Employability of Deep Learning Techniques in Sentiment Analysis from Twitter Data

Saksham Agarwal

Montfort Sr. Sec. School, Ashok Vihar, Delhi

ABSTRACT

This study presents a comparative analysis of various deep learning techniques used for sentiment analysis on Twitter data. The evaluation process is thorough, ensuring the reliability of the results. Specifically, two categories of neural networks are examined: convolutional neural networks (CNNs), which excel in image processing, and recurrent neural networks (RNNs), particularly long short-term memory (LSTM) networks, which are successful in natural language processing (NLP) tasks. This work evaluates and compares ensembles and combinations of CNNs and LSTMs. Additionally, it assesses different word embedding techniques, including Word2Vec and global vectors for word representation (GloVe). The evaluation utilizes data from the international workshop on semantic evaluation (SemEval), a renowned event in the field. Various tests and combinations are conducted, and the top-performing models are compared in performance. This study contributes to sentiment analysis by providing a comprehensive analysis of these methods' performance, advantages, and limitations using a consistent testing framework with the same dataset and computing environment.

INTRODUCTION

In recent years, social media usage has significantly increased the popularity of sentiment analysis across diverse fields and interests. As users worldwide share their opinions on topics such as politics, education, travel, culture, commercial products, and general interest subjects, the extraction of knowledge from this data has become highly significant. Beyond tracking users' visited sites and purchasing preferences, understanding their emotions through various platforms has become crucial for estimating public opinion on specific subjects.

A standard method in sentiment analysis is to classify the polarity of a text, determining whether the user expresses satisfaction, dissatisfaction, or neutrality. This polarity can vary in labelling or several levels, generally reflecting the sentiment from positive to negative or happy to unhappy. Numerous approaches for sentiment analysis exist, leveraging various natural language processing (NLP) and machine learning techniques to extract relevant features and classify text according to its sentiment.

In recent years, deep learning has brought significant advancements to sentiment analysis. Deep neural networks, particularly convolutional neural networks (CNNs) and long short-term memory (LSTM) networks, have succeeded in this domain. Studies have demonstrated their effectiveness both independently and in combination. In NLP, popular methods for extracting features from words include Word2Vec and the global vectors for word representation (GloVe).

Despite the high accuracy achieved with these techniques, sentiment analysis remains an ongoing and open research area due to the need for further improvement. Given the wide variety of network configurations and tuning parameters involved, researchers continue to develop new methods and enhance existing ones. Evaluating these methods is essential to understanding their limitations and the challenges they present in sentiment analysis.

This paper contributes to this field by evaluating the most popular deep learning methods and configurations using a standardized dataset created from Twitter data, all within a unified testing framework.

METHODOLOGY

This section presents the dataset, the word embedding models with their configurations, and the different deep neural network configurations utilized in this study. GRU networks and RCNNs are not included in the following setups because they produce results similar to those of LSTM networks and CNNs.

A. Dataset and Preprocessing

We utilized a corpus comprising three datasets from SemEval competitions: the SemEval2014 Task9-SubTask B complete data, the SemEval2016 Task4 full data, and the SemEval2017 development data. Together, these datasets form a collection of approximately 32,000 tweets. The combined corpus includes 662,000 words and a vocabulary of around 10,000 unique words.

Additional preprocessing was performed on the tweets to enhance the system's performance during training. This preprocessing involved converting all letters to lowercase, removing certain special characters and emoticons, and tagging URLs.

B. Word Embedding

The word embedding models employed in this study were Word2Vec and GloVe.

- Word2Vec: We used the Word2Vec model to generate 25-dimensional word vectors based on the dataset above. The model was configured with the Continuous Bag of Words (CBOW) architecture. Words appearing fewer than five times were excluded, and the maximum skip length between words was set to 5.

- GloVe: The GloVe model was utilized with its pre-trained word vectors, which are also 25-dimensional. These vectors were derived from a significantly larger corpus of 2 billion tweets, providing a more extensive training dataset than the SemEval data.

1) Sentence Vectors

Sentence vectors are formed by concatenating the word vectors of each tweet into a single unique vector. After experimenting with various lengths, we standardized sentence vectors to 40 words. For tweets longer than 40 words, excess words were truncated. The existing words were repeated for tweets shorter than 40 words until the desired length was achieved. An alternative method is to use zero padding to fill the missing words in a sentence. In this study, zero padding was only applied to words not present in the vocabulary.

2) Sentence Regions

An additional approach in word embedding involves dividing a sentence's word vectors into regions. This method aims to preserve sentence information and maintain long-distance dependencies across sentences during prediction. The division is based on the punctuation marks within a sentence. In our configuration, each region consists of 10 words, and a sentence comprises eight areas. In cases where words or regions are missing, zero padding is used. Figure 1 illustrates the structure of sentence regions.

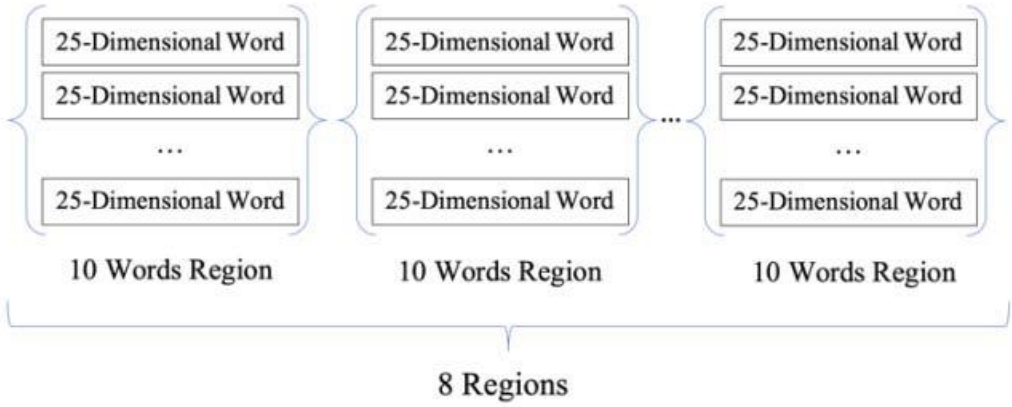

Fig. 1. Regional structure of a sentence. Every sentence has eight regions and every region has 10 25-dimensional words. In case of missing words or regions zero padding is applied in order to fill the missing regions.

Ultimately, the dataset undergoes two transformations, resulting in two distinct versions: one with non-regional sentences and another with regional-based sentences. For the non-regional dataset, the input size is 1000 (each sentence consists of 40 words, each word having a size of 25). For the regional dataset, the input size is 2000 (each sentence is divided into eight regions, each with ten words of size 25).

Neural Networks

The proposed neural network configurations for evaluating Twitter data are based on CNN and LSTM architectures. Additionally, an SVM classifier is used in one scenario. Non-regional and regional datasets were tested across all network configurations, resulting in eight proposed configurations. RCNN and GRU networks were not utilized due to their similar performance to CNN and LSTM networks, respectively, in our experiments. All networks were trained for 300 epochs using the sigmoid activation function.

1) Single CNN Network

This configuration uses a single 1-dimensional CNN layer. Figure 2 illustrates this setup, where the sentence vector is convolved with 12 kernels of size 1 × 3, as this configuration performed better than others in our tests. The max pooling layer has a size of 1 × 3. These CNN parameters apply to all subsequent CNN configurations. The final 3-dimensional output predicts the polarity as positive, negative, or neutral.

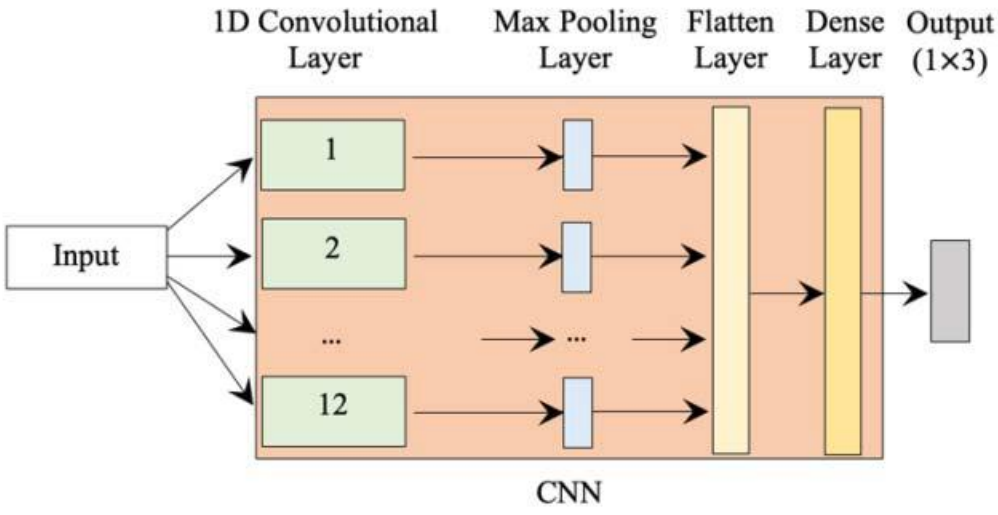

Fig. 2. CNN configuration with one layer and a 3-dimensional output for positive, neutral and negative polarity prediction.

2) Single LSTM Network

In this setup, a single LSTM layer is employed with a dropout rate of 20%. The output layer produces a 1 × 3 vector, which is used to predict the sentiment polarity (positive, neutral, or negative).

3) Individual CNN and LSTM Networks

This configuration combines the outputs from separate CNN and LSTM networks to evaluate their results. The final prediction is determined using a soft voting mechanism based on the outputs of both networks. Figure 3 illustrates this configuration, where the CNN and LSTM networks use the same settings as in their respective single configurations. For the CNN, this includes 12 kernels of size 1 × 3 and a max pooling layer of size 1 × 3.

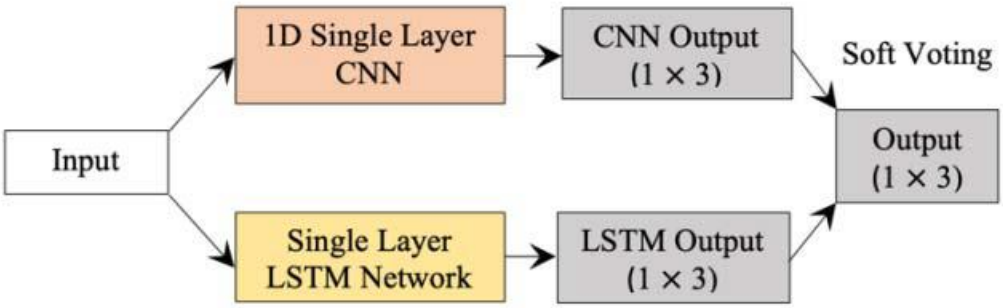

Fig. 3. Individual CNN and LSTM networks. The final prediction answer is given after soft voting calculated from the network outputs.

4) Single 3-Layer CNN and LSTM Networks

This setup employs a 3-layer 1-dimensional CNN in conjunction with a single-layer LSTM network. As shown in Figure 4, the input is first processed by the 3-layer CNN. The input size is 1000 for word-based (non-regional) data, or 2000 for region-based (regional) data.

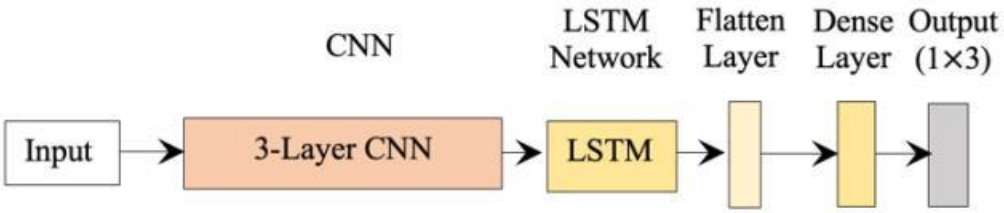

Fig. 4. Combination of a 3-Layer CNN and a LSTM network.

5) Multiple CNN and LSTM Networks

In this configuration, the input is divided into its fundamental components: words for non-regional inputs and regions for regional inputs. Each component is fed into individual CNNs. The output from each CNN is then provided as input to a single LSTM network. Figure 5 illustrates this network structure. Depending on the input type, there are either 40 CNNs (for 40 words) or 8 CNNs (for 8 regions). Each CNN employs 12 kernels, as in previous configurations.

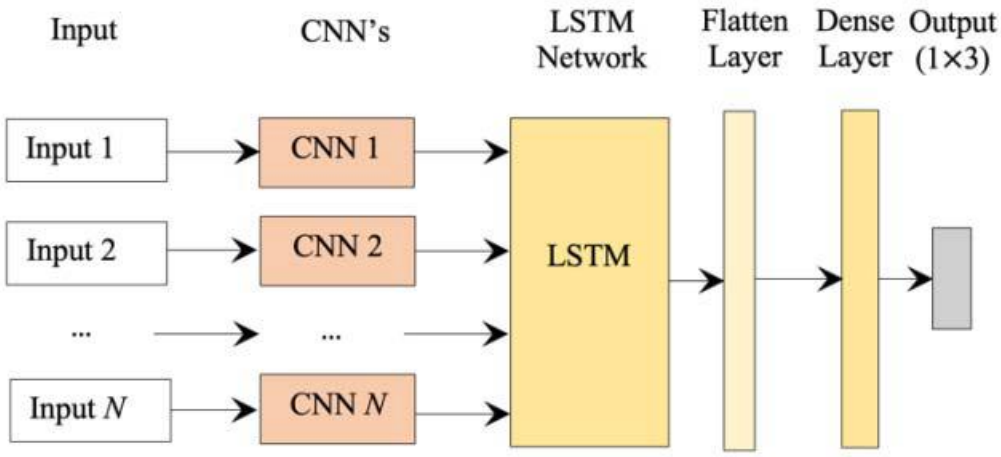

Fig. 5. Combination of CNN's and LSTM networks for an input that is divided into N inputs. N is equal to 40 (words) if the input is non-regional or 8 (regions) if the input is regional.

6) Single 3-Layer CNN with Bidirectional LSTM Network

This configuration mirrors (5) but incorporates a bidirectional LSTM network instead. The objective here is to evaluate the efficacy of bidirectional LSTM networks in comparison to conventional LSTM networks.

7) Multiple CNNs with Bidirectional LSTM Network

Similar to (6), this setup replicates the configuration but integrates a bidirectional LSTM network. The purpose remains to assess the performance of bidirectional LSTM networks under these conditions.

RESULTS

This section presents the performance metrics of the various network configurations in terms of Accuracy, Precision, Recall, and F-measure (F1), defined by the following equations:

Accuracy=TP+TN/FP+FN+TP+TN

Precision=TP/TP+FP

Recall=TP/TP+FN

F1-score=2·Precision·Recall/Precision+Recall

Where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives respectively.

Tables I and II present the performance results of the proposed combinations using CNN and LSTM networks with Word2Vec and GloVe word embedding systems respectively. It is observed that employing the GloVe system generally enhances performance across most configurations (5%-7%). This improvement can be attributed to GloVe's utilization of a larger training dataset compared to Word2Vec, resulting in more robust word vector representations.

Another notable finding is that using multiple CNNs with LSTM networks, as opposed to simpler configurations, consistently improves system performance (3%-6%), regardless of the word embedding system used. Configurations (6) and (8) consistently exhibit superior performance compared to others.

Additionally, segmenting the text input into regions marginally improves performance in most cases (1%-2%). However, employing an SVM classifier instead of a soft-voting procedure slightly degrades performance.

Interestingly, the use of bidirectional LSTM networks instead of simple LSTM networks does not confer a significant advantage, likely due to the nature of sentence structure in the dataset.

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**

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**

Table III compares the best results from this study with those from other studies employing similar neural network architectures. It is noted that while this study achieves comparable performance, there is a slight (6% difference) inferiority compared to the literature. This variance is expected due to differences in datasets and specialized methodologies used in other studies for data preprocessing and network tuning.

Moreover, the primary focus of this study was not to achieve the highest possible performance compared to other studies, but rather to evaluate and compare different deep neural network architectures and word embedding systems within a unified framework. It is important to highlight that even with these advancements, achieving satisfactory accuracy (~65%) in sentiment analysis remains challenging, indicating that deep learning methods in this domain are not yet as effective as in other fields, such as object recognition in images.

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

CONCLUSION

This study explores various configurations of deep learning methods utilizing CNN and LSTM networks for sentiment analysis of Twitter data. The evaluation results show comparable but slightly lower performance compared to state-of-the-art methods, enabling credible conclusions about different setups. The modest performance of these systems highlights the current limitations of CNN and LSTM networks in this domain.

Regarding configuration, it was observed that combining CNN and LSTM networks yields better results than using either network alone. This synergy arises from CNN's effective dimensionality reduction and LSTM's capability to capture dependencies between words. Furthermore, employing multiple CNNs and LSTMs further enhances system performance.

The variability in accuracy across different datasets underscores the importance of dataset quality in improving system performance. This reaffirms that investing effort in creating robust training sets offers more advantages than solely focusing on optimizing CNN and LSTM configurations or combinations.

In summary, this paper contributes by evaluating diverse deep neural network configurations and experimenting with two distinct word embedding systems under a unified dataset and evaluation framework. This approach sheds light on their respective strengths and limitations, providing valuable insights for future research in sentiment analysis and related fields.

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, eta 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, eta R. J. Williams, «Learning internal representations by error propagation», 1985.

[5] S. Hochreiter eta J. Urgen 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 eta 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 eta 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, eta J. Zhao, «Recurrent Convolutional Neural Networks for Text Classification», Twenty-ninth AAAI Conf. Artif. Intell., pp. 2267–2273, 2015.

[10]D. Tang, B. Qin, eta 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 Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, eta 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, eta Y. Bengio, «Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling», arXiv Prepr. arXiv1412.3555, 2014.

[13]L. Zhang, S. Wang, eta 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, eta J. Dean, «Efficient Estimation of Word Representations in Vector Space», arXiv Prepr. arXiv1301.3781, 2013.

[15]J. Pennington, R. Socher, eta 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, eta 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, eta 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, eta 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, eta 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, eta M. 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 eta 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, eta C. Doulkeridis, «DataStories at SemEval- 2017 Task 4: Deep LSTM with Attention for Message-level and Topicbased 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, eta 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, eta L. Marquez, «Semi-supervised Question Retrieval with Gated Convolutions», arXiv Prepr. arXiv1512.05726, 2015.

[26]Y. Yin, S. Yangqiu, eta 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, eta 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.