Comparison of Text Classification Methods using Deep Learning Neural Networks

Maaz Amjad
Dept. of Intelligent Information Systems and Technology,
Moscow Institute of Physics and Technology, Russia
Center for Computing Research (CIC)
Instituto Politécnico Nacional (IPN), Mexico
maazamjad@phystech.edu

Ilia Voronkov
Dept. of Intelligent Information Systems and Technology,
Moscow Institute of Physics and Technology, Russia
iliav_scn@mail.ru

Alexander Gelbukh
Center for Computing Research (CIC)
Instituto Politécnico Nacional (IPN), Mexico
gelbukh@gelbukh.com

Anna Saenko
Dept. of Intelligent Information Systems and Technology,
Moscow Institute of Physics and Technology, Russia
anna.saenko@phystech.edu

Abstract — In this article, we tend to examine the text classification task by using various neural networks. A small number of previously classified texts can change the accuracy of the studied text classifiers. In many text classification applications, this is often vital because an oversized range of uncategorized data is effortlessly reachable. However, getting an annotated text is a quite challenging task. The article additionally demonstrates that the Convolution Neural Network (CNN) does not demand the semantic or syntactic knowledge and can perform in a better way on words level. Secondly, a Recurrent Neural Network (RNN) model can effectively classify the text data (sequence type). RNN outperforms the other Neural Networks for the sequence test classification task. We used corpora of two different types from separate sources (IMDB and self-created bloggers corpus). The results of our experimentas now provide evidence that vector representation of the text can improve the score of the task.

I. Introduction

Research on NLP has a long tradition. lately, text categorization has been an eye-catching topic in modern research field era and has numerous emerging applications in marking, politics, academia and classification of the personality traits. Therefore, text classification got attention of many researchers.

There are growing appeals for text classification tasks. Feature representations play a substantial role in solving text classification tasks — traditional language models, like bag-of-words model where uni-gram, bi-gram, and in generic case n-grams are used for feature extraction from text. Researchers have developed several rigorous feature extraction methods like Latent Semantic Analysis (LDA) [2], Probabilistic Latent Semantic Analysis (PLSA) [4], frequency and MI [5] to catch more powerful features. Even though a lot of scientists presented some complex features like tree kernel [6] to obtain related information and correct word order from text, however, there are some challenging problems, for example, data sparseness that has the substantial effect on the classification score. In recent years, Deep Learning (DL) neural networks have been using to solve many natural language processing (NLP) problems. DL neural networks and word embedding have helped to address many crucial challenges in solving NLP problems. Word embedding is a scattered attribute learning over sequences of words and massively mitigate the data sparsity problem. Some of the researchers [3, 7] demonstrated that it could be better to use trained word embeddings to extract useful semantic and syntactic regularities. Additionally, it is significant to mention that, with the help of word embeddings, to obtain the semantic representation of text, the composition-based methods can be used.

Recursive Neural Networks (Recursive NN) have been using to construct sentence illustration [8, 9, 10] and presented good performance. Recursive NNs use tree structure technique to extract the semantics of a sentence. implementation of the textual tree development heavily effects on the performance of Recursive NN. However, the time-complexity of constructing textual tree is at least $O(n^2)$- n represents the length text. This approach would be extremely time-killing in case of a long sentence or document. On the other hand, Recursive NN may face some difficulty to develop a relationship between two sentences. Therefore, Recursive NNs are dissatisfactory in molding of large sentences in length or documents.

To address the issues of the Recursive Neural Networks, we advise a Recurrent Convolutional Neural Network (RCNN). We tend to use a recurrent structure (bi-directional)

to induce excellent results for learning word representation. This word representation approach has the power to extract appropriate contextual information because it introduces fewer noise as compared to windows-based neural network. Additionally, after word representations, the model will have the power to understand an even large range of words. In the secondly place, in order to determine which feature is significant to capture the critical segment throughout the classification task, we used maximum pooling layer. By merging the max-pooling layer and recurrent structure, our model takes the advantage of each CNN and RNN models with O(n) time complexity- n represents the length of the text. Ultimately, we compared proposed technique with state-of-the-art methods by using three different types of tasks in English language. Our model surpassed the state-of-the-art methods on different corpus.

The main contributions of this research are follows:

– experiments for text classification using Non-Neural Network(supervised learning)

– experiments for text classification using Neural Networks

– comparison for both supervised learning and deep learning for the text analysis task.

The rest of the article is structured as follows. Section 2 provides a review of state-of-the-art. Section 3 presents the methodology of IMDB and gender classification of the author of the text. Section 4 describes a baseline approach for text classification and the obtained results. Section 5 concludes the results and conclusion.

II. Related Work

In recent years, DL plays a substantial role in solving many significant NLP problems. To date, some successful methods involve DL neural networks. These neural networks involve Recurrent Neural Network (Recurrent NN), Convolutional Neural Network (CNN), Multi-Layer Neural Networks, etc. Each of these neural networks has specific pros and cons and apply to address different NLP tasks.

Recurrent NN is a type of neural networks. RNN is used to investigate a text at a word level. [11] during a rigid-sized hidden layer RNN can store the semantics of all the previous text. Furthermore, this model is useful in capturing the semantics of a long text; however, the model is biased. The reason for biases is that this model pays more attention to words that come subsequently. It means model particularly looks out for the words that come earlier. However, in a document, all words are carrying the equal probability. Consequently, this factor may reduce the efficiency to capture the semantics of a document.

To unravel the biases issue that was found in RNN, another model CNN was introduced. CNNs primarily were considered for image processing and recognition of videos; however, There has been numerous studies to investigate that CNN might also be used to handle the NLP problems [12]. Additionally, as compared to recurrent NN, this model can retrieve the semantic of texts in a very systematic manner with time complexity. It also has the power to extract important dialogues in a text with the help of a max-pooling layer. Max-pooling layer is a small window search that looks for to obtain the best features using different window size. [13, 14] The technique is called “kernel as a rigid window”.

Additionally, a series of recent studies has indicated that it is difficult to seek the most appropriate size of a window, because the kernel size cannot be specified to address different problems. In both cases, the size of the kernel causes particular issues. A small kernel size could lead to inaccurate results by missing discriminative information. On the other hand, with a large kernel size, it would be time-taking to train the model due to many parameter.

To address the classification problem, normally there are three main parts to deal with; how to propose techniques to get useful attributes, by using designed techniques, how to extract distinct attributes, and how to design the algorithms for Machine Learning (ML). To obtain features, traditional language model, for instance, Bag-of-words (Dictionary base model) is used for feature selection. Additionally, for complex features selection, some authors have driven the further development of some others features selection models, for example [18] part-of-speech tags and tree kernels, [15] noun-phrases. To achieve the better performance of the classification task, the removal of noisy features from the text is extremely important. One of the most primary used strategies is to eliminate stop words (e.g., "the," "a," "an"). More importantly, some authors have also suggested that some other modern techniques, for example information retrieval, [5] mutual-information and [17] L1 regularization are being used to obtain powerful features. The majority of prior research has applied different types of ML algorithms in training like, Support Vector Machine (SVM), Naïve Bayes (NB) and Logistic Regression (LR) etc. However, data sparsity related issues can be raised using these techniques.

We can reduce data sparsity issues by using DL especially Deep Neural Networks and word representation learning [19, 20]. [3, 7, 13, 16, 21, 22] for word representation, different types of neural networks models have been proposed. In text classification, word representation (word embedding) is the neural illustration of a word, that is a real-valued vector. The word-embedding technique is generally used to check how different words are relevant. Certain algorithms, such as word2vec, Glove are used to vectorize the words (embedding vectors) to find the words connectedness. If the vectors of two

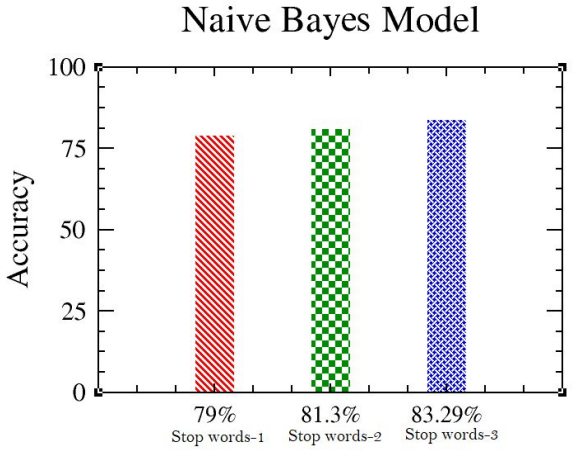
words will have less distance, then these words have a similar meaning in high dimensions.

III. Experimental Evaluation and Analysis

A binary classification problem (positive or negative reviews) is addressed using IMDB dataset. This dataset contains 25,000 documents (12,500 positive reviews, 12,500 negative reviews) in training and 25,000 (12,500 positive reviews, 12,500 negative reviews) document in testing part. We used different types of neural networks and then compare with the non-neural network approach.

Non-Neural Network Approach

For the non-neural network approach, we considered Naive Bayes algorithm using three different stop words lists. The first list of stop words was collected from NLT, second from Google and the third list was the combination of both Google and NLTK stop words. We achieved different accuracies with different stop words. We achieved (83.29%) best results using a combination of Google and NLTK stop words list.



Neural Networks Approach

For the neural network approach, initially, we used a Multi-Layer Perceptron (MLP), which has three layers (input layer, a single hidden layer, and output layer). We insert pre-trained word embedding's of IMDB dataset as input to the Multi-Layer Perceptron. In experiments, 6000 most frequent words are used to build a dictionary. A vector whose dimension is thirty-two represents each word of the movie review. It is incredibly significant to have the same length as the text. To do so, we limited the range of each movie review at maximum 500 words and removed longer movie review from the dataset. To obtain the same length for shorter movie reviews, we padded these movie reviews with zero values.

Finally, Multi-Layer Perceptron (MLP) creating a word-embedding input layer (first layer). The input layer has a length of five hundred and a thirty two-dimensional vector is used to denote each word of the movie review. Therefore, the matrix size of the input layer would be (thirty-two x five hundred). The inputs of the first layer leveled to one dimension subsequently used this one-dense hidden-layer for activation function (rectifier) which contains two hundred fifty units. The last segment (output layer) comprises a single neuron. Sigmoid activation is used to get the output values between 0 and 1 - positive or negative in our binary-classification task. In training of our model, we considered the batch size of 128. We used Adam algorithm, due to its ability to control the learning rate and possibility of best solutions [24]. Adam uses parameter's moving averages and which allows the algorithm to take into account the large effective step size, even without fine-tuning the algorithm can reach to this step size. After training, we evaluated our trained model and MLP achieved 92.16% accuracy. We can get even better accuracy if we increase hidden-layers with substantial embedding.

Secondly, we used CNN for text classification. CNN is useful to highlight the position of learned objects in the photographs. CNN models are substantially valuable because these models are capable of learning to identify objects in images. Consequently, the same idea can be applied to one-dimensional sequences of tokens in a review to discover the hidden pattern in a text to get the specific position of the features.

In CNN model architecture, the input of the first layer was word Embedding of the movie reviews. A one-dimensional convolutional layer; which has 32 feature maps with kernel size (window size) three is inserted as the second layer. Subsequently, there is a maximum pooling layer (one-dimensional) with a length and stride of 2. This objective of maximum pooling layer is to compresses the features representation of the convolutional layer by halving it. The remaining architecture of the CNN looks precisely the same as the MLP. In the computational point of view, memory units in CNN models share weights, which requires less time to train the CNN model. Additionally, it is important to mention that CNN's perform well to get the spatial relationships between words. Finally, CNN model got 93.75% accuracy.

Thirdly, we used Long Short-Term Memory (LSTM), recurrent NN models, to perform sequence-classification task on movie review corpus. LSTM models have been used in addressing the sequence classicization problems. The primary goal of applying LSTM is to get the actual class (in our case, either a positive or negative class) provided the sequence of words. As the length of the movie review (series of words) can vary, and some sequences of words might have more words

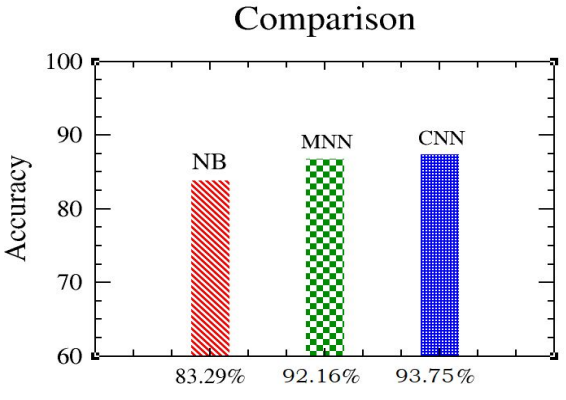
than others – the range of the words can be different. It implies that every review holds distinct dictionary size (the number of words) and the task is to classify the sentiment of each movie review, this is why it cannot be considered as a trivial case. To learn the dependencies between patterns (symbols) in the input sentences or long-term context, sequences of words may require to be supplied to NN models. In the LSTM model, the first layer is the input layer where the word embeddings are fed into it. The second layer is the LSTM layer that includes 100-smart neurons (memory-units). Here, we also used the sigmoid activation function to activate a single neuron (a dense output of the LSTM layer) concerning our 2-classes classification problem. 64 batch size is used in model training, and we got 95.07% accuracy by using LSTM.

Overfitting is a challenging issue in Recurrent NN, such as LSTM. Researchers have proposed a powerful technique known as a dropout to overcome this (overfitting) crucial point; Dropout layers can be added between the different layers of the neural network. In various experiments, these layers are inserted in between LSTM - dense output layers and embedding - LSTM. In the initial investigation, the model obtains the accuracy of 93.05% which is marginally less than simple LSTM case. Additionally, an alternative method is used to analyze the scores; we inserted a dropout layer between the input layer - LSTM layer (memory units' recurrent connections). Subsequently, during the next experimentation process, the trained model obtained 93.71% score. We can say, LSTM-specific dropout has less influence on the layer-wise dropout and more pronounced influence on neural networks' convergence.

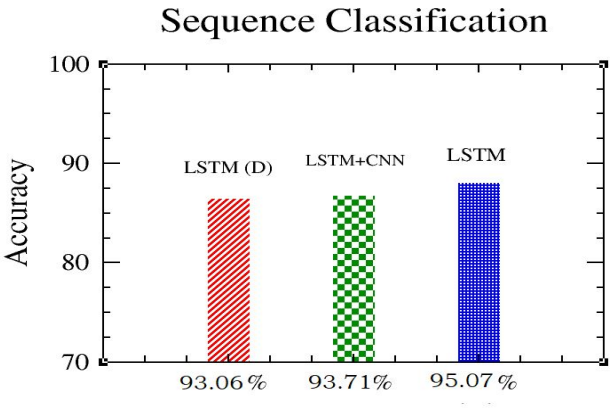
CNN are well-performed neural networks at capturing the hidden spatial pattern in the text. However, LSTM needs to do a more computational task to get similar results. CNN can choose invariant attributes to classify Pos. and Neg. sentiment. Furthermore, this learned spatial attributes might then be learned by LSTM layer as patterns. After the Embedding layer, it is an easy task to amend the neural network by adding a 1-dimensional max-pooling layers and CNN. Then, this Embedding layer provides the combined features to the LSTM. A small set consists of 32 features is used with size 3 of a kernel. In the further part, the max pooling layer to half the size of the attribute matrix. Finally, with less parameter tuning and faster training time, the model achieved 93.06% accuracy. We can say that, for Sequence Classification with Dropout, the model generated the similar scores to LSTM.

A. Analysis of the Experiments

- This research aims to develop more sophisticated methods for the classification task. With this aim in mind, in this paper, we present a comparison of text classification methods using Deep Learning Neural Networks and non-Deep Learning Neural Networks using two different datasets. For our first goal, we focus on IMDB movie review dataset to classify the sentiments of the movie reviews. For our second goal, we focus on self-created bloggers corpus dataset to classify the gender of the author of the text. Here we investigate the results of the different method separately, and we have verified that using different ways produces different results. We used tenfold cross-validation which calculates average classification score, described as the no. of correctly predicted classes. Additontionaly, we also make a comparison of our results to the majority Baseline, which assigned the title to each document of the largest class.
- Most experiments have been carried out with deep learning. To compare deep learning models with traditional machine learning models, we applied Naïve Bayes classifier for the movie review dataset (IMDB). This study used different stop words chosen from various origins. These stop words contain different N-grams; uni-grams, combinations of 1-2 grams, combinations of 1-3 grams as features to classify the sentiment of the movie reviews. We obtain good results using a combination of Google and NLTK stop words list although we can't pinpoint the exact reason for getting the best results.
- Contrary to the findings of preliminary experiments (Naïve Bayes classifier), we used specific neural networks, such as Multilayer Neural Perceptron (MLP) and Convolutional Neural Network (CNN). For this study, IMDB dataset (50% for training and 50% for testing) was used to perform experiments. After two epochs, as the results of the tests, models achieved an accuracy of 92.16% and 93.75% respectively. However, even better results can be obtained after several epochs. Consequently, in contrast with traditional machine learning models, our results cast a new light on the fact that neural networks are good predicting models. The results lead to a similar conclusion where it can be concluded that Convolutional Neural Network (CNN) outperform the Multilayer Neural Perceptron (MLP) due to its architecture which can emphasize the hidden text patterns in the training phase of the neural network.



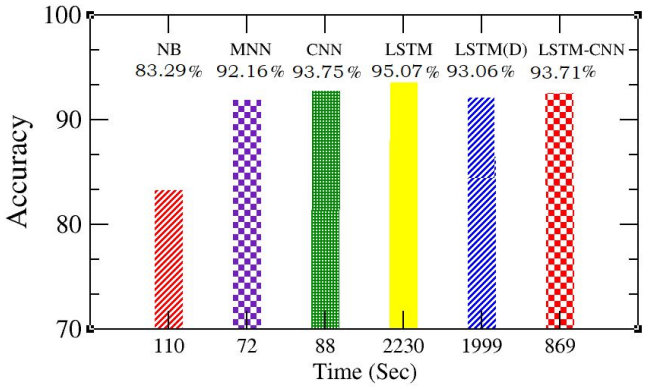
- The LSTM NN was used to experimentally investigate the IMDB corpus for the sequence classification task. This LSTM neural network contained a layer of 100 memory units (known as smart neurons). For this study, movie review dataset (50% for training and 50% for testing) was used to perform experiments. After three epochs, as the results of the tests, LSTM model yields increasingly good results with an accuracy of 95.07%. The main focus of the investigations was to compare the performance of CNN and LSTM to address the sequence classification problem using same corpus distribution (50% for training and 50% for testing). However, LSTM delivers significantly better results due to its architecture which contains memory cells that can memorize and forget text patterns.



By using IMDB movie review dataset, we investigate and compare the results obtained by using different methods. The result of this analysis reveals a promising fact that for the classification task, CNN outperforms other neural networks. On the other hand, a further novel finding is that for sequence classification, the LSTM neural network leads to substantially better results. Extensive results carried out for sequence classification are shown in the diagram.

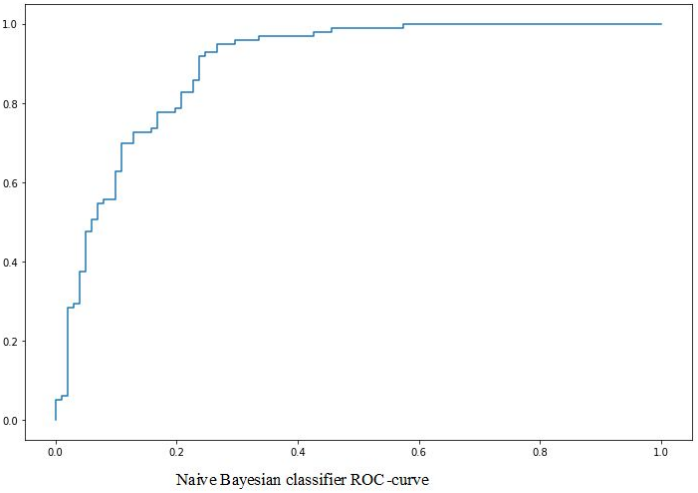
B. Comparison Tables

Here we compare the results of the Deep Learning Neural Networks and non-Deep Learning Neural Networks using IMDB dataset are shown in the diagram.. It is worth discussing these exciting facts revealed by the results of Convolutional Neural Networks(CNN), for sentimental analysis task; CNN neural network provides significantly better results compared with traditional machine learning algorithms. Moreover, the results lead to a similar conclusion that CNN depicted a high learning rate for corpora (IMDB and self-created bloggers corpus).



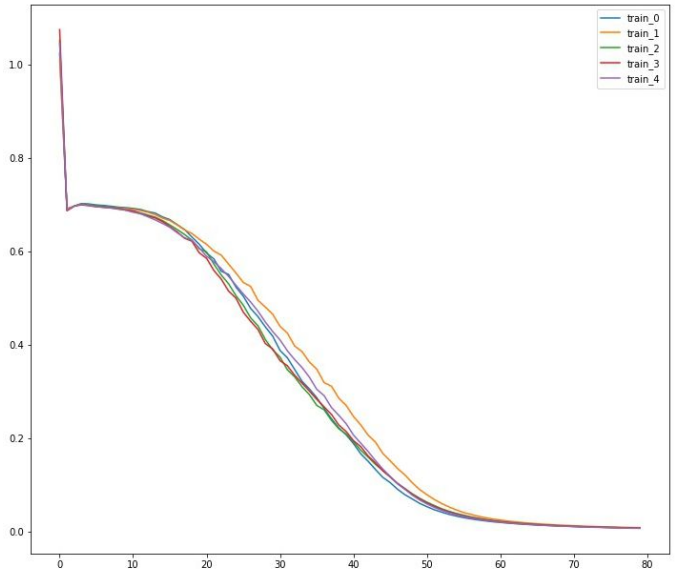
After testing different algorithms on IMDB dataset, a similar study was conducted for the second dataset (self-created bloggers corpus). We performed additional data collection on the basis of the texts of various bloggers. This dataset has been used to address binary classification task. Contrary to the findings of the movie reviews in IMDB dataset, the classification was carried out to determine the gender of the author of the text. The data distribution of the second dataset consists of texts of various people 500 men and 500 women. Each text contains more than 8,000 words and more than 64,000 characters. All texts are the union of several publications by one author, the main language of texts is Russian using single words in English, French, Spanish, German and Chinese.

For the second dataset, to compare deep learning models with traditional machine learning models, initially we applied Naïve Bayes algorithm for the self-created bloggers corpus. This delivers significantly bad results (64.5% accuracy). The exact reason for this observed decline might be due to the fact that the texts was quite large. The figure shows the ROC-curve obtained after training the Naïve Bayesian classifier.

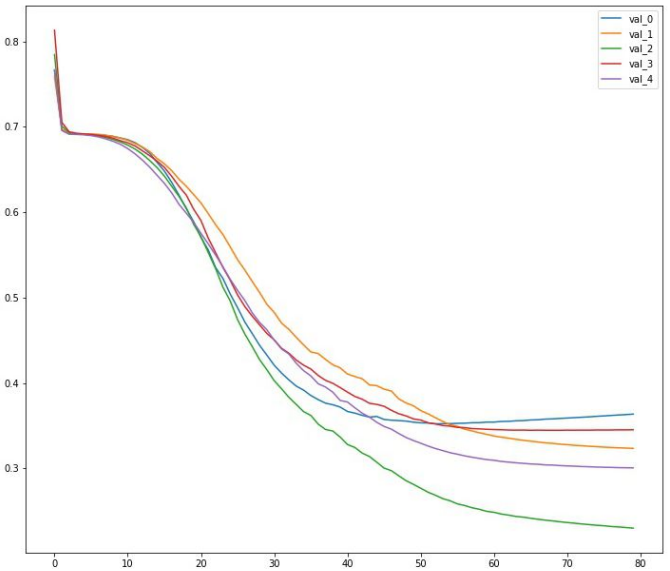


The result of this analysis is then compared with Convolutional Neural Network (CNN). The next step in the experiments was the use of a Convolutional Neural Network (CNN). To form the input data for the neural network, the texts were processed using the udpipe library and the lemmatization of words was performed. It is important to mention that for the texts in Russian language, word embeddings (word vectors) was obtained by using the model ruwikiruscorpora_upos_skipgram_300_2_2018.vec [32] [33].

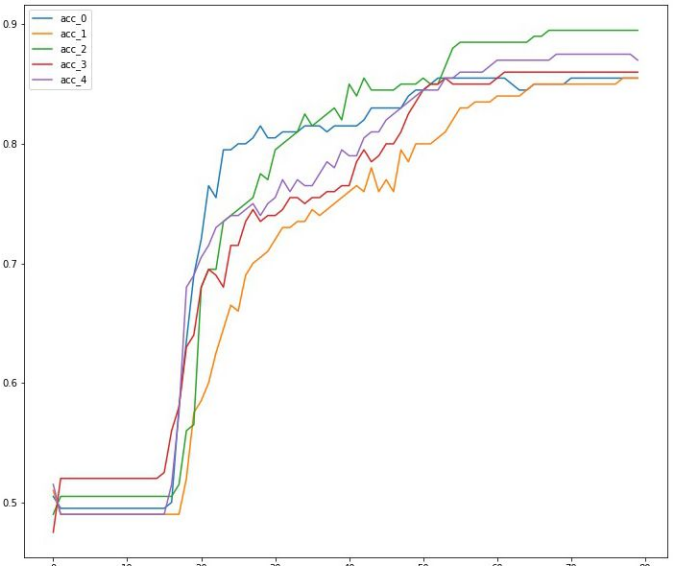
The structure of the neural network consists of four layers. The first layer is word2vec embedings with a dimension of 300 (pre-trained). The second layer is a one-dimensional convolutional layer, the width of the window is 3, the number of units is 200. Third layer - max pooling. Fourth layer - linear layer, 201 parameter. The training of this convolutional neural network took place until the moment when the error in the validation sample ceased to decrease (the experiments required 50 epochs). In this case, the Adam optimization algorithm was used and batch size 32 was used. This yields increasingly good results (accuracy up to 85.2% for the task of classifying the author's gender by text) using this dataset. However, even better results are achieved when the texts of one author were divided into many short texts of 500 words each, which made it possible to increase the accuracy of the trained convolutional neural network to 86.8%. By increasing the amount of processed data, subsequent training was conducted using cross-validation with five folds. The learning process for each fold is presented using the graphs of changes in the loss for each epoch of training separately for the training data and for the test data. As well as on the test data for each epoch of training an accuracy graph is plotted.



Loss on the train data at every fold of cross-validation



Loss on the test data at every fold of cross-validation



Accuracy on the test data at every fold of cross-validation

Conclusion

It is important to highlight the fact that, by analogy with various problems of image classification, convolutional neural networks have performed well, given good results for such various tasks as sentiment classification and gender classification of the author. At the same time, a very important factor in successfully solving the task is the use of distributive semantic models, but this approach works equally well for different languages - convolutional neural networks showed the best results in comparison with other approaches. Future research should be devoted to the development of more robust algorithms to get better results and further reduce the time complexity.

References

[1] Aggarwal, C. C., and Zhai, C. (2012). A survey of text classification algorithms. In Mining text data. Springer. 163-222.

[2] Hingmire, S.; Chougule, S.; Palshikar, G. K.; and Chakraborti, S. (2013). Document classification by topic labeling. In SIGIR, 877-880.

[3] Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. (2003). A Neural Probabilistic Language Model. JMLR 3:1137-1155.

[4] Cai, L., and Hofmann, T. (2003). Text categorization by boosting automatically extracted concepts. In SIGIR, 182-189.

[5] Cover, T. M., and Thomas, J. A. (2012). Elements of information theory. John Wiley Sons.

[6] Post, M., and Bergsma, S. (2013). Explicit and implicit syntactic features for text classification. In ACL, 866-872.

[7] Mikolov, T.; Yih, W.T. and Zweig, G. (2013) Linguistic regularities in continuous space word representations. In hlt-Naacl, 746-751.

[8] Sahami, M., Dumais, S., Heckerman, D., Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. . Learning for Text Categorization: Papers from the AAAI Workshop, pp. 55-62. Tech. rep. WS-98-05, AAAI Press.

[9] Socher, R.; Huang, E. H.; Pennington, J.; Ng, A. Y.; and Manning, C. D. (2011a). Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In NIPS, volume 24, 801-809.

[10] Socher, R.; Pennington, J.; Huang, E. H.; Ng, A. Y.; and Manning, C. D. (2011b). Semi-supervised recursive autoencoders for predicting sentiment distributions. In EMNLP, 151-161.

[11] Elman, J. L. (1990). Finding structure in time. Cognitive science 14(2):179-211.

[12] Kim Y. Convolutional neural networks for sentence classification arXiv preprint arXiv:1408.5882. – 2014.

[13] Collobert R. et al. Natural language processing (almost) from scratch Journal of Machine Learning Research. – 2011. – T. 12. – №. Aug. – C. 2493-2537.

[14]] Kalchbrenner, N., and Blunsom, P. (2013). Recurrent convolutional neural networks for discourse compositionality. In Workshop on CVSC, 119-126.

[15] Lewis, D. D. (1992). An evaluation of phrasal and clustered representations on a text categorization task. In SIGIR, 37-50.

[16] Mnih, A., and Hinton, G. (2007). Three new graphical models for statistical language modelling. In ICML, 641-648.

[17] Ng, A. Y. (2004). Feature selection, l1 vs. l2 regularization, and rotational invariance. In ICML, 78.

[18] Post, M., and Bergsma, S. (2013). Explicit and implicit syntactic features for text classification.

[19] Hinton, G. E., and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science 313(5786):504-507.

[20] Bengio, Y.; Courville, A.; and Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE TPAMI 35(8):1798-1828.

[21] Huang, E. H.; Socher, R.; Manning, C. D.; and Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. . In ACL, 873-882.

[22] Mikolov, T. (2012). Statistical language models based on neural networks. . Ph.D. Dissertation, Brno University of Technology.

[23] Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In EMNLP, 1631-1642.

[24] Kingma,P.D; Ba,L.J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[25] SemEval-2016 URL: http://alt.qcri.org/semeval2016/ (accessed date : 31.05.2017)

[26] Mikolov T. et al. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems. – 2013. – C. 3111-3119.

[27] Glove URL: https://nlp.stanford.edu/projects/glove/ (accessed date: 31.05.2017)

[28] Mohammad S. M., Kiritchenko S., Zhu X. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. arXiv preprint arXiv:1308.6242. – 2013.

[29] Gamallo P., Garcia M. Citius: A Naive-bayes strategy for sentiment analysis on english tweets. Proceedings of SemEval. – 2014. – C. 171-175.

[30] Agarwal A. et al. Sentiment analysis of twitter data . Proceedings of the workshop on languages in social media. – Association for Computational Linguistics, 2011. – C. 30-38.

[31] Go A., Bhayani R., Huang L. Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford. – 2009. – T. 1. – No. 12.

[32] RusVectōrēs:https://rusvectores.org/ru/models/ (accessed date: 29.12.2018)

[33] Kutuzov A., Kuzmenko E. (2017) WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models. In: Ignatov D. et al. (eds) Analysis of Images, Social Networks and Texts. AIST 2016. Communications in Computer and Information Science, vol 661. Springer, Cham