

Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data

Sani Kamsı
Yeditepe University
Istanbul, Turkey
sani.kamis@gmail.com

Dionysis Goularas
Yeditepe University
Istanbul, Turkey
goularas@cse.yeditepe.edu.tr

Abstract— This study presents a comparison of different deep learning methods used for sentiment analysis in Twitter data. In this domain, deep learning (DL) techniques, which contribute at the same time to the solution of a wide range of problems, gained popularity among researchers. Particularly, two categories of neural networks are utilized, convolutional neural networks (CNN), which are especially performant in the area of image processing and recurrent neural networks (RNN) which are applied with success in natural language processing (NLP) tasks. In this work we evaluate and compare ensembles and combinations of CNN and a category of RNN the long short-term memory (LSTM) networks. Additionally, we compare different word embedding systems such as the Word2Vec and the global vectors for word representation (GloVe) models. For the evaluation of those methods we used data provided by the international workshop on semantic evaluation (SemEval), which is one of the most popular international workshops on the area. Various tests and combinations are applied and best scoring values for each model are compared in terms of their performance. This study contributes to the field of sentiment analysis by analyzing the performances, advantages and limitations of the above methods with an evaluation procedure under a single testing framework with the same dataset and computing environment.

Keywords—sentiment analysis, deep learning, convolutional neural networks, LSTM, word embedding models, Twitter data.

I. INTRODUCTION

In recent years, thanks to the increase in the use of social media, sentiment analysis gained popularity among a wide range of people with different interests and motivations. As users all over the world have the possibility to express their opinion about different subjects related with politics, education, travel, culture, commercial products, or subjects of general interest, extracting knowledge from those data became a subject of great significance and importance. Besides information concerning users' visited sites, purchasing preferences etc., knowing their feelings as they are expressed by their messages in various platforms, turned out to be an important element for the estimation of people's opinion about a particular subject. A very common method is to classify the polarity of a text in terms of user's satisfaction, dissatisfaction or neutrality. The polarity can vary in terms of labeling or number of levels from positive to negative but in general it denotes the feelings of a text varying from a happy to an unhappy mode. The approaches used for sentiments analysis are numerous and are based on different methods of natural language processing and machine learning techniques for extracting adequate features and classifying text in appropriate polarity labels. Since some years, with the popularity that deep learning methods have gained, various

deep neural networks were utilized on the field with success. Particularly, the convolutional neural networks and LSTM networks proved to be performant for sentiment analysis tasks. Various studies showed their effectiveness alone or in combination between them. In the field of natural language processing, among the methods that are used for extracting features from words, Word2Vec and the global vectors for word representation (GloVe) are the most popular ones.

The accuracy achieved with the above techniques is high but still not satisfactory, thus making sentiment analysis an ongoing and open research subject. For this reason researchers try to develop new methods or improve the present ones. As the existing methods have a big variety in terms of network configuration, tuning, etc., a research upon the evaluation of the already used methods is still necessary in order to have a clear and precise idea about their limits and the challenges on sentiment analysis. This paper contributes to this area by evaluating the most popular deep learning methods and configurations based on an approved dataset about sentiment analysis which is created from Twitter data under a single testing framework.

The paper is divided as follows: section 2 presents the related work on this field. Section 3 demonstrates the methodology and the different neural network configurations that are applied. Section 4 shows the results, compares the different methods between them and discusses the findings. Finally, section 5 concludes the paper.

II. BACKGROUND

With the expansion and popularity of social media and various platforms allowing people to express their opinion upon different subjects, sentiment analysis and opinion mining became a subject that attracted the attention of researchers worldwide. In a work published in 2008, the authors described the various methods that were used until that day [1]. In the last years deep neural networks proved to be particularly performant in sentiment analysis tasks. Among them, convolutional neural networks [2],[3] and recurrent neural networks [4] were widely implemented because CNN respond very well to the dimensionality reduction problem and a category of RNN the LSTM networks [5] handle with success temporal or sequential data. In the pioneer works presented in [6] and [7] the authors demonstrated that CNN architectures can be utilized with success for sentence classification. Moreover it was demonstrated that CNN perform slightly better than traditional methods [8]. In [9] the efficiency of RNN was demonstrated as they outperformed the state of the art methods and in [10] an implementation of CNN and LSTM networks was presented, showing the significant advantages of using together these two neural networks. In parallel, the GRU networks, introduced in 2014 [11] can be

used effectively with similar results in the place of LSTM [12]. In a survey of deep learning methods in sentiment analysis [13], it can be seen that the word embedding is done mainly with two methods, Word2Vec [14] or GloVe [15].

Nowadays, Twitter is one of the most influencing social media platforms which serves as an information sharing medium in countries all over the world [16]. Therefore, extracting public opinion from tweets about various subjects, measuring the influence of different events or classifying sentiments became a subject of great interest. The early works for sentiment analysis were using different methods for extracting features based mainly on bi-grams, unigrams, POS specific polarity features and were utilizing machine learning classifiers like the Bayesian networks or support vector machines [17], [18], [19]. In the last years, in different places all over the world, various science competitions were organized in order to attract the interest of researchers. Among them, the international workshop on semantic evaluation is organizing competitions on this field for the last 13 years. Today, deep learning methods are dominant and the related studies attempt to gain high scores in the competition using mainly different combinations of neural networks and various configurations of word embedding features. Concerning the sentiment analysis in Twitter data, a number of studies was distinguished in terms of performance. In [20] the authors proposed two different CNN configurations using different word embeddings, Word2Vec and GloVe, respectively, where their results are combined in a random forest classifier. In another study the authors are utilizing embeddings trained on lexical, part-of-speech and sentiment embeddings which are initializing the input of a deep CNN architecture [21]. In [22] the authors proposed two configurations based on bidirectional LSTM networks. The word embedding is achieved with GloVe. Another study [23] proposed a combination of CNN and LSTM networks. The authors experimented with three different word embedding models, the Word2Vec, GloVe and FastText [24] and they reported that GloVe had a poor performance compared to the other models. Finally, a variation of CNN's the RCNN's [25] were used successfully in [26]. Despite the performances of the above studies, when trying to do a comparison between them it was relatively difficult to evaluate the role of a dataset, a network configuration or a specific setup and tuning. The motivation of this study came from this issue, aiming to create a single framework in order to compare these methods and clarify the advantages and limitations of each particular configuration.

III. METHODOLOGY.

In this section we present the dataset, the word embedding models with their configurations, and the different deep neural network configurations that are used in this study. In the following setups GRU networks and RCNN's are not included because they give similar results with LSTM networks and CNN's.

A. Dataset and Preprocessing

A corpus of different datasets was utilized based on three datasets used in SemEval competitions. More specifically, the SemEval2014 Task9-SubTask B full data, the SemEval2016 full data Task4 and the SemEval2017 development data were used forming a total of around 32.000 tweets. They consist of a body of 662.000 words with a vocabulary of around 10.000 words. The next step was to

process the tweets in order to increase the system's performance during training. For this reason, an extra pre-processing task was carried out aiming to remove and modify some characters. This task included the conversion of all letters to lowercase, the removal of some special characters and emoticons or the tagging of urls.

B. Word Embedding

The word embedding models used in this study were the Word2Vec, and GloVe. The Word2Vec model was utilized to create 25-dimensional word vectors based on the dataset described before. The configuration of Word2Vec, was done by using the CBOW model. Additionally, words that appeared less than five times were discarded. Finally, the maximum skip length between words was set to 5. GloVe was utilized with its pretrained word vectors. They are also 25-dimensional vectors and were created from 2 billion tweets, which constitutes a significantly larger training dataset than the dataset extracted from SemVal data. All vectors were normalized using the following equation:

$$v'_i = \frac{v_i - v_{min}}{v_{max} - v_{min}} \quad (1)$$

Where v'_i is the normalized i-value of a 25-dimensional vector, v_{min} the minimum value and v_{max} the maximum value of the vector.

1) Sentence vectors

The sentence vectors are created after concatenating the word vectors of a tweet in order to form a unique vector. After experiencing with various lengths we created sentences with a length of 40 words. As tweets vary in length, in case that a tweet has more words, the extra words were removed. When they were less than 40 the words of the tweet were repeated until the desired size was achieved. An alternative method is to use zero padding in order to fill the missing words in a sentence. In the approach followed in this work, zero padding was used only in case of words that were not present in the vocabulary.

2) Sentence Regions

A supplementary approach in word embedding is to divide the word vectors of a sentence in regions, in an effort to preserve information in a sentence and long-distance dependency across sentences during the prediction process [27]. The division is done with the punctuation marks existing on a sentence. In the current configuration each region is composed of 10 words and a sentence has eight regions. In case of missing words or regions, zero padding is applied. Figure 1 presents the structure of regions in a sentence.

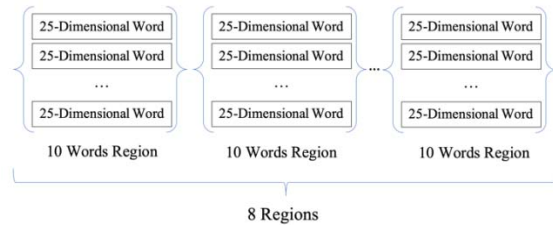


Fig. 1. Regional structure of a sentence. Every sentence has eight regions and every region has 10 25-dimensional words. In case of missing words or regions zero padding is applied in order to fill the missing regions.

In the end the dataset is converted two times forming two distinct datasets, one with non-regional and another with regional based sentences. In the first case the input size is 1000 (a sentence has 40 words where each of them has a size of 25) and in the second is 2000 (a sentence is divided into eight regions where each of them has 10 words of size 25)

C. Neural Networks

The neural network configurations that are proposed for the evaluation of twitter data are based on CNN and LSTM networks. Additionally, in one case a SVM classifier is used. All the networks were tested with both non-regional and regional datasets. In total, eight network configurations are proposed. As mentioned above, RCNN and GRU networks are not utilized because in our experiments they had very similar performance with CNN and LSTM networks correspondingly. All networks were trained with 300 epochs and used sigmoid activation function.

1) Single CNN network

In this network a single 1-dimensional CNN layer is used. Figure 2 presents this configuration where the sentence vector is convolved with 12 kernels with size 1×3 (from our tests it performed better when compared with other kernel configurations). The max pooling layer has a size of 1×3 . The CNN parameters will be the same for the following CNN configurations. Finally, a 3-dimensional output predicts the polarity in terms of positive, negative or neutral answer.

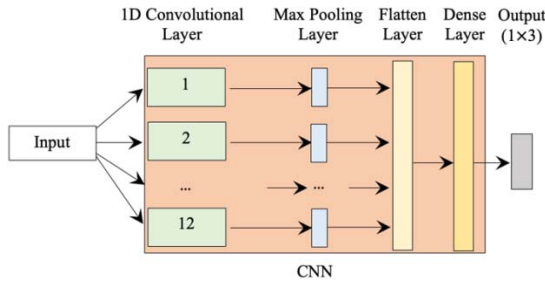


Fig. 2. CNN configuration with one layer and a 3-dimensional output for positive, neutral and negative polarity prediction.

2) Single LSTM network

In this configuration a single LSTM layer is used with a dropout of 20%. The output is again 1×3 in order to predict the polarity (positive, neutral or negative).

3) Individual CNN and LSTM networks

The aim of this configuration is to take the outputs of individual CNN and LSTM networks and evaluate together their results. A soft voting based on the outputs of the networks decides about the prediction answer. Figure 3 shows the structure of this configuration where the CNN and the LSTM networks have the same settings as in the two previous configurations (for CNN 12 kernels with size 1×3 and a max pooling layer with a size of 1×3).

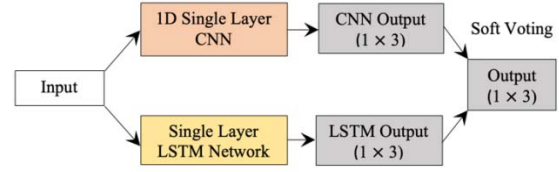


Fig. 3. Individual CNN and LSTM networks. The final prediction answer is given after soft voting calculated from the network outputs.

4) Single 3-Layer CNN and LSTM Networks

This setup utilizes a 3-layer 1-dimensional CNN and a single layer LSTM network. Figure 4 displays this configuration where the input is directed to a 3-layer CNN. The input has a size of 1000 if it is based on words (non-regional) or a size of 2000 if it is based on regions (regional).

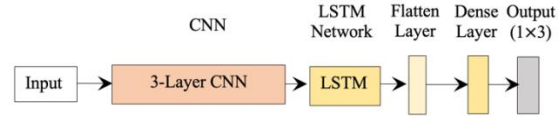


Fig. 4. Combination of a 3-Layer CNN and a LSTM network.

5) Multiple CNN's and LSTM Networks

In the current setup the input is divided into its basic elements, words for non-regional inputs and regions for regional inputs. Those elements serves as an input to individual CNN's. Then the output of every CNN is directed as an input to a single LSTM network. Figure 5 presents the network structure. In total according to the type of the input we have 40 or eight corresponding CNN's (40 words or eight regions). Every CNN network utilizes as previously 12 kernels.

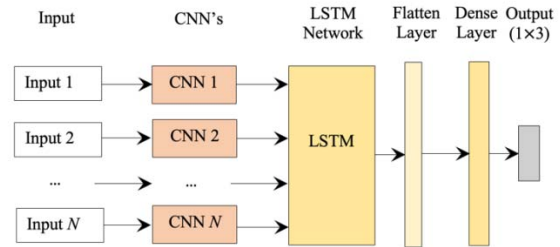


Fig. 5. Combination of CNN's and LSTM networks for an input that is divided into N inputs. N is equal to 40 (words) if the input is non-regional or 8 (regions) if the input is regional.

6) Single 3-Layer CNN and bidirectional LSTM Network

This setup includes a configuration same as (5) with the difference that this time a bidirectional LSTM network is used. The aim of this setup is to test the effectiveness of bidirectional LSTM networks compared to simple LSTM networks.

7) Multiple CNN's and bidirectional LSTM Network

Again this setup includes a configuration identical to (6) with the difference that this time a bidirectional LSTM network is used.

IV. RESULTS

This section presents the performance results of the previous network configurations in terms of Accuracy, Precision Recall, and F-measure (F1) as described in the following equations:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$F_Measure = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \quad (5)$$

In the above equations TP are the true positive, TN are the true negative, FP the false positive and FN the false negative predictions.

Table I and Table II presents the performance results of the proposed combinations using CNN and LSTM networks with Word2Vec and GloVe word embedding systems correspondingly. First, we can observe that utilizing the GloVe system increased the performance of almost all configurations (5%-7%). The reason behind it lies to the fact that with Word2Vec the vectorization of words has been made with a relatively small training dataset, around 32.000 tweets compared to the pretrained word vectors made with GloVe that used a significantly larger training dataset. The second observation is that using multiple CNN with LSTM networks instead of simple configurations increases the performance of the system, independently of the word embedding system (3%-6%). We can observe that the configurations (6) and (8) have almost always the best performance when compared with the other configurations. A third observation is that separating the text input into regions in most cases doesn't really improve the performance of a configuration (1%-2%). Concerning the use of SVM classifier instead of a soft-voting procedure it can be seen that it gives a slightly worse performance. A last observation is that the use of bidirectional LSTM networks instead of simple LSTM networks doesn't present any real advantage, which can be eventually explained by the nature of the data (the structure of words in a sentence).

Table III compares the best results of this study with the results of other works that used similar neural networks. We can observe that the current study has similar but slight inferior performance with the literature studies (6% difference). This is expected and it is due to the different datasets and specialized methods that are used in the other studies for shaping the dataset or tuning the network. Moreover, the scope of this study was not focused on achieving the best performance in comparison with other studies, but rather to evaluate and compare different deep neural networks and word embedding systems on a single framework. At this point, it is worth to mention that the best performance in the literature in terms of accuracy (~65%) is still not satisfactory, thus revealing that on sentiment analysis deep learning methods are still far from guaranteeing a performance comparable to other fields where the same networks are used with higher success rate (e.g. deep learning networks for object recognition in images).

TABLE I. Sentiment prediction of different combinations of CNN and LSTM networks with Word2Vec word embedding system with no-regional and regional settings from a set of around 32.000 tweets

Embedding Word System: Word2Vec					
Network Model	Type	Recall	Prec.	F1	Acc.
1. Single CNN network	N-R ^a	0.33	0.35	0.33	0.49
	R	0.32	0.34	0.33	0.51
2. Single LSTM network	N-R	0.43	0.51	0.39	0.51
	R	0.44	0.49	0.39	0.50
3. Individual CNN and LSTM Networks	N-R	0.43	0.47	0.37	0.50
	R	0.46	0.52	0.42	0.52
4. Individual CNN and LSTM Networks with SVM classifier	N-R	0.45	0.46	0.43	0.49
	R	0.42	0.54	0.38	0.51
5. Single 3-Layer CNN and LSTM Networks	N-R	0.41	0.52	0.40	0.46
	R	0.40	0.46	0.35	0.48
6. Multiple CNN's and LSTM Networks	N-R	0.43	0.47	0.37	0.50
	R	0.46	0.52	0.43	0.52
7. Single 3-Layer CNN and bi-LSTM Networks	N-R	0.42	0.45	0.39	0.48
	R	0.42	0.47	0.36	0.48
8. Multiple CNN's and bi-LSTM Networks	N-R	0.43	0.50	0.38	0.51
	R	0.46	0.51	0.44	0.52

a. N-R: Non-Regional, R: Regional word embedding

TABLE II. Sentiment prediction of different combinations of CNN and LSTM networks with GloVe word embedding system with no-regional and regional settings from a set of around 32.000 tweets.

Embedding Word System: GloVe					
Network Model	Type	Recall	Prec.	F1	Acc.
1. Single CNN network	N-R ^a	0.44	0.41	0.4	0.54
	R	0.35	0.31	0.31	0.48
2. Single LSTM network	N-R	0.5	0.58	0.48	0.55
	R	0.51	0.55	0.51	0.55
3. Individual CNN and LSTM Networks	N-R	0.53	0.6	0.53	0.58
	R	0.55	0.6	0.55	0.56
4. Individual CNN and LSTM Networks with SVM classifier	N-R	0.52	0.55	0.53	0.56
	R	0.49	0.6	0.5	0.56
5. Single 3-Layer CNN and LSTM Networks	N-R	0.5	0.5	0.5	0.52
	R	0.43	0.61	0.39	0.53
6. Multiple CNN's and LSTM Network	N-R	0.53	0.60	0.53	0.58
	R	0.55	0.6	0.56	0.59
7. Single 3-Layer CNN and bi-LSTM Network	N-R	0.52	0.59	0.53	0.57
	R	0.50	0.57	0.50	0.55
8. Multiple CNN's and bi-LSTM Network	N-R	0.54	0.60	0.55	0.59
	R	0.55	0.6	0.56	0.59

a. N-R: Non-Regional, R: Regional word embedding

TABLE III. Comparison of the state of the art methods with the best results of the current study.

Study	Network System	Word Embedding	Dataset (labeled Tweets)	Accuracy
Baziotis et al. [22]	bi-LSTM	GloVe	~50.000	0.65
Cliche [23]	CNN+LSTM	GloVe FastText Word2Vec	~50.000	0.65
Deriu et al. [20]	CNN	GloVe Word2Vec	~300.000	0.65
Rouvier and Favre [21]	CNN	Lexical, POS, Sentiment	~20.000	0.61
Wange et al. [27]	CNN+LSTM	Regional Word2Vec	~8.500	1.341 ^a
Current study	CNN+LSTM	Regional, GloVe	~31.000	0.59

^a RMSE

V. CONCLUSION

In this paper different configurations of deep learning methods based on CNN and LSTM networks are tested for sentiment analysis in Twitter data. This evaluation gave slight inferior but similar results with the state of the art methods, thus allowing to extract credible conclusions about the different setups. The relatively low performance of these systems showed the limitations of CNN and LSTM networks on the field. Concerning their configuration, it was observed that when CNN and LSTM networks are combined together they perform better than when used alone. This is due to the effective dimensionality reduction process of CNN's and the preservation of word dependencies when using LSTM networks. Moreover, using multiple CNN and LSTM networks increases the performance of the system. The difference in accuracy performance between different datasets demonstrates that, as expected, having an appropriate dataset is the key element for increasing the performance of such systems. Consequently, it looks like spending more time and effort in order to create good training sets presents more advantages rather than experimenting with different combinations or settings for CNN and LSTM networks configurations. To summarize, the contribution of this paper is that it allowed to evaluate different deep neural network configurations and experimented with two different word embedding systems under a single dataset and evaluation framework allowing to shed more light on their advantages and limitations.

REFERENCES

- [1] L. L. Bo Pang, «Opinion Mining and Sentiment Analysis Bo», *Found. Trends® Inf. Retr.*, vol. 1, no. 2, pp. 91–231, 2008.
- [2] K. Fukushima, «Neocognition: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position», *Biol. Cybern.*, vol. 202, no. 36, pp. 193–202, 1980.
- [3] Y. Lecun, P. Ha, L. Bottou, Y. Bengio, S. Drive, et al. R. B. Nj, «Object Recognition with Gradient-Based Learning», in *Shape, contour and grouping in computer vision*, Heidelberg: Springer, 1999, pp. 319–345.
- [4] D. E. Ruineihart, G. E. Hinton, et al. R. J. Williams, «Learning internal representations by error propagation», 1985.
- [5] S. Hochreiter et al. J. Urgan Schmidhuber, «Ltsm», *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [6] Y. Kim, «Convolutional Neural Networks for Sentence Classification», *arXiv Prepr. arXiv1408.5882*, 2014.
- [7] C. N. dos Santos et al. M. Gatti, «Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts», in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 69–78.
- [8] P. Ray et al. A. Chakrabarti, «A Mixed approach of Deep Learning method and Rule-Based method to improve Aspect Level Sentiment Analysis», *Appl. Comput. Informatics*, 2019.
- [9] S. Lai, L. Xu, K. Liu, et al. J. Zhao, «Recurrent Convolutional Neural Networks for Text Classification», *Twenty-ninth AAAI Conf. Artif. Intell.*, pp. 2267–2273, 2015.
- [10] D. Tang, B. Qin, et al. T. Liu, «Document Modeling with Gated Recurrent Neural Network for Sentiment Classification», in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.
- [11] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, et al. Y. Bengio, «Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation», *arXiv Prepr. arXiv1406.1078*, 2014.
- [12] J. Chung, C. Gulcehre, K. Cho, et al. Y. Bengio, «Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling», *arXiv Prepr. arXiv1412.3555*, 2014.
- [13] L. Zhang, S. Wang, et al. B. Liu, «Deep Learning for Sentiment Analysis: A Survey», *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, pp. e1253, 2018.
- [14] T. Mikolov, K. Chen, G. Corrado, et al. J. Dean, «Efficient Estimation of Word Representations in Vector Space», *arXiv Prepr. arXiv1301.3781*, 2013.
- [15] J. Pennington, R. Socher, et al. C. Manning, «Glove: Global Vectors for Word Representation», *Proc. 2014 Conf. Empir. Methods Nat. Lang. Process.*, pp. 1532–1543, 2014.
- [16] H. Kwak, C. Lee, H. Park, et al. S. Moon, «What is Twitter, a social network or a news media?», in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 591–600.
- [17] A. Go, R. Bhayani, et al. L. Huang, «Proceedings - Twitter Sentiment Classification using Distant Supervision (2009).pdf», *CS224N Proj. Report, Stanford*, vol. 1, no. 12, 2009.
- [18] A. Srivastava, V. Singh, et al. G. S. Drall, «Sentiment Analysis of Twitter Data», in *Proceedings of the Workshop on Language in Social Media*, 2011, pp. 30–38.
- [19] E. Kouloumpis, T. Wilson, et al. M. Johanna, «Twitter Sentiment Analysis: The Good the Bad and the OMG!», in *Fifth International AAAI conference on weblogs and social media*, 2011.
- [20] J. Deriu, M. Gonzenbach, F. Uzdilli, A. Lucchi, V. De Luca, et al. Jaggi, «SwissCheese at SemEval-2016 Task 4: Sentiment Classification Using an Ensemble of Convolutional Neural Networks with Distant Supervision», *Proc. 10th Int. Work. Semant. Eval.*, pp. 1124–1128, 2016.
- [21] M. Rouvier et al. B. Favre, «SENSEI-LIF at SemEval-2016 Task 4: Polarity embedding fusion for robust sentiment analysis», *Proc. 10th Int. Work. Semant. Eval.*, pp. 207–213, 2016.
- [22] C. Baziotis, N. Pelekis, et al. C. Doukeridis, «DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis», *Proc. 11th Int. Work. Semant. Eval.*, pp. 747–754, 2017.
- [23] M. Cliche, «BB twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs», *arXiv Prepr. arXiv1704.06125*.
- [24] P. Bojanowski, E. Grave, A. Joulin, et al. T. Mikolov, «Enriching Word Vectors with Subword Information», *Trans. Assoc. Comput. Linguist.*, Vol 5, pp. 135–146, 2016.
- [25] T. Lei, H. Joshi, R. Barzilay, T. Jaakkola, K. Tymoshenko, A. Moschitti, et al. L. Marquez, «Semi-supervised Question Retrieval with Gated Convolutions», *arXiv Prepr. arXiv1512.05726*, 2015.
- [26] Y. Yin, S. Yangqiu, et al. M. Zhang, «NNEMBs at SemEval-2017 Task 4: Neural Twitter Sentiment Classification: a Simple Ensemble Method with Different Embeddings», *Proc. 11th Int. Work. Semant. Eval.*, pp.

621–625, 2017.

- [27] J. Wang, L.-C. Yu, K. R. Lai, et al. X. Zhang. «Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model», *Proc. 54th Annu. Meet. Assoc. Comput. Linguist. (Volume 2 Short Pap., Vol 2*, pp. 225–230, 2016.