Twitter Sentimental Analysis: Comparing the performance of several algorithms and word embeddings

Yasaman Shahrasbi
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, Canada
yasaman.shahrasbi@uottawa.ca

***Abstract*—In this work, the goal is to present a sentimental analysis on a Twitter data set. The combination of several machine learning classifiers (support vector machine, multi-layer perceptron, XGBoost) and word embedding methods (token2vec, fastText, word2vec, GloVe, and BERT-based techniques) are compared based on recall and f1-score evaluation metrics. This project shows that the combination of BERT-based techniques with the classifiers outperforms the other combinations.**

***Index Terms*—Natural Language Processing, Sentimental Analysis**

I. Introduction

The growth of the Internet has brought facilities such as websites, social media, and blogs, where from individuals who own businesses to writers are allowed to offer and display their products. As the text information available online is increasing and more individuals write about their opinions, reviews, and ratings, it has become of utmost importance for businesses and owners of websites to gain information regarding these textual data. Sentimental analysis in Natural Language Processing is a set of techniques used to automatically extract or classify the sentiment or the true meaning that lies in text data. The primary purpose of performing sentimental analysis is to make beneficial use of the feedback provided by individuals and customers to improve the quality of content, marketing, and decision-making. [9]

Sentimental analysis in Twitter is one of the popular areas of research. Twitter is a platform where not only do the specialized business owners or service providers can express ideas or give information, but also users and consumers can express their opinions and convey their views regarding products and societal issues. It is needless to say that using sentimental analysis in Twitter can help companies compete with their competitors and help government officials know the public opinion and act opon. [7]

According to the importance of performing research in this area, in this project, an extensive study is carried out using several machine learning classifiers and word embedding models to find out which method is the most useful one to detect the sentiments lie within tweets.

The objectives of this work are to answer two main questions:

1) Which word embedding method and which algorithm perform the best?
2) Which one of the two proposed methods works the best in terms of recall and f1-score evaluation metrics?

II. Background and related work

A. Support Vector Machine

Support Vector Machine (SVM) is a classification technique where the primary purpose is to find a hyperplane that maximizes the margin between different classes. For non-linear decision boundaries, kernel trick method is used in order to find a hyperplane that maximizes the margin between classes in a higher dimension. [21]

B. Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is a distributed, boosting algorithm that in each step, a weak learner is trained on the data and the goal is to minimize the training error. This algorithm can be used for sparse data and since several tree algorithms are trained in a parallel manner, this approach is fast. [5]

C. Multi-layer Perceptron

Multi-layer Perceptron (MLP) is a feedforward neural network algorithm that uses backpropagation for training. The architecture of this model consists of at least three layers: input layer, hidden layer, and output layer, where the nodes in each layer are connected as a directed graph.

D. Word Embedding

Word embedding is a fixed-length representation of text data that is useful for text mining tasks such as chunking, sentiment analysis, and question-answering. There are two approaches for word embeddings: prediction-based and count-based models. Word2vec, fastText, GloVe, and token2vec are just a few word embedding toolkits to name. [2]

E. Text Classification

With the advent of new technologies, more and more online text data are available. Text mining is an area of research where the goal is to extract underlying patterns and knowledge from unstructured text data. [18] Text mining has several applications such as classification, summarization, and filtering documents. Text classification is an area of text mining which is the process of classifying unstructured text data existing in documents, social networks, websites, e-mails, etc, into predefined groups. [3]

F. Sentimental Analysis Challenges

There are several challenges in the sentimental analysis: The length of the texts, relevancy, type of communication, stop words, and multi-modal content. [3]

Length of texts: Bermingham et al.'s study shows that classifying tweets is easier than longer text documents. [4]

Relevancy: In Twitter sentimental analysis, as the length of tweets is shorter compared to other documents, in order to find the relevance of a tweet to a topic, the researchers suggest that looking for words or hashtags is a strong sign.

Type of communication: Most textual data available on web pages or documents are written in formal language. However, the sentimental analysis in tweets has some challenges due to the informal communication type, the uppercase form of words, and, when intentionally, some words are elongated.

Stop words: In Twitter sentimental analysis, as opposed to other documents, a different set of stop words should be defined and removed from the text. The reason is that some words such as "like," which is considered a stop word in other texts, can make a difference in Twitter sentimental analysis.

Multi-modal content: A difference between Twitter and other text documents is that tweets include other types of content such as videos and images. Performing sentimental analysis in tweets requires to be done on the text and these other types of information.

G. Literature Review

Sentimental Analysis in Twitter uses four different methods: machine learning, lexicon-based, hybrid, and graph-based.

- **Machine Learning**
  Several classifiers are used in the machine learning method, such as Naive Bayes, Logistic Regression, Random Forest, Support Vector Machine, and Maximum Entropy.
  Go et al. studied a binary classification of tweets into negative and positive classes. In this work, Naive Bayes, SVM, and Maximum Entropy algorithms were trained on a set of 1,600,000 tweets. Also, bigrams, unigrams, and POS tags were used for features. This study shows that using bigrams as features and Naive Bayes as the algorithm has the best performance- 82.7%. [8]
  In another study done by Pak et al., a multi-class classification problem of tweets into three positive, negative, and neutral classes, three classifiers -Support Vector Machine, Multinomial Naive Bayes, and Conditional Random Field- are compared based on the accuracy performance. [13]
  Da Silva et al. used an ensemble learning method where Support Vector Machine, Multinomial Naive Bayes, and Logistic Regression are combined. Bag of words and feature hashing are used as feature representation methods. [6] This experiment shows that the classification accuracy is improved as opposed to the work by Lin et al., where several Logistic Regressions are combined as the ensemble classifier. [12]
  Dong et al., employed an Adaptive Recursive Neural Network method for the sentiment analysis task. This method obtained 65.9% f1-score. [10] However, in the work proposed by Vo et al., using the same data set used in Dong et al.'s work, using a different word embedding method increased the f1-score by 4%. [20]
- **Lexicon-based**
  In lexicon-based algorithms, the score of negativity and positivity of tweets are calculated. One of the upsides of these methods, as opposed to machine learning-based algorithms, is that they do not require any models to be trained.
  SentiStrength, proposed by Thelwall et al., is a lexicon-based algorithm that is able to find sentiments in informal texts such as tweets. SentiStrength includes emoticons, negations, and boosting as well as a list of 700 most frequently used words and phrases in social media. In this paper, the SentiStrength method is compared to several machine learning-based approaches, and it is shown that this method is as promising as machine learning-based methods. [19]
  SentiCircles is a method proposed by Saif et al., in which the scores of negativity and positivity of words in tweets are updated in each step by considering the patterns extracted from words that occur in different contexts together. This method is compared on three datasets: OMD [16], HCR [17], and STS-Gold [14]. The results show that SentiCircles perform better than MPQA and SentiWordNet methods. [15]
- **Hybrid**
  In these methods, machine learning and lexicon-based approaches are combined.
  Khuc et al. use a lexicon-based method with the Online Logistic Regression as the classifier. For the lexicon-based classifier, using the MapReduce method, a matrix of bi-gram words occurring together is built, and the cosine similarity is calculated. This work shows that using Online Logistic Regression along with the lexicon-based classifier outperforms the case when only the lexicon-based method is used. [11]
- **Graph-based**
  Even though machine learning methods outperform other methods in Twitter sentimental analysis, they need a large amount of labeled text data. Therefore, semi-supervised methods have come to notice. Semi-supervised learning(SSL) methods are useful for cases where the number

of unlabeled instances is far more than the labeled ones. [7]

III. Dataset

This project uses Twitter data as part of the Twitter and Reddit sentimental analysis dataset. [1] This dataset, which contains tweets and opinions of officials and people for an election in India in 2019, was compiled using the Tweepy API as part of a multi-source social media sentiment analysis project. The Twitter dataset contains 162980 tweets, along with sentimental labels: -1, 0, and 1 that indicate negative, neutral, and positive sentiment, respectively. Fig. 1 illustrates the dataset.

In this project, 10000 of the tweets are used for the experiment, where 70% of the data is used for training and the rest is used for testing.

IV. Framework

The training data set consists of 6699 instances, and the test data set consists of 3300 data instances. Figures. 2 and 3 show the percentage of instances in each class, which shows that this problem is a multi-class classification problem with imbalanced distribution of data.

This work includes two experiments. [1]

- **Part 1** In the first part, the text data, which are tweets, is lemmatized using the SpaCy library in Python. Secondly, all the symbols and punctuations are removed using regular expressions. Thirdly, using the list of stopwords, all the stopwords such as a, be, can, day, etc, are removed from the data. Lastly, using four different word embeddings: Token2vec, fast text, word2vec, and glove, all the word embeddings of the remaining tokens are calculated. For each instance, the word embeddings for each datum are averaged into a single vector. Further, Support Vector Machine(SVM), three different multi-layer perceptron (MLP), and XGBoost algorithms are compared for this classification task.
  The default parameters are used in order to train the SVM model. The three MLP models are built as follows: The first model has one, the second one has three, and the third one has five hidden layers. All the three models have 100 hidden units and the number off epochs is 40. For the XGBoost model, 2000 decision trees are used as the base estimators, and the other parameters are set as default.
- **Part 2**
  In the second part, two BERT-based word embedding methods (small BERT and talking heads) are used for preprocessing the text data. Afterwards, for each data representation, three different Multi-layer perceptron (MLP) algorithms are used for the classification task. The first MLP model has one, the second model has 100, and the third model has 1000 hidden layers. The number of epochs is 40 for each of the models.

[1]https://tinyurl.com/4wznk4ss

SVM (recall)	Token2vec	Fast text	word2vec	glove
Neutral class	0.78	0.82	0.76	0.76
Positive class	0.65	0.66	0.62	0.63
Negative class	0.42	0.41	0.13	0.33
Weighted Average	0.63	0.65	0.54	0.60

TABLE I
SVM Recall Score

V. Experimental Evaluation

In this section the results are provided. To compare the several word embedding methods and the models, I compare the recall scores per class, weighted average of recall scores per class, and the weighted average of f1-scores. The recall evaluation metric (fig. 5) shows how successful a classifier is by looking at the percentage of correctly classified instances of a specific class. The precision metric indicates what percentage of instances classified as belonging to a specific class actually belong to it. F1-score (fig. 6) is the harmonic mean of recall and precision, which is a good metric to compare the performance of several classifiers. In order to compare the performance of the classifiers used in this experiment, the weighted average of recall scores and f1-score will be computed as the nature of the data set used in this work is imbalanced.

The reason for considering these evaluation measures in this setting is that this is an imbalanced, multiclass classification problem. It is crucial to see how well the classifiers perform in each class. Recall and f1-score metrics are useful metrics for this purpose.

VI. Results and Discussions

According to tables I through X, which correspond to the experiments done in the part1, among the four different word embedding methods that is used in this project, the combination of all the five algorithms (SVM, MLP1, MLP2, MLP3, and XGBoost) with fastText word embedding has the highest weighted recall and weighted f1-score metrics. Moreover, it is worth noting that almost in all the cases, the performance of using word2vec word embedding is also promissing.

On the other hand, according to tables XI through table XIV, according to recall weighted average and f1-score weighted average, the combinations of both BERT word embedding methods and the three MLP classifiers are the same. Also, according to the recall per class of using BERT models along with the three MLP classifiers, it is clear that both the techniques almost perform the same.

VII. Conclusion

Overall, to compare the results from part1 and part2 of this experiment, it is evident that the second part has higher results.

Therefore, the answers to the two objectives mentioned in the Introduction part are as follows:

1) In part1, the fastText and word2vec word embedding techniques along with all the classifiers outperform the combination of other word embeddings and classifiers.

#	Tweet	Label
0	when modi promised “minimum government maximum governance” expected him begin the difficult job reforming the state why does take years get justice state should and not business and should exit psus and temples	-1
1	talk all the nonsense and continue all the drama will vote for modi	0
2	what did just say vote for modi welcome bjp told you rahul the main campaigner for modi think modi should just relax	1
3	asking his supporters prefix chowkidar their names modi did great service now there confusion what read what not now crustal clear what will crass filthy nonsensical see how most abuses are coming from chowkidars	1
4	answer who among these the most powerful world leader today trump putin modi may	1
5	kiya tho refresh maarkefir comment karo	0

Fig. 1.

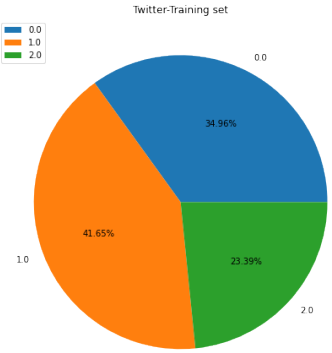

Fig. 2.

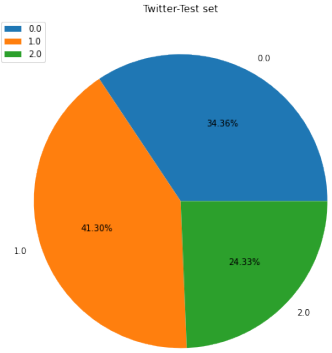

Fig. 3.

SVM (f1-score)	Token2vec	Fast text	word2vec	glove
Neutral class	0.69	0.71	0.64	0.67
Positive class	0.67	0.68	0.59	0.63
Negative class	0.50	0.50	0.20	0.42
Weighted Average	0.63	0.64	0.5	0.59

TABLE II
SVM F1-SCORE

MLP1 (Recall)	Token2vec	Fast text	word2vec	glove
Neutral class	0.69	0.74	0.72	0.66
Positive class	0.72	0.70	0.67	0.72
Negative class	0.52	0.52	0.11	0.39
Weighted Average	0.66	0.67	0.54	0.61

TABLE III
MLP1 RECALL SCORE

MLP1 (f1-score)	Token2vec	Fast text	word2vec	glove
Neutral class	0.70	0.72	0.65	0.66
Positive class	0.69	0.70	0.60	0.66
Negative class	0.55	0.55	0.17	0.46
Weighted Average	0.66	0.67	0.50	0.61

TABLE IV
MLP1 F1-SCORE

MLP2 (Recall)	Token2vec	Fast text	word2vec	glove
Neutral class	0.76	0.70	0.68	0.61
Positive class	0.73	0.64	0.70	0.72
Negative class	0.37	0.60	0.12	0.45
Weighted Average	0.65	0.65	0.55	0.61

TABLE V
MLP2 RECALL SCORE

MLP2 (f1-score)	Token2vec	Fast text	word2vec	glove
Neutral class	0.70	0.71	0.64	0.65
Positive class	0.69	0.67	0.61	0.66
Negative class	0.47	0.55	0.18	0.48
Weighted Average	0.64	0.65	0.51	0.61

TABLE VI
MLP2 F1-SCORE

MLP3 (Recall)	Token2vec	Fast text	word2vec	glove
Neutral class	0.65	0.74	0.64	0.65
Positive class	0.72	0.69	0.80	0.76
Negative class	0.52	0.51	0.00	0.30
Weighted Average	0.64	0.66	0.54	0.6

TABLE VII
MLP3 RECALL SCORE

MLP3 (f1-score)	Token2vec	Fast text	word2vec	glove
Neutral class	0.68	0.71	0.63	0.66
Positive class	0.68	0.68	0.62	0.66
Negative class	0.55	0.55	0.00	0.40
Weighted Average	0.64	0.65	0.46	0.60

TABLE VIII
MLP3 F1-SCORE

XGBoost (Recall)	Token2vec	Fast text	word2vec	glove
Neutral class	0.70	0.73	0.66	0.66
Positive class	0.72	0.74	0.63	0.68
Negative class	0.43	0.42	0.23	0.36
Weighted Average	0.64	0.66	0.54	0.59

TABLE IX
XGBOOST RECALL SCORE

XGBoost (f1-score)	Token2vec	Fast text	word2vec	glove
Neutral class	0.69	0.72	0.63	0.65
Positive class	0.68	0.70	0.58	0.63
Negative class	0.49	0.49	0.28	0.42
Weighted Average	0.63	0.65	0.55	0.58

TABLE X
XGBOOST F1-SCORE

Small BERT (recall)	MLP1	MLP2	MLP3
Neutral class	0.81	0.82	0.82
Positive class	0.87	0.85	0.84
Negative class	0.64	0.65	0.69
Weighted Average	0.79	0.79	0.80

TABLE XI
SMALL BERT RECALL SCORE

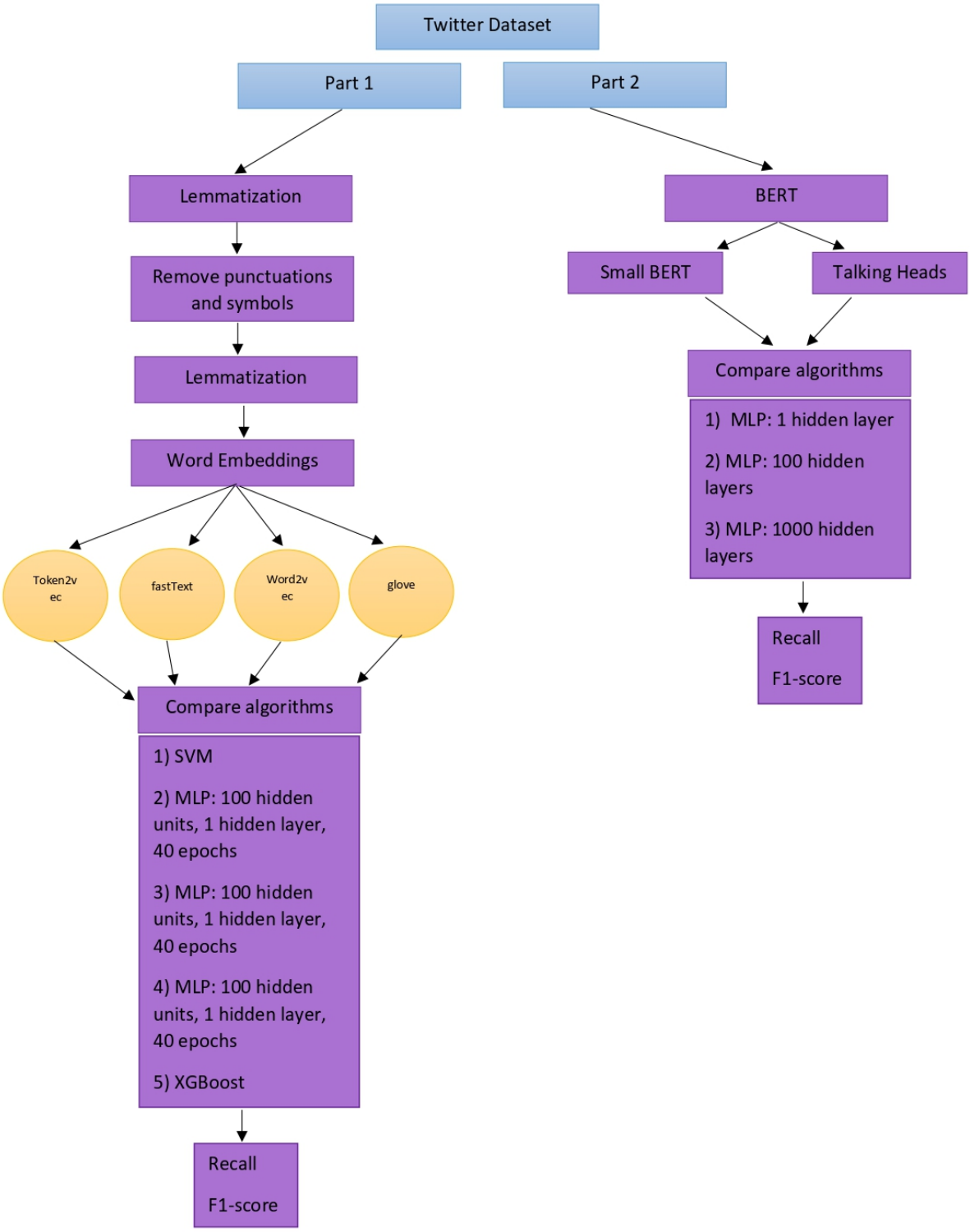

Fig. 4. Framework

In part2, the combinations of BERT word embedding methods with all the three Multi-layer Perceptron models almost perform the same.

2) All the experiments and combinations in part2 outperform the ones in part1.

VIII. Future Work

In future works, one can focus on using Deep Learning models such as Recurrent Neural Networks with word embedding models and try to improve the results presented in this project.

Another research direction is to experiment with other techniques, such as graph-based models, as the number of annotated data available is much less than the unlabeled data.

References

[1] Chaithanya Kumar A. Twitter and reddit sentimental analysis dataset, Nov 2019.

[2] Felipe Almeida and Geraldo Xexéo. Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*, 2019.

[3] Berna Altınel and Murat Can Ganiz. Semantic text classification: A survey of past and recent advances. *Information Processing & Management*, 54(6):1129–1153, 2018.

[4] Adam Bermingham and Alan F Smeaton. Classifying sentiment in microblogs: is brevity an advantage? In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1833–1836, 2010.

[5] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[6] Nadia FF Da Silva, Eduardo R Hruschka, and Estevam R Hruschka Jr. Tweet sentiment analysis with classifier ensembles. *Decision support systems*, 66:170–179, 2014.

[7] Anastasia Giachanou and Fabio Crestani. Like it or not: A survey of twitter sentiment analysis methods. *ACM Computing Surveys (CSUR)*, 49(2):1–41, 2016.

[8] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

[9] Doaa Mohey El-Din Mohamed Hussein. A survey on sentiment analysis challenges. *Journal of King Saud University-Engineering Sciences*, 30(4):330–338, 2018.

[10] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 151–160, 2011.

[11] Vinh Ngoc Khuc, Chaitanya Shivade, Rajiv Ramnath, and Jay Ramanathan. Towards building large-scale distributed systems for twitter sentiment analysis. In *Proceedings of the 27th annual ACM symposium on applied computing*, pages 459–464, 2012.

[12] Jimmy Lin and Alek Kolcz. Large-scale machine learning at twitter. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 793–804, 2012.

[13] Alexander Pak and Patrick Paroubek. Twitter based system: Using twitter for disambiguating sentiment ambiguous adjectives. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 436–439, 2010.

[14] Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold. 2013.

[15] Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. Contextual semantics for sentiment analysis of twitter. *Information Processing & Management*, 52(1):5–19, 2016.

[16] David A Shamma, Lyndon Kennedy, and Elizabeth F Churchill. Tweet the debates: understanding community annotation of uncollected sources. In *Proceedings of the first SIGMM workshop on Social media*, pages 3–10, 2009.

[17] Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 53–63, 2011.

[18] Ah-Hwee Tan et al. Text mining: The state of the art and the challenges. In *Proceedings of the pakdd 1999 workshop on knowledge disocovery from advanced databases*, volume 8, pages 65–70. Citeseer, 1999.

[19] 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.

[20] Duy-Tin Vo and Yue Zhang. Target-dependent twitter sentiment classification with rich automatic features. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[21] Mohammed J Zaki, Wagner Meira Jr, and Wagner Meira. *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Fig. 5.

$$f1score = 2 \times \frac{Precision\ \times Recall}{Precision + Recall}$$

Fig. 6.

Small BERT (f1-score)	MLP1	MLP2	MLP3
Neutral class	0.83	0.82	0.83
Positive class	0.83	0.83	0.83
Negative class	0.67	0.67	0.7
Weighted Average	0.79	0.79	0.80

TABLE XII
SMALL BERT F1-SCORE

Talking Heads BERT (recall)	MLP1	MLP2	MLP3
Neutral class	0.84	0.82	0.82
Positive class	0.85	0.85	0.83
Negative class	0.63	0.65	0.71
Weighted Average	0.79	0.79	0.79

TABLE XIII
TALKING HEADS RECALL SCORE

Talking Heads BERT (f1-score)	MLP1	MLP2	MLP3
Neutral class	0.83	0.83	0.83
Positive class	0.83	0.83	0.82
Negative class	0.67	0.69	0.69
Weighted Average	0.79	0.79	0.79

TABLE XIV
TALKING HEADS F1-SCORE