On One Approach of Solving Sentiment Analysis Task for Kazakh and Russian Languages Using Deep Learning

Narynov Sergazy Sakenovich
and Arman Serikuly Zharmagambetov(✉)

"Alem Research" LLP, Dostyk Avenue 132, Office 13,
050051 Almaty, Kazakhstan
sergazy@gmail.com, armanform@gmail.com

Abstract. The given research paper describes modern approaches of solving the task of sentiment analysis of the news articles in Kazakh and Russian languages by using deep recurrent neural networks. Particularly, we used Long-Short Term Memory (LSTM) in order to consider long term dependencies of the whole text. Thereby, research shows that good results can be achieved even without knowing linguistic features of particular language. Here we are going to use word embedding (word2vec, GloVes) as the main feature in our machine learning algorithms. The main idea of word embedding is the representations of words with the help of vectors in such manner that semantic relationships between words preserved as basic linear algebra operations.

Keywords: NLP · Sentiment analysis · Deep learning · Machine learning · Text classification

1 Introduction

In recent years, there is an active trend towards using various machine learning techniques for solving problems related to Natural Language Processing (NLP). One of these problems is the automatic detection of emotional coloring (positive, negative, neutral) of the text data, i.e. sentiment analysis. The goal of this task is to determine whether a given document is positive, negative or neutral according to its general emotional coloring. We don't perform sentiment analysis related to particular object, i.e. it is not an aspect based sentiment analysis. Therefore, we deleted from our dataset document with mixed sentiment. Nevertheless, analyzing general sentiment of a document is difficult task by itself. The difficulty of sentiment analysis is determined by the emotional language enriched by slang, polysemy, ambiguity, sarcasm; all this factors are misleading for both humans and computers.

The high interest of business and researchers to the development of sentiment analysis are caused by the quality and performance issues. Apparently the sentiment analysis is one of the most in-demand NLP tasks. For instance, there are several international competitions and contests [1], which try to identify the best method for sentiment classification. Sentiment analysis had been applied on various levels, starting

N.T. Nguyen et al. (Eds.): ICCCI 2016, Part II, LNAI 9876, pp. 537–545, 2016.
DOI: 10.1007/978-3-319-45246-3_51

from the whole text level, then going towards the sentence and\or phrase level. In general, importance of solving this problem is considered in [16].

It is obvious that similar sentimental messages (text, sentence…) can have various thesauruses, styles, and structure of narration. Thus, points corresponding to the similar messages can be located far away from each other that make the sentiment classification task much harder [2]. The scientific novelty of the paper is in applying Word2Vec algorithm [3] in the sentiment classification task for Kazakh and Russian languages and use this word vectors as input to deep recurrent neural networks to deal with long term dependency of the textual document.

2 Related Works

The study of sentiment analysis has relatively small history. Reference [4] is generally considered the principal work on using machine learning methods of text classification for sentiment analysis. The previous works related to this field includes approaches based on maximum relative entropy and binary linear classification [5] and unsupervised learning [6].

Most of these methods use well known features as bag-of-words, n-grams, tf-idf, which considered as the simplest one [7]. But as show the experiment results the simple models often works better than complicated ones. Reference [7] use distant learning to acquire sentiment data. Additionally, since they mostly work with movie comments and tweets, they used additional features as ending in positive emoticons like ":)" ":-)" as positive and negative emoticons like ":(" ":-(" as negative. They build models using Naive Bayes, Maximum Entropy and Support Vector Machines (SVM), and they report SVM outperforms other classifiers. In terms of feature space, they try a Unigram, Bigram model in conjunction with parts-of-speech (POS) features. They note that the unigram model outperforms all other models.

The other feature is syntactic meta information. It's obvious that the recursively enumerable grammar describes the most complete of any natural language. The computational performance of the best syntactic parser of context free grammar is linear, so syntactic information is expensive for the sentiment analysis task. However, those experiments that involved dependency relations showed that syntax contributes significantly to both Recall and Precision of most algorithms. For the task of text classification in general see [8–10] deal with a task sentiment classification based on syntactic relations. In reference [11] it was shown that POS-tagging and other linguistic features contributes to the classifier accuracy. The experiments were conducted on feedback data from Global Support Services survey.

Sentiment analysis using recurrent and recursive neural networks described in [17–19]. Important fact is that authors could not find previous works related to automatic sentiment classification for Kazakh language.

3 Dataset

The labeled by human data set consists of ~30,000 news articles in Russian language, specially selected for sentiment analysis, which consist of 11,286 neutral, 10,958 positive and 7756 negative articles. The sentiment of each document can be one of the following: positive, negative, neutral. 18,000 reviews from this dataset were chosen as training data, 6,000 as cross validation dataset and 6,000 as test dataset. Furthermore, ~10000 (3021 positive, 2548 negative, 4431 neutral) news articles in Kazakh language were labeled in order to train sentiment classifier. Each entry on this dataset consists of the following field:

- Id - Unique ID of each review.
- Sentiment - Sentiment of the review: 1 for positive reviews, 0 for negative reviews, 2 to the neutral reviews.
- Text - Text of the document (on Kazakh or Russian language).

The goal is to increase the accuracy (precision and recall) in sentiment classification of test dataset.

Furthermore, we used ~70 GB of plain text data in Russian language and ~10 GB plain text data in Kazakh language in order to train word embedding by unsupervised method. These texts we obtained from open electronic libraries, news articles, crawled web sites, etc.

4 Learning Model

The first step of learning model is unsupervised training of word embedding. Word2vec, published by Google in 2013, is a neural network implementation that learns distributed representations for words. Distributed word vectors, i.e. word embeddings are powerful and can be used for many applications, particularly word prediction and translation. It accepts large un-annotated corpus and learns by unsupervised algorithms. There are two different architectures of Word2Vec algorithm. At Fig. 1 continuous bag of words (CBOW) architecture presented, the purpose of such network topology assumes mapping from context to particular term and vice versa at Fig. 1 skip-gram architecture that maps particular term to its context.

We used "gensim" [13] python library with build-in Word2Vec model. It accepts large textual dataset for training. As was mentioned above, 70 GB and 10 GB raw data in Russian and Kazakh languages, respectively. The following options were used while training for Word2Vec: 300 dimensional space, 40 minimum words and 3 words in context. The vector representation of word has a lot of advantageous. It raises the notion of space, and we can find distance between words and finding semantic similar words. The simple result that can be obtained for such vector presented in Table 1.

Finally, map with a word as key and N dimensional vectors as value is obtained from abovementioned word2vec algorithm. Next, these vectors will be used in classification task. But before, each article should be preprocessed.

The preprocessing includes the following: (1) The all HTML tags, punctuations, were removed by "Beautiful Soup" python library. There are HTML tags such as

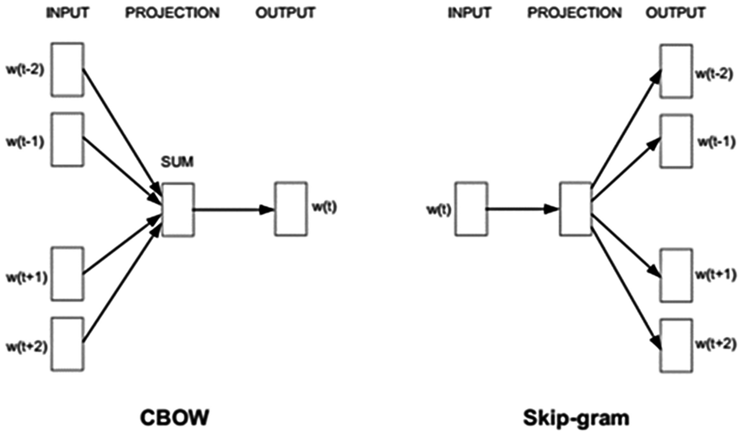

Fig. 1. The architectures of Word2Vec model. The CBOW architecture predicts the current word based on the context **(left)**. The skip-gram predicts surrounding words given the current word **(right)**. Taken from [3].

Table 1. Semantic similar words to Russian word - ‘man’

Words	cos dist
Woman	0,6056
Guy	0,4935
Boy	0,4893
Men	0,4632
Person	0,4574
Lady	0,4487
Himself	0,4288
Girl	0,4166
His	0,3853
He	0,3829

“< br/>”, abbreviations, punctuation - all common issues when processing text from online. (2) Moreover, numbers and links were replaced by tags NUM and LINK, respectively. (3) Removing stop words. Conveniently, there is Python package - Natural Language Toolkit (NLTK) [12] that removes stop words with built in lists. (4) Lemmatization of each word. For Russian language was used lemmatization tool “Mystem” [23], which was developed by Yandex. For Kazakh language there is no lemmatization. In future works we plan to implement morphological lemmatization tool for Kazakh language.

Next, each word is mapped to vector and for one document we get a sequence of N dimensional vectors which will be given as input to LSTM recurrent neural network.

Long Short-Term Memory (LSTM) is a special type of recurrent neural networks which was invented to consider sequential dependencies as a set of words in some text. Furthermore, LSTM overcomes common problems of RNN as exploding gradient or

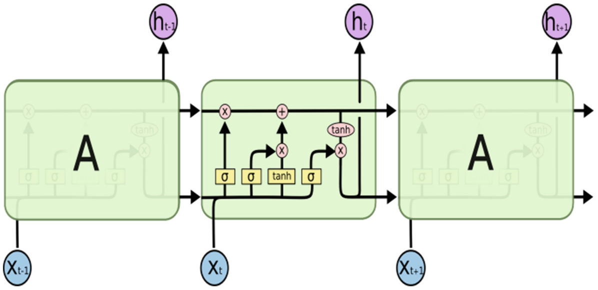

Fig. 2. The architecture of LSTM. Instead of one output it uses three gates: input, forget, output. Taken from [24].

vanishing gradient. To overcome this drawback LSTM uses additional internal transformation, which operate with memory cell more cautious represented in Fig. 2. For detailed information how LSTM works you can refer to [20].

For sentiment classification task we compared several model architectures with LSTM, which are presented in Figs. 3, 4, 5 and 6. General idea for all schemes is the same – input word vectors are processed via LSTM units, and then outputs from these units go further via vanilla neural networks or logistic regression unit. Below, in results section we give comparison of results of these various neural networks architectures.

Training algorithm was implemented using Theano [21] and Lasagne [22] packages for python language. C-extension for python (cython [15]) and GPU were used for acceleration and efficient calculations. In our experiments, using GPU gives up to 6–7 times faster calculation compared with CPU usage with multithread. Abovementioned packages give opportunity to easily implement various deep learning algorithms

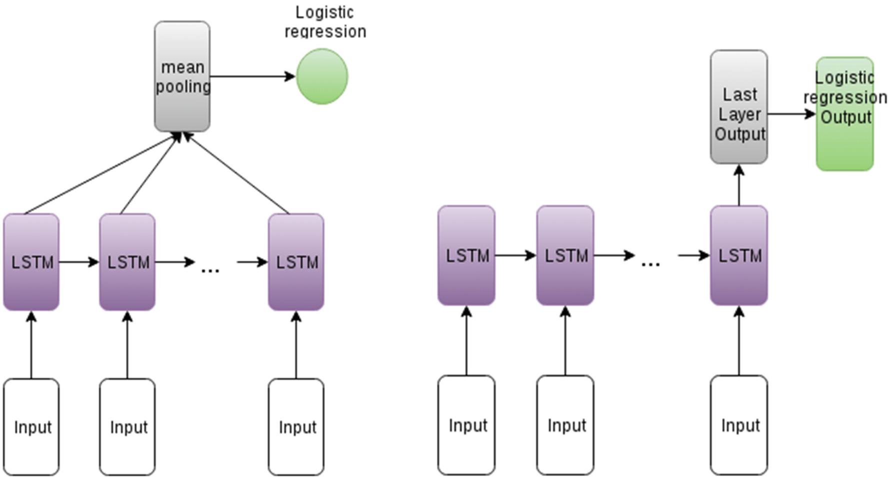

Fig. 3. Mean pooling from each output of LSTM followed by logistic regression **(left)**. Stacked LSTM where the last layer is sliced and proceed to logistic regression unit **(right)**.

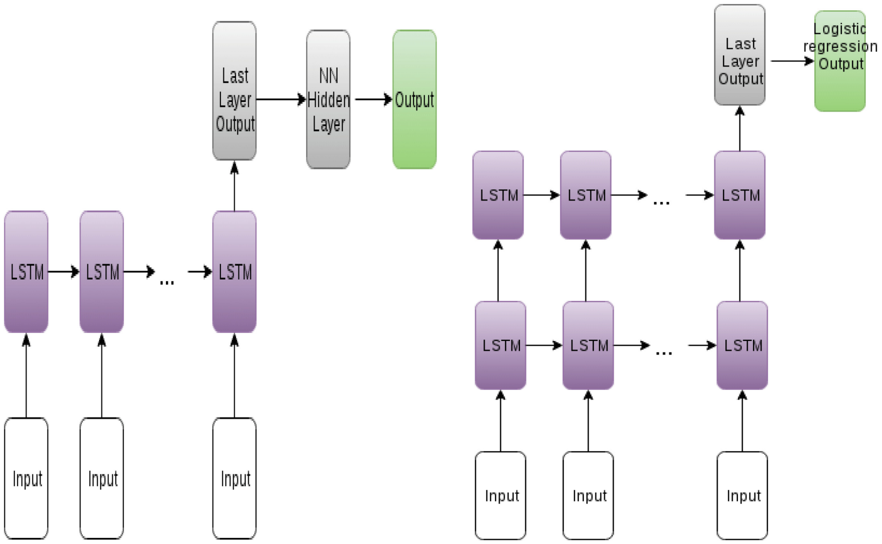

Fig. 4. Stacked LSTM where the last layer is sliced and proceeded to multilayer perceptron (Neural Network) unit **(left)**. Stacked two layer LSTM where the last layer is sliced and proceeded to logistic regression unit **(right)**.

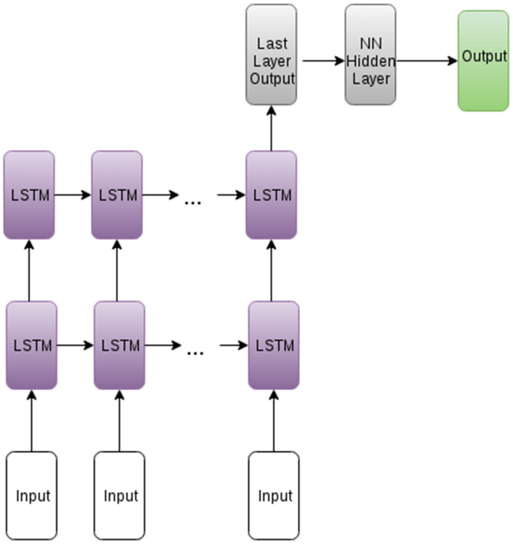

Fig. 5. Stacked two layer LSTM where the last layer is sliced and proceeded to multilayer perceptron (Neural Network) unit.

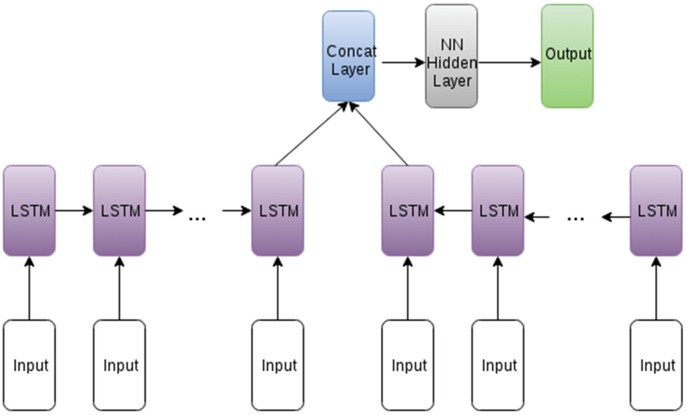

Fig. 6. Stacked bidirectional LSTM where the last layers of each direction are concatenated and proceeded to multilayer perceptron unit.

including LSTM, multilayer perceptron, etc. Also they have extension to use GPU to optimize computations.

Sigmoid and tanh functions were used as LSTM internal activation functions. Similarly, softmax was used as activation function for logistic regression and Neural Networks units. Advantage of softmax functions is that it gives correct generalization of the logistic sigmoid to the multinomial case:

$$h_i = \frac{e^{a_i}}{\sum_{j=0}^{N} e^{a_j}} \tag{1}$$

We used categorical cross entropy to define the loss function which should be optimized while training:

$$J = -\sum_{i=0}^{N}\sum_{j=0}^{m} y_j^{(i)} * \log(h_j^{(i)}) \tag{2}$$

5 Results and Discussions

Table 2 summarizes the results on sentiment classification. As mentioned above there are 30,000 news articles in Russian language and 10,000 news articles in Kazakh language were chosen for train and test data.

It can be seen that the best model for Russian sentiment analysis is - “Stacked two layer LSTM with one hidden layer Neural Networks” (Fig. 5) and for Kazakh language is - “LSTM with mean pooling and logistic regression unit” (Fig. 4). Another important fact is that sentiment analysis for Kazakh language shows worse result. Probably, it can be explained due to relatively small amount of training data set and lack of lemmatization.

Table 2. Results of sentiment classification

Method	Average precision (ru/kz)	Average recall (ru/kz)	Accuracy (ru/kz)
Mean pooling + log. reg. (Fig. 3 - left)	80.2 % ***73.2 %***	73.7 % 72.2 %	76.3 % ***72.8 %***
Stacked LSTM + log.reg (Fig. 3 - right)	81.3 % 69.1 %	77.2 % 61.3 %	80.8 % 67.3 %
Stacked LSTM + NN (Fig. 4 - left)	78.2 % 70.1 %	82 % **72.6 %**	82.8 % 70.5 %
Stacked two LSTM + log.reg (Fig. 4 - right)	81.6 % 70.4 %	85.7 % 71.3 %	85.2 % 70.9 %
Stacked two LSTM + NN (Fig. 5)	***84.5 %*** 71.1 %	***86.4 %*** 66.7 %	***86.3 %*** 69.8 %
Biderect. LSTM + NN (Fig. 6)	76.8 % 62.9 %	69.3 % 61.2 %	71.1 % 62.3 %

The given work shows that deep recurrent neural networks can be efficiently applied to the task of sentiment classification. Particularly, LSTM shows stable results even for long sequential data as words or sentences in a news article. Additionally, word embedding helps extract semantic relations between words which have effect to training process. Future works will be dedicated to improvement of sentiment classification by studying deeply long term dependencies in the text document and by extracting syntax relations. Neural Turing Machines, adversarial neural networks will be considered instead of or jointly with recurrent relation. Moreover, aspect based sentiment classification task will be studied.

References

1. Chetviorkin, I., Braslavskiy, P., Loukachevich, N.: Sentiment analysis track at ROMIP 2011. In: International Conference "Dialog 2012": Computational Linguistics and Intellectual Technologies, Bekasovo, pp. 1–14 (2012)
2. Pak, A.A., Narynov, S.S., Zharmagambetov, A.S., Sagyndykova, S.N., Kenzhebayeva, Z.E., Turemuratovich, I.: The method of synonyms extraction from unannotated corpus. In: DINWC 2015, Moscow, pp. 1–5 (2015)
3. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Workshop at ICLR, Scottsdale, AZ, USA (2013)
4. Bo, P., Lee, L.: A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: ACL (2004)
5. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
6. Turney, P.D.: Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In: 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002), Philadelphia, Pennsylvania, pp. 417–424 (2002)

7. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. Technical report, Stanford (2009)
8. Furnkranz, J., Mitchell, T., Riloff, E.: A case study in using linguistic phrases for text categorization on the WWW. In: AAAI/ICML Workshop on Learning for Text Categorization, pp. 5–12 (1998)
9. Caropreso, M.F., Matwin, S., Sebastiani, F.: A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In: Chin, A.G. (ed.) Text Databases and Document Management: Theory and Practice, pp. 78–102. Idea Group Publishing, USA (2001)
10. Nastase, B., Shirabad, J.S., Caropreso, M.F.: Using dependency relations for text classification. In: 19th Canadian Conference on Artificial Intelligence, Quebec City, pp. 12–25 (2006)
11. Gamon, M.: Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In: COLING 2004, Geneva, pp. 841–847 (2004)
12. Natural Language Toolkit. http://www.nltk.org/
13. Gensim: Topic modeling for humans. https://radimrehurek.com/gensim/
14. Sci-kit: Machine learning in python. http://scikit-learn.org/stable/
15. Cython: C-Extensions for Python. http://cython.org/
16. Liu, B.: Sentiment analysis and opinion mining. Synth. Lect. Hum. Lang. Technol. **5**(1), 1–167 (2012)
17. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 142–150. Association for Computational Linguistics (2011)
18. Tarasov, D.S.: Deep recurrent neural networks for multiple language aspect based sentiment analysis of user reviews. In: Dialog 2015, Moskow (2015)
19. Socher, R., Perelygin, A., Jean, Y.W., Chuang, J., Manning, C.D, Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1642–1656. Citeseer, Seattle (2013)
20. Hochreiter, S., Schmidhuber, J.: Long short-term memory. J. Neural Comput. **9**(8), 1735–1780 (1997)
21. Theano: Framework for python. http://deeplearning.net/software/theano/
22. Lasagne: Framework for python. https://github.com/Lasagne/Lasagne
23. Mystem: Morphology analysis tool. https://tech.yandex.ru/mystem/
24. Understanding LSTM Networks. Colah’s personal blog. http://colah.github.io/posts/2015–08-Understanding-LSTMs/