 tweets with all of their corresponding first-order replies, collected by a combination of the two collection strategies, were given to three trained annotators. Annotators were asked to judge intuitively whether the replies taken together indicate an overall positive, negative or neutral reaction to the tweet and decide on *one* final sentiment label for the replies. Annotators were not shown the respective source tweet in order to avoid a bias based on their own reaction to the tweet. *Positive* and *negative* polarity is defined in the same way as for the SemEval tasks. In addition to tweets which actually have a neutral overall polarity among their first-order replies, annotators were asked to also assign neutral label to a tweet when neither positive nor negative replies have predominance on the other one. For the final dataset, we only chose tweets for which all annotators judged unanimously, leading to a data set of 1519 source tweets manually labeled with the overall sentiment of their replies (excluding sentiment of the source tweet itself).

During the manual annotation process, we observed that most disagreements between the annotators were related to neutral labels, i. e., either one annotator classified the overall label of tweet as neutral and the other two not, or two classified as neutral and the other one not. Although the annotators were trained in a same way, the concept of neutral label was still not as clear as positive/negative to them. This can be because human brain, when trying to annotate such phrases, expects more to face phrases with positive or negative polarities than neutral and is more sensitive towards these polarities. Thus, chance of having different opinions between negative and positive is very lower than between neutral and positive/negative. Label distribution in our final test set is as follows: 32.5 % positive, 37.5 % negative, and 30.0 % neutral; the training set contains 23.5 % positive, 32.5 % negative, and 44.0 % neutral labels.

⁴This makes it highly unlikely that the list of replies for any given source is exhaustive, i. e., there will usually be many other first-order replies to the source tweet that are not included in the dataset.

Accuracy	F1 score	Recall	Precision
61.7	71.9	79.1	66.1

		(a) Predicted label		
		Neutral	Positive	Negative
True label	Neutral	90	117	243
	Positive	61	387	50
	Negative	71	40	460

(b)

TABLE II: (a) evaluation results of the final classifier on the RETWEET test set. F1 score, recall, and precision are calculated based on Equ. (1). All results given in percent. (b) Detailed confusion matrix.

V. EXPERIMENTS AND RESULTS ON RETWEET

We utilize the same data pre-processing, training process, and architecture as for the standard Twitter sentiment classifier (Sec. II); the only differences being that the lengths of hidden and cell states are 300, vocabulary is built choosing 750K of the most common words of the training data, and a learning rate of 9×10^{-5} with a decay of 10^{-4} is chosen.

Tab. II shows the confusion matrix and evaluation results of the final classifier on the RETWEET test set. Even though the standard sentiment polarity classifier (Sec. II) used for automatic labelling of the training data (Sec. III), our classifier for response sentiment prediction already achieves highly promising results, especially considering the general difficulty of message-level sentiment analysis and the indirect nature of the RETWEET task.

In order to explore the correlation between tweet sentiments and their reply sentiments, we additionally create a “direct” baseline classifier by predicting the sentiment of the original tweet and assuming that its replies will have the same predominant sentiment. Tab. III shows the evaluation metrics of our proposed system versus the “direct” baseline on the RETWEET test set. We observe a highly significant increase in F_1 score by 15.4 percent points.

Furthermore, we follow the state-of-the-art model of standard tweet sentiment polarity classification proposed by Cliche [14], implementing a Convolutional Neural Network-based model and averaging its predictions with those of our BiLSTM model.

Firstly, each input embedding vector is fed to three 1D convolutional layers with filter sizes of 3, 4, and 5, respectively, and the Rectified Linear Unit (ReLU) is used as activation function. All filters have the same output feature map dimension of 200. Moreover, in order to remove the dependency of the feature maps on the length, we additionally feed them to 1D max pooling layers with kernel sizes of equal to sentence lengths of each activation. Finally, the results are concatenated, leading to a 600-dimensional feature map for every input,

TABLE III: Comparison between our proposed method (Sec. III) and directly classifying the tweet and assuming its replies will have the same dominant polarity as itself. F1 score, recall, and precision are calculated based on Equ. (1). All results given in percent.

Metric	Direct	Proposed
Accuracy	49.7	61.7
F1 score	56.5	71.9
Recall	54.0	79.1
Precision	60.0	66.1

TABLE IV: Evaluation results of the ensemble model (cf. Sec. V) on the RETWEET test set. F1 score, recall, and precision are calculated based on Equ. (1). All results given in percent.

Accuracy	F1 score	Recall	Precision
62.0	73.2	81.0	66.8

followed by a dropout, and a fully connected layer (followed by a SoftMax) is used for the classification.

We train the CNN independently with similar optimization and data pre-processing parameters as for our BiLSTM model. The evaluation results on the RETWEET test set (see Tab. IV), unexpectedly, show only a very slight improvement over the BiLSTM model, which indicates the fact that having already a good enough classifier architecture, the choice of the label assignment algorithm plays a more vital role than maximizing the capacity of the classifier architecture in this task.

Furthermore, Tab. IIb shows that most of the wrong classification results have actually neutral reply polarity. It is because, using algorithm 1, we make our system sensitive to positive/negative sentiments in order to boost our performance of interest. And it does not hurt us to lose to a certain extent the neutral prediction performance as long as we are improving in our primary goal.

VI. CONCLUSION AND DISCUSSION

In this paper, we introduced a new challenge: prediction of the overall polarity of first-order replies to an English source tweet. As a baseline for solving this task, we proposed a Deep Learning approach. We first predicted the overall message-level polarity of the tweets, i. e., whether it is received positive, negative, or neutral, which led to creation of automatic labels for replies, and subsequently trained a network that predicts the predominant reaction of the tweet audience. We used a heuristic algorithm for the final label selection in the automatic labeling process.

Additionally, we created RETWEET, the first public dataset for sentiment prediction of first-order Twitter replies. The evaluation results of our approach on RETWEET show its effectiveness. Although utilizing more training data may still somewhat increase the performance, whether a reply is positive or negative, except for extreme cases, may not purely be determined only by the content of the tweet itself but in a rather dynamic environment. Therefore, our proposed method serves as an upper-bound baseline in prediction of the polarity of predominant first-reaction to a given tweet.

As mentioned in Sec. IV, the RETWEET is extracted in a way to be as random as possible, since we do not consider the dynamic environment and rather only do content-based prediction. For future work, it would be beneficial to extend the method by taking some prior knowledge into consideration. Time of posting, status quo when the tweet is posted, etc., can be influential factors. Consequently, more task-specific data collection strategies will be explored to extend RETWEET.

REFERENCES

- [1] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.
- [2] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 task 4: Sentiment analysis in twitter," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 502–518.
- [3] P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter, and T. Wilson, "SemEval-2013 task 2: Sentiment analysis in twitter," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: Association for Computational Linguistics, Jun. 2013, pp. 312–320.
- [4] S. Rosenthal, S. M. Mohammad, P. Nakov, A. Ritter, S. Kiritchenko, and V. Stoyanov, "SemEval-2015 task 10: Sentiment analysis in twitter," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval-2015)*. Denver, Colorado: Association for Computational Linguistics, 2015, pp. 451–463.
- [5] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, "SemEval-2016 task 4: Sentiment analysis in twitter," in *Proceedings of the 10th International Workshop on Semantic Evaluation*. San Diego, California: Association for Computational Linguistics, 2016, pp. 1–18.
- [6] A. Ghosh, G. Li, T. Veale, P. Rosso, E. Shutova, J. Barnden, and A. Reyes, "SemEval-2015 task 11: Sentiment analysis of figurative language in twitter," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, 2015, pp. 470–478.
- [7] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "SemEval-2016 task 6: Detecting stance in tweets," in *Proceedings of the 10th International Workshop on Semantic Evaluation*. San Diego, California: Association for Computational Linguistics, 2016, pp. 31–41.
- [8] S. Kiritchenko, S. Mohammad, and M. Salameh, "SemEval-2016 task 7: Determining sentiment intensity of English and Arabic phrases," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, 2016, pp. 42–51.
- [9] I. Russo, T. Caselli, and C. Strapparava, "SemEval-2015 task 9: CLIPeval implicit polarity of events," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, 2015, pp. 443–450.
- [10] S. Rosenthal, P. Nakov, A. Ritter, and V. Stoyanov, "SemEval-2014 task 9: Sentiment analysis in twitter," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*. Dublin, Ireland: Association for Computational Linguistics, 2014, pp. 73–80.
- [11] M. Honnibal and M. Johnson, "An improved non-monotonic transition system for dependency parsing," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1373–1378.
- [12] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.
- [13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [14] M. Cliche, "BB_twttr at SemEval-2017 task 4: Twitter sentiment analysis with CNNs and LSTMs," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 573–580.
- [15] T. Bennett, L. Grossberg, and M. Morris, *New Keywords: A Revised Vocabulary of Culture and Society*. Blackwell, 2005.
- [16] D. Mottl and J. Henriquel, "Getoldtweets3," 2019. [Online]. Available: <https://pypi.org/project/GetOldTweets3/>