FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

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

Soroosh Tayebi Arasteh[1,2], Mehrpad Monajem[1], Vincent Christlein[1], Philipp Heinrich[1], Anguelos Nicolaou[1], Hamidreza Naderi Boldaji[1], Mahshad Lotfinia[3] and Stefan Evert[1]

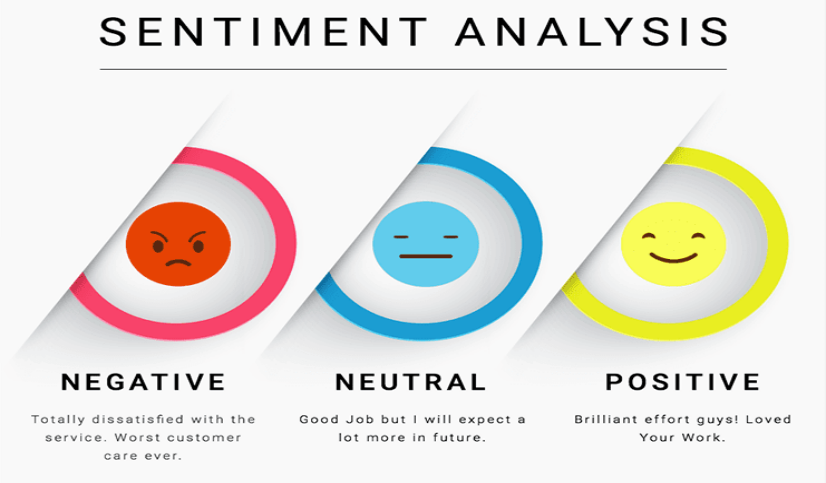
[1]Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
[2]Harvard Medical School, United States [3]Sharif University of Technology, Iran

Presented by: Soroosh Tayebi Arasteh

1. Overview

2. Standard Tweet Sentiment Polarity Classification
3. Methodology
4. RETWEET
5. Experiments and Results on RETWEET
6. Conclusion

Overview: What is Sentiment Analysis?

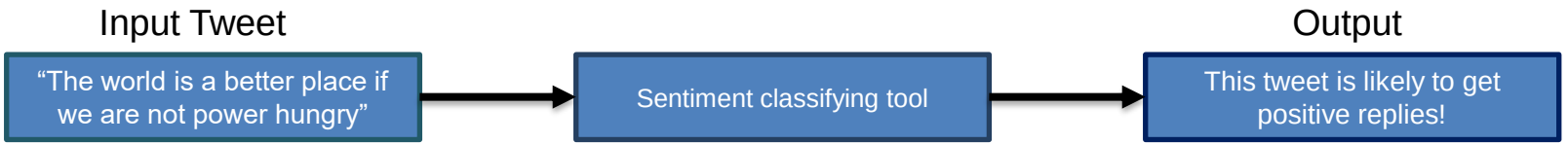


- Sentiment analysis is a research area in Natural Language Processing that aims to identify the opinions, attitudes or emotions expressed in a text document or sentence, often with respect to a particular topic.

* B. Liu, “Sentiment analysis and opinion mining,” Synthesis Lectures on Human Language Technologies, vol. 5, no. 1, pp. 1–167, 2012.
** Image source: https://www.newgenapps.com/blog/the-secret-way-of-measuring-customer-emotions-social-media-sentiment-analysis/

Overview: Goal

- Tremendous amounts of unorganized public text data are being created every day, with **Twitter status messages ("tweets")** being one of the main examples.
- 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.
- The aim of this paper is to build a sentiment classifying tool which, given a tweet predicts how likely it gets positive, negative or neutral replies.
  - ✓ In other words, it is a tool to predict the sentiment of replies of a tweet is most likely to get.
- Here we do **Message-level** sentiment analysis, i.e., we classify the sentiment of the whole given message (tweet or reply).



Overview: Data Pre-Processing

- The **maximum vocabulary** size of **50K** for Standard Tweet Sentiment Classifier and **750K** for the main method**.**
- Everything else will be ragarded as the unkown token *UNK_IDX*.
- And one extra padding token *PAD_IDX*.
- **Only on the training set** and not the validation set.
- Words in the vocabulary **initialized** with **Gaussian distribution**.

Overview: Data Pre-Processing

- The **maximum vocabulary** size of **50K** for Standard Tweet Sentiment Classifier and **750K** for the main method**.**
- Everything else will be ragarded as the unkown token *UNK_IDX*.
- And one extra padding token *PAD_IDX*.
- **Only on the training set** and not the validation set.
- Words in the vocabulary **initialized** with **Gaussian distribution**.

- **Tokenizer** using **SpaCy** with ***Pack-Padded-Sequence***.

- Embedding layer:
  - ✓ Using the pre-trained ***glove.twitter.27B.200d*** (200-dimensional **GloVe** model on 27 billion tweets).

* Honnibal et al., “An improved non-monotonic transition system for dependency parsing”, in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.
** Pennington et al., Glove: Global vectors for word representation”, in EMNLP, 2014.

Standard Tweet Classification: Overview

- For our method, we will need a state-of-the-art sentiment classifier for tweets.
  - ✓ As a baseline, we evaluate models for **task 10.B of SemEval 2015**,
  - ✓ Which is the task of three-fold classification according to ***positive*****,** ***negative*****,** or ***neutral*** sentiment for **English** tweets.

- SemEval (Semantic Evaluation) is an ongoing series of evaluations of computational semantic analysis systems; it evolved from the Senseval word sense evaluation series. (Wikipedia)

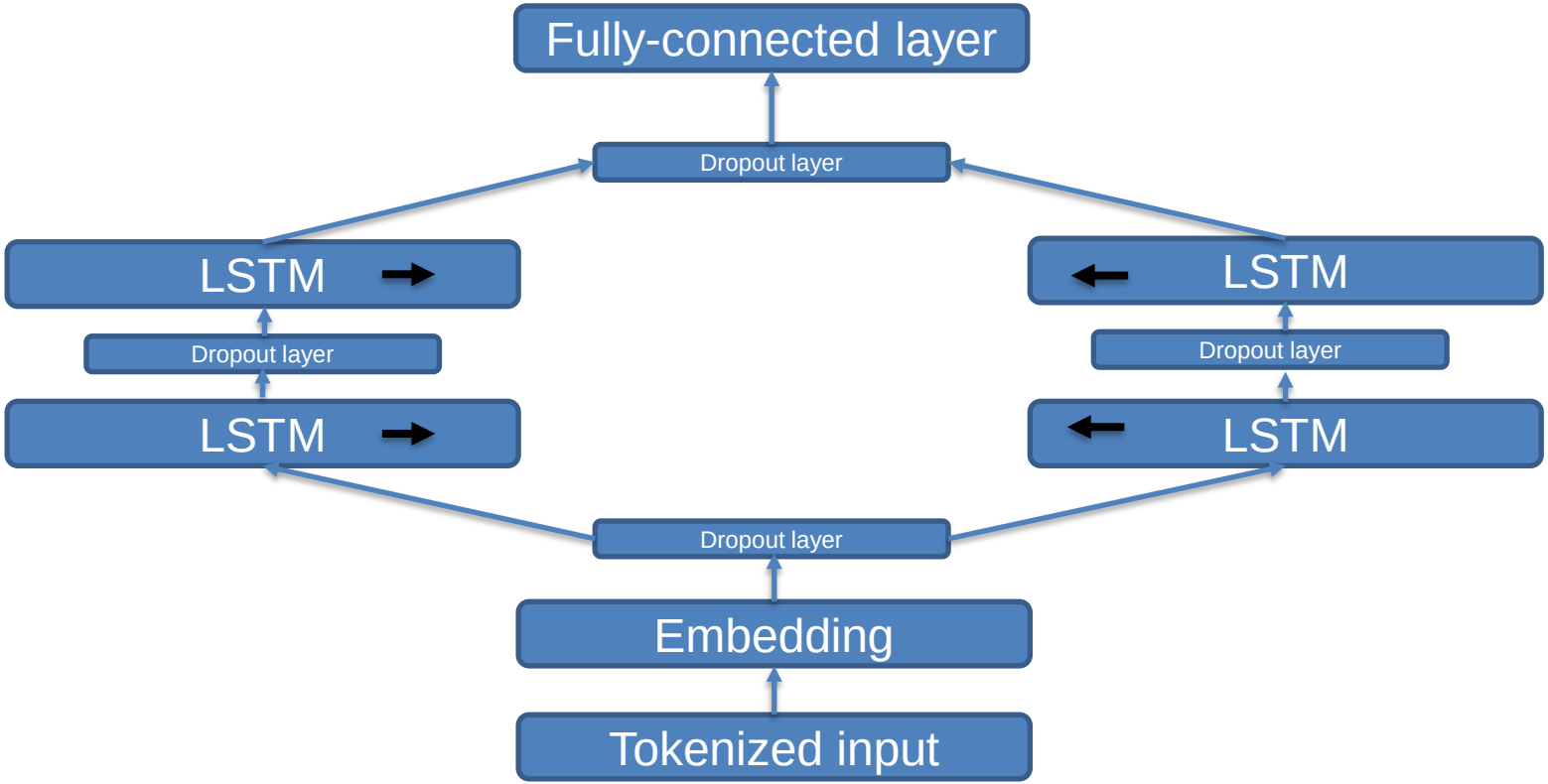
- The datasets contain annotated tweets with a variety of topics including politics, social issues, products, movies, events etc.

Standard Tweet Classification: Dataset

- **Training data**
  - ✓ **51,875** manually labeled tweets from the SemEval dataset (2013-2017).
- **Validation data**
  - ✓ **9,155** manually labeled tweets from the SemEval dataset (2013-2017).
- **Test data**
  - ✓ Official Test gold dataset of SemEval 2014: **1,852 tweets.**
  - ✓ Official Test gold dataset of SemEval 2015: **2,389 tweets**

* Nakov et al., “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 Workshopon Semantic Evaluation (SemEval 2013), 2013.
** Rosenthal et al., “SemEval-2014 task9: Sentiment analysis in twitter”, in Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014), 2014.
*** Rosenthal et al., “SemEval-2015 task 10: Sentiment analysis in twitter”, in Proceedings of the 9th International Workshop on Semantic Evaluation(SemEval-2015), 2015.
**** Nakov et al., “SemEval-2016 task 4: Sentiment analysis in twitter”, in Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016.
***** Rosenthal et al., “SemEval-2017 task 4: Sentiment analysis in twitter”, in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017.

Standard Tweet Classification: Model Architecture



Standard Tweet Classification: Model Parameters

- Main building-blocks of our model: two Bi-directional Long-Short Term Memory units (**BiLSTM**).
- **Hidden & cell** dimensions of the BiLSTMs: **256**
- Embedding dimmension: 200
- **Drop probability** of all the Dropout layers: **0.5**
- Loss function: **Cross Entropy loss**
- Optimizer: **ADAM**, with a **learning rate of 1e-4** with weight decay of 1e-5

* S. Hochreiter and J. Schmidhube., ““Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

Quantitative Analysis Approach

- In order to be consistent with historical editions of the SemEval, we use the **average F1 scores of the positive and negative** classes as the metric of interest throughout the paper:

$$score = \frac{score_{pos} + score_{neg}}{2}$$

- The overall F1 score for each class is calculated according to the following equations and total number of true positive (TP), false positive (FP), and false negative (FN) values are used for the calculations:

$$Recall = \frac{TP}{TP+FN} \qquad Precision = \frac{TP}{TP+FP} \qquad F_1 = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision}$$

Standard Tweet Classification: Evaluation Results

- 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.

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
Our sentiment polarity classifier	**0.652**	**0.624**

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

* M. Cliche., ""BB_twtr at SemEval-2017 task 4: Twitter sentiment analysis with CNNs and LSTMs", in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017).

Methodology: Overview

- **Originally Unsupervised** problem.

- **Idea**: To leverage the Standard Tweet Sentiment Polarity Classification, to tackle the unsupervised nature of the problem.

Methodology: Steps

1. Extract tweets from Twitter with their corresponding replies.

Methodology: Steps

1. Extract tweets from Twitter with their corresponding replies.

2. If a tweet has multiple replies, regard each reply as a separate data and repeat the tweet.

Methodology: Steps

1. Extract tweets from Twitter with their corresponding replies.

2. If a tweet has multiple replies, regard each reply as a separate data and repeat the tweet.

3. First ignore the tweets and predict the sentiment of each reply using the trained Standard Tweet Sentiment Classifier.

Methodology: Steps

1. Extract tweets from Twitter with their corresponding replies.

2. If a tweet has multiple replies, regard each reply as a separate data and repeat the tweet.

3. First ignore the tweets and predict the sentiment of each reply using the trained Standard Tweet Sentiment Classifier.

4. For the replies corresponding to the same tweet, choose the label which has the maximum occurance and assign it as the tweet‘s label and then ignore the replies.

Methodology: Steps

1. Extract tweets from Twitter with their corresponding replies.

2. If a tweet has multiple replies, regard each reply as a separate data and repeat the tweet.

3. First ignore the tweets and predict the sentiment of each reply using the trained Standard Tweet Sentiment Classifier.

4. For the replies corresponding to the same tweet, choose the label which has the maximum occurance and assign it as the tweet‘s label and then ignore the replies.

5. Now we have a training set of some tweets with their labels, SUPERVISED!

Methodology: Steps

1. Extract tweets from Twitter with their corresponding replies.
2. If a tweet has multiple replies, regard each reply as a separate data and repeat the tweet.
3. First ignore the tweets and predict the sentiment of each reply using the trained Standard Tweet Sentiment Classifier.
4. For the replies corresponding to the same tweet, choose the label which has the maximum occurance and assign it as the tweet‘s label and then ignore the replies.
5. Now we have a training set of some tweets with their labels, SUPERVISED!
6. Train another model with this data.

Methodology: Steps

1. Extract tweets from Twitter with their corresponding replies.
2. If a tweet has multiple replies, regard each reply as a separate data and repeat the tweet.
3. First ignore the tweets and predict the sentiment of each reply using the trained Standard Tweet Sentiment Classifier.
4. For the replies corresponding to the same tweet, choose the label which has the maximum occurance and assign it as the tweet‘s label and then ignore the replies.
5. Now we have a training set of some tweets with their labels, SUPERVISED!
6. Train another model with this data.
7. Now the final model is ready. Given only tweets as the input, this model predicts the sentiment of the potential replies that tweet is most likely to receive!

Methodology: Automatic Label Assignment (Step 4)

- In the **step 4 of the general strategy**, we generally expect most of the polarity predictions to be neutral as not all the tweets carry positive or negative sentiments.

- There is usually no absolute positive or negative reply vector for a tweet, even for extreme cases, e. g., birthday wishes.

Methodology: Automatic Label Assignment (Step 4)

- In the **step 4 of the general strategy**, we generally expect most of the polarity predictions to be neutral as not all the tweets carry positive or negative sentiments.
- There is usually no absolute positive or negative reply vector fora tweet, even for extreme cases, e. g., birthday wishes.
- ✓ Therefore, a heuristic algorithm is used here to choose a final label representing the whole label vector to avoid all the labels being neutral.

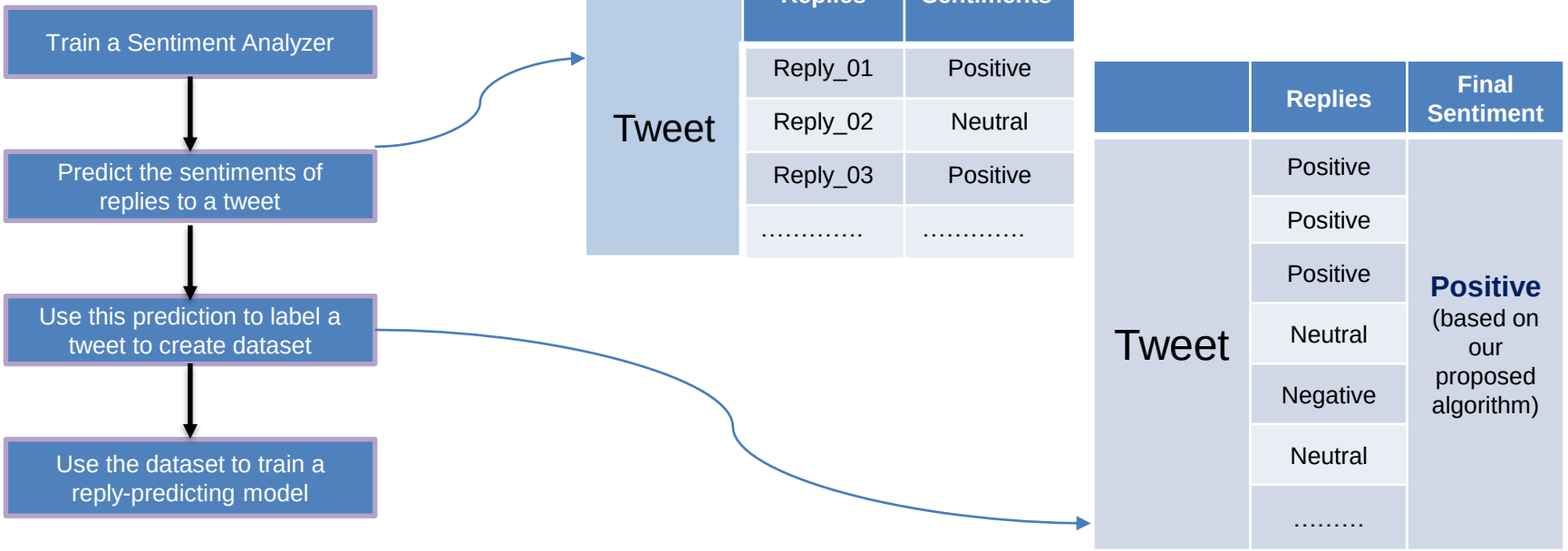
Methodology: Automatic Label Assignment (Step 4)

For all tweets, we set a tweet to:

- **Neutral:** if the proportion of neutral replies is **larger than 85%.**

- Otherwise:
  - \+ **Positive**: if the number of total positive replies is **larger than 1.5** of the total number of negative replies.
  - \- **Negative**: if the number of total negative replies is **larger than 1.6** of the total number of positive replies.
  - * **Neutral**: else

Method in Nutshell



1. Overview
2. Standard Tweet Sentiment Polarity Classification
3. Methodology
4. **RETWEET**
5. Interactive Sentiment Analysis
6. Conclusion

RETWEET: Overview

- As there is no available dataset, we downloaded data ourselves using the Twitter API, which led to the creation of the ***RETWEET*** dataset.

- Total of **35,072 training tweets** collected.

  - ✓ Corresponding to **1,519,504 replies** in total,
  - ✓ 10% of them used for validation.

RETWEET: Data Extraction

- To download all the replies to a tweet, the **Twitter Search API** should be used.

- **Limitations:**

  - ✓ The Search API is limited to **only 75,000 requests per hour**, which causes the mining and downloading process to be slow.

  - ✓ Using the Twitter API, there is no possibility of downloading absolute random data.

RETWEET: Data Extraction

- To download all the replies to a tweet, the **Twitter Search API** should be used.
- **Limitations:**
  - ✓ The Search API is limited to **only 75,000 requests per hour**, which causes the mining and downloading process to be slow.
  - ✓ Using the Twitter API, there is no possibility of downloading absolute random data.
- Thus, we tried to make the procedure **as random as possible** by utilizing two different strategies for data selection, in an intermixed manner.

RETWEET: Data Extraction Methods

1. List of keywords

- Based on a sample of English tweets obtained by filtering the Twitter stream via a list of cultural keywords.
- Consisting of **147 words** that are deemed to play a "*pivotal role in discussions of culture and society*".
- Covering diverse words such as *environment, feminism, power, tourism,* or *youth.*
- Every tweet must contain **at least 20 first-order replies**.
- Both the source tweet as well as all the replies must contain at least one word from the list of keywords.
  - ✓ Therefore, there might be many more first-order replies to the source tweet that are not in the dataset.

* Bennett et al., "New Keywords: A Revised Vocabulary of Culture and Society," Blackwell, 2005.

RETWEET: Data Extraction Methods

2. GetOldTweets3

- It is a command line tool which only downloads tweets.
  - ✓ However, by modifying it, we can extract all the replies of each tweets as well.
- To increase randomness, we manually chose keywords,
  - ✓ which are most likely to include long discussions, such ***Coronavirus*** and ***football***
  - ✓ or ones which are most likely to contain strong opinions such as ***birthday***, ***war***, or ***racism***.
- Every tweet must contain **at least 20 first-order replies**.
- Every **tweet** and also every **reply** should contain **at least 20 strings**.
  - ✓ Because our standard tweet classifier is optimized based on the message-level classification paradigm and thus relies on a sufficient number of words in the message.

* D. Mottl and J. Henriquel, “Getoldtweets3,” 2019. Available: https://pypi.org/project/GetOldTweets3/

RETWEET: Manual Annotations

- Initially, total of **5015 tweets** with all their corresponding first-order replies collected.

- 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 for the replies without observing the original tweet,
  - ✓ in order to avoid having prior knowledge.
- **Positive** and **negative** polarity is defined in the same way as for the **SemEval** tasks.
- **Neutral** label **when neither positive nor negative** replies have predominance on the other one.

- Only tweets for which all annotators judged unanimously, were chosen,

RETWEET: Manual Annotations

- Initially, total of **5015 tweets** with all their corresponding first-order replies collected.

- 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 for the replies without observing the original tweet,
  - ✓ in order to avoid having prior knowledge.
- **Positive** and **negative** polarity is defined in the same way as for the **SemEval** tasks.
- **Neutral** label **when neither positive nor negative** replies have predominance on the other one.

- Only tweets for which all annotators judged unanimously, were chosen,

  ✓ leading to a test set of **1519 manually labeled tweets.**

RETWEET: Class Distributions

- Class distributions in the training set, after applying the Automatic Label Assignment strategy, and in the RETWEET test set.

RETWEET Subset	Positive	Negative	Neutral
Training	23.5%	32.5%	44.0%
Test	32.5%	37.5%	30.0%

Experiments: Model Parameters

- The same architecture as for the Standard Tweet Classification model, with the following parameters:

- **Hidden & cell** dimensions of the BiLSTMs: **300**
- Embedding dimmension: 200
- Drop probability of all the Dropout layers: 0.5
- Loss function: Cross Entropy loss
- Optimizer: ADAM, with a **learning rate of 9e-5** with **weight decay of 1e-4**.

Experiments: Evaluation Results

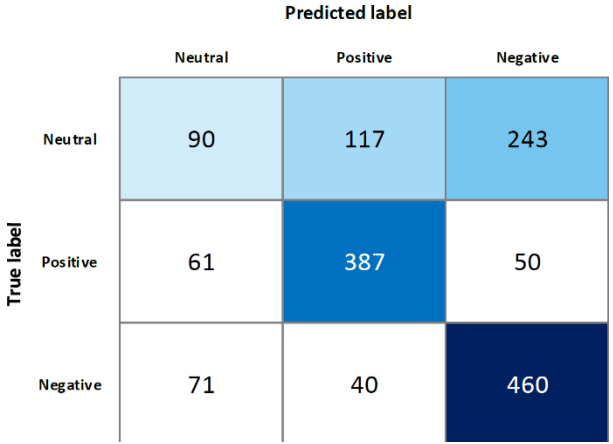
- In order to explore the correlation between tweet sentiments and their reply sentiments, we additionally create a ***"direct"* baseline classifier** by predicting the sentiment of the original tweet and assuming that its replies will have the same predominant sentiment.

  - ✓ We observe a highly significant **increase in F1 score by 15.4% points**!

Metric	Direct	**Proposed**
F1 score	56.5%	**71.9%**
Recall	54.0%	**79.1%**
Precision	60.0%	**66.1%**

Experiments: Error Analysis

- The confusion matrix shows that most of the wrong classification results have actually **neutral reply polarity**!
  - ✓ It is because, using the automatic label assignment strategy, we make our system **sensitive to positive/negative** sentiments in order to boost our performance of interest.
  - ✓ 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.



Experiments: Ensemble Model

- Following the state-of-the-art model of standard tweet sentiment polarity classification proposed by M. Cliche, we implement a **Convolutional** Neural Network-based model and **average** its predictions with those of our **BiLSTM** model.

- **The CNN architecture:**

  - ✓ 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.
  - ✓ 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.
  - ✓ The results are concatenated, leading to a **600-dimensional feature map** for every input, followed by a dropout.
  - ✓ Finally, a fully connected layer (followed by a SoftMax) is used for the classification.

* M. Cliche, “BB_twtr 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.

Experiments: Ensemble Model

- We train the CNN independently with **similar optimization and data pre-processing parameters** as for our BiLSTM model.

- The evaluation results of the ensemble model on the RETWEET test:

F1 score	Recall	Precision
73.2%	81.0%	66.8%

Experiments: Ensemble Model

- We train the CNN independently with **similar optimization and data pre-processing parameters** as for our BiLSTM model.

- The evaluation results of the ensemble model on the RETWEET test:

F1 score	Recall	Precision
73.2%	81.0%	66.8%

- Unexpectedly, we observe only a very slight improvement over the BiLSTM model!

  - ✓ 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**!

1. Overview
2. Standard Tweet Sentiment Polarity Classification
3. Methodology
4. RETWEET
5. Experiments and Results on RETWEET
6. **Conclusion**

Conclusion

- **A new challenge in the area of Twitter Sentiment Analysis**
  - ✓ Prediction of the overall polarity of first-order replies to an English source tweet.

- **RETWEET**
  - ✓ The first public dataset for sentiment prediction of first-order Twitter replies.

- **Future work**
  - ✓ To extend the method by taking some prior knowledge into consideration.
  - ✓ More task-specific data collection strategies will be explored to extend RETWEET.

Important Links

- The RETWEET dataset: https://kaggle.com/soroosharasteh/retweet
- This presentation file: https://github.com/starasteh/retweet
- The presentation video: https://www.youtube.com/channel/UCr8grdeS636T8Bk4c_rufCg
- The source code of the project: https://github.com/starasteh/retweet

Thank you for listening!