Deep Recurrent Neural Networks for Bilingual Sentiment Analysis of Short Texts

Yelaman Abdullin
Kazan Federal University
Kazan, Russian Federation
a.elaman.b@gmail.com

Radhakrishnan Delhibabu
Kazan Federal University
Kazan, Russian Federation
r.delhibabu@gmail.com

Vladimir Ivanov
Innopolis University
Innopolis, Russian Federation
v.ivanov@innopolis.ru

***Abstract*—Sentiment analysis of short texts such as Twitter messages and comments in news portals is challenging because of the limited contextual information that they normally contain. In this paper, we propose our deep neural network model that use bilingual word embeddings for effectively solving classification problem for both languages. We apply our approach for two corpora of two different language pairs: English-Russian and Russian-Kazakh. We show how to train a classification model in one language and predict in another. Our approach achieves good results for English with 73% accuracy and Russian 74% accuracy. Also we have built a baseline method for Kazakh sentiment analysis with 60% accuracy and have proposed a method to learn bilingual embeddings from a large unlabeled corpus using set of bilingual word pairs.**

I. Introduction

Sentiment analysis is an increasingly active problem. Consumers use the web as an advisory body in influencing their view on matters. Knowing what is said on the web allows to react upon negative sentiment and to monitor positive sentiment. The social media connect the entire world and thus people can much more easily in influence each other. Hundreds of millions of people around the world actively use websites such as Twitter or News portals to express their thoughts. Thus there is growing interest in sentiment analysis of texts where people express their thoughts or their opinion across a variety of domains such as commerce [1] and health [2], [3].

Sentiment analysis is the process of automatically determining sentiment expressed in natural language. As social media cover almost the entire world, a sentiment expressed by users of social media is written in a multitude of languages. Here we face a new problem. For some languages, e.g. kazakh language, there is no large enough labeled corpora to use them as a training data for sentiment analysis. The problem we study in this paper is to determine the general opinion expressed in texts written in one natural language taking into account another language and how to apply these information for train a sentiment classifier.

Here we focus on short text, in particular, on social media data: news comments and micro-blogging posts, like Twitter messages. Sentiment analysis of short texts is challenging because of the limited amount of contextual data in this type of text. In this work, we propose a deep recurrent neural network that use bilingual word embeddings to capture semantic features between words of two languages. We perform experiments on two language pairs: English-Russian and Russian-Kazakh. A sentiment have been one of the two classes: positive and negative.
In this paper, we describe an approach to building bilingual word embeddings and how to use it to create a deep neural network classifying model, which achieves good competitive performance on sentiment analysis for Russian language. We evaluate the model on a baseline in sentiment analysis for Kazakh language.

II. Related Works

Distributed representations of words also known as word embeddings have been introduced as a part of neural network architectures for statistical language modeling ([4], [5], [6], [7]). Generally, word embeddings is a very natural idea which treat words like a math objects. Classical approach to building word embeddings constructs one-hot encoding, where each word corresponds with its own dimension. Obviously, there is a necessity to train representations for individual words, basically, as reduction of dimensionality. In particular, distributed word representations solve this problem. It maps each word occurring in the dictionary to a Euclidean space, attempting to capture semantic attitudes between words as geometric relationships. Thus distributed word representations are very useful in different NLP tasks such as semantic similarity [8], information retrieval [9] and sentiment analysis [10].

There are few methods to build multilingual word embedding. In particular, Zou et al. [8] introduced bilingual word embeddings through utilizing Machine Translation word alignments to translational equivalence. Vulic and Moens [11] proposed a simple effective approach of learning bilingual word embeddings from non-parallel document-aligned data. Also Lu et. al [12] extend the idea of learning deep non-linear transformations of word embeddings for two languages, using the deep canonical correlation analysis.

Sentiment analysis tasks of short text is a very popular task in NLP. Mohammad et. al [13] described one of the state-of-the-art Twitter message-level sentiment classifying using SVM. Dos Santos and Gatti [10] proposed a deep convolutional neural network exploiting character-level and word-level embeddings to perform sentiment analysis of short texts, and achieved state-of-the-art results in binary classification, with 85.7% accuracy. Although there are many works related to these models, little work has been done to use bilingual word embeddings to improve sentiment analysis, especially for Kazakh language.

III. Method

A. Background

1) Bilingual Word Embeddings: Assume that we have two large not aligned corpora in the source language W_S and the target language W_T , respectively, and set of bilingual word pairs(dictionary) for each language V_S and V_T. Our goal is to generate vectors x and y in space R^{S+T} and retain semantic relationships between vectors from both source spaces and supplement them with new relationships between words of two languages. For example, in the joint semantic space the Russian word '*школа*'(school) is expected to be close to its Kazakh translation '*мектеп*'(school). Besides words '*школы*'(plural form of Russian word 'школа', schools) and '*мектептер*'(plural form of Kazakh word 'мектеп', schools) which are not contained in the dictionary is also expected to be near.

There are several way to solve this problem. We consider two methods in this paper. We propose a relatively straightforward method to creating multilingual word embeddings. Main idea of this method is generating a single "pseudo-bilingual" corpus through mixing source corpora with a second language. In the first step, we clean dictionaries V_S and V_T depending on the frequency of words in their corpora. We delete very common words using threshold. Because that words commonly used in different contexts, they have different meanings. Following that, we have randomly splitted source language corpus to two parts and replace every n-th word in first half with direct translation given in dictionary V_S. Exactly the same step we apply to target language corpus.This have been done in order to extend context of using particular word to two language. Having a bilingual contexts for each word in pseudo-bilingual corpus, we train the final model and construct a shared multilingual embedding space. The second method is to train word embeddings for each language and then applying linear regression transform word embeddings from source to target language. This method was proposed by Mikolov et al. [14]. The objective function in regression task looks as follows:

$$\min_{\beta} \sum_{i} ||\beta x_i - y_i||^2 \tag{1}$$

where β is a transformation matrix which we have to find out, x_i and y_i are word vectors of source and target language word spaces respectively.

2) Neural network architecture: As a deep neural network we use the Long short-term memory (LSTM) model, which is proposed by Hochreiter and Schmidhuber [15]. LSTM model is a type of recurrent neural network (RNN). In a traditional recurrent neural network, during the gradient back-propagation phase, the gradient signal can end up being multiplied a large number of times by the weight matrix associated with the connections between the neurons of the recurrent hidden layer. This means that, the magnitude of weights in the transition matrix can have a strong impact on the learning process. RNN makes all predictions sequentially, and the hidden layer from one prediction is fed to the hidden layer of the next prediction. This gives the network "memory", in the sense that the results from previous predictions can inform future predictions. LSTMs are explicitly designed to avoid the long-term dependency problem. Thus LSTM networks is especially good in sequence labeling tasks [16].

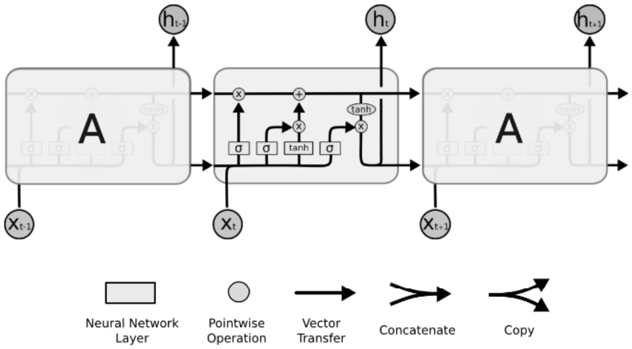

Fig. 1. This diagram shows LSTM memory block architecture. Picture from [17]

Likewise, we are interested in evaluating the performance one more recently proposed recurrent unit - GRU. A gated recurrent unit (GRU) was proposed by Cho et al.[18] to make each recurrent unit to adaptively capture dependencies of different time scales. Similarly to the LSTM unit, the GRU has gating units that modulate the flow of information inside the unit, however, without having a separate memory cells.

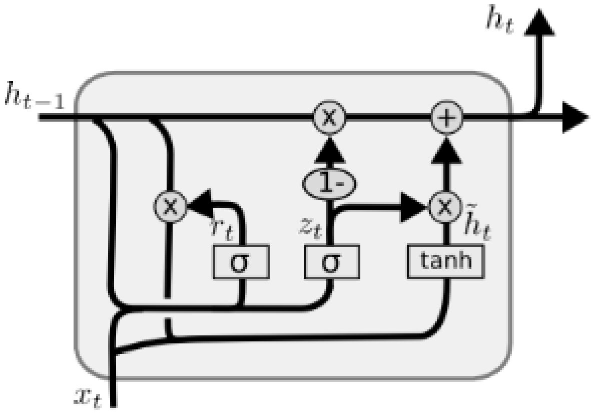

Fig. 2. This diagram shows GRU memory block architecture. Picture from [17]

In order to score a sentence, the network takes as input the sequence of words in the sentence, and passes it through a sequence of layers where features with increasing levels of complexity are extracted.

B. Scoring and Network Training

Given a sentence x with n words $w_1, w_2, ..., w_n$, which have been converted to joint word-level embeddings. We use a special padding token for sentences with small sizes. Then we get sentence-level representation passing word-level embeddings through two LSTM layers. Finally, the vector r_x, the obtained feature vector of sentence x, is processed by two fully connected(dense) neural network layers, which extract one more level of representation and compute a score for each sentiment label $c \in C$ as a logistic classifier.

The network was trained using RMSProp([19], [20]), which worked better than using an annealed learning rate. Also we use dropout [21] as a powerful regularizer, even when the network was only two layer deep. Also we use dropout technique to regularize hidden states in LSTM layer. So that we have achieved good generalization capability getting opportunity to train a neural network in one language and predict in another.

IV. Experimental Setup and Results

A. Sentiment Analysis Datasets

We apply our model for two different language pairs: English-Russian and Russian-Kazakh. As the English sentiment labeled dataset we use the Standford Twitter Sentiment corpus introduced by Go et al. [22]. In our experiments, to speedup the training process we use only a sample of the training data consisting of 100K randomly selected tweets. As the Russian dataset we use Russian Twitter corpus introduced by Rubtsova and Zagorulko [23]. For the Kazakh dataset we collect a corpus from news comments including about 1400 documents. At the preprocessing step we have deleted sentence boundaries, non-letter characters(except apostrophe symbol) and have replaced all URLs to hashtag ”#ReplacedUrl”. Also we removed all emoticons, because training corpora was built using emoticon labeling and it have a huge impact to final results, whereas our goal is to achieve competitive results in bilingual text evaluations.

TABLE I. Sentiment Analysis datasets

Dataset	tweets / documents		classes
English	train test	80K 20K	2
Russian	train test	80K 20K	2
Kazakh	train test	1100 300	2

B. Unsupervised Learning of Bilingual Word Embeddings

Word embeddings play very important role in the our model architecture. They are meant to capture syntactic and semantic information, which are very important to sentiment analysis. In our experiments, we perform unsupervised learning of word embeddings using the word2vec tool [24], which implements the continuous bag-of-words and skip-gram architectures for computing vector representations of words [6]. We use the English Wikipedia corpus, a collection of Russian news documents and a collection of Kazakh news documents [25] as a source of unlabeled data. We removed all documents that are less than 50 characters long. Also we lowered case all words and substituted each numerical digit by a 0(e.g., 25 becomes 00). The resulting cleaned corpora contains about 280 million tokens for English, about 190 million tokens for Russian and about 20 million tokens for Kazakh.

After the preprocessing we start to ”mix” mentioned corpora to each other in the following manner. We select replacing window size of 6 and further the same window size will be used for training skip-gram. Following that we get two corpora for English-Russian pair and Russian-Kazakh pair.

When running the word2vec tool, we set that a word must occur at least 4 times in order to be included in the vocabulary, which resulted in a vocabulary of about 900K entries for English-Russian pair and about 600K for Russian-Kazakh pair. The training time for the English-Russian pair corpus is around 4h and around 1h for Russian-Kazakh pair corpus using 6 threads in a Intel(R) Core i5-3470 3.20GHz machine.

We show visualization of learned embeddings using Russian-Kazakh ”pseudo-bilingual” corpus in Fig. 2. The two-dimensional vectors for this visualization is obtained with t-SNE [26].

For linear transformation approach we use the same preprocessing methods, but have trained word embeddings for each language separately. Following that, using Ridge regression we transform word vectors in source language space into target language word vectors space. In section IV-D we show results for both described approaches.

C. Model setup

We implemented our model using Keras library [27] and Theano library [28] as ”backend” of Keras. We use the development sets to tune the neural network hyper-parameters. Many different combinations of hyper-parameters can give similarly good results. We spent more time tuning the regularization parameters than tuning other parameters, since it is the hyper-parameter that has the largest impact in the prediction performance. For both language pairs, the number of training epochs varies between two and four. In the Table II, we show the selected hyper-parameters.

D. Results for English-Russian pair

For this experiment we use two different bilingual word embeddings and compare them in solving bilingual sentiment analysis problem. For using the linear transformation approach we utilize bilingual dictionary to collecting training set. The collected training set have contained about 90K samples. We use Ridge regression, which introduced in Scikit learn library[29] with following parameters: regularization constant(alpha) - 0.01, precision of the solution(tol) - 0.0001.

First, we start to train our model only in English training data and evaluate model on Russian test dataset. Following that, we use both of English and Russian training sets in different concentrations. In Table III we show how the quality grows while we add more Russian training data.

In Table III, we also compare our model performance with other approaches proposed by the Go et al. [22]. Our results do not outperform the previous approaches, because we don’t use preprocessing features mentioned in his paper. Also training bilingual word vectors makes some noise into our word vector space.

E. Results for Russian-Kazakh pair

We have ran exactly the same experiments with Russian-Kazakh pair. But our Kazakh sentiment labeled dataset was

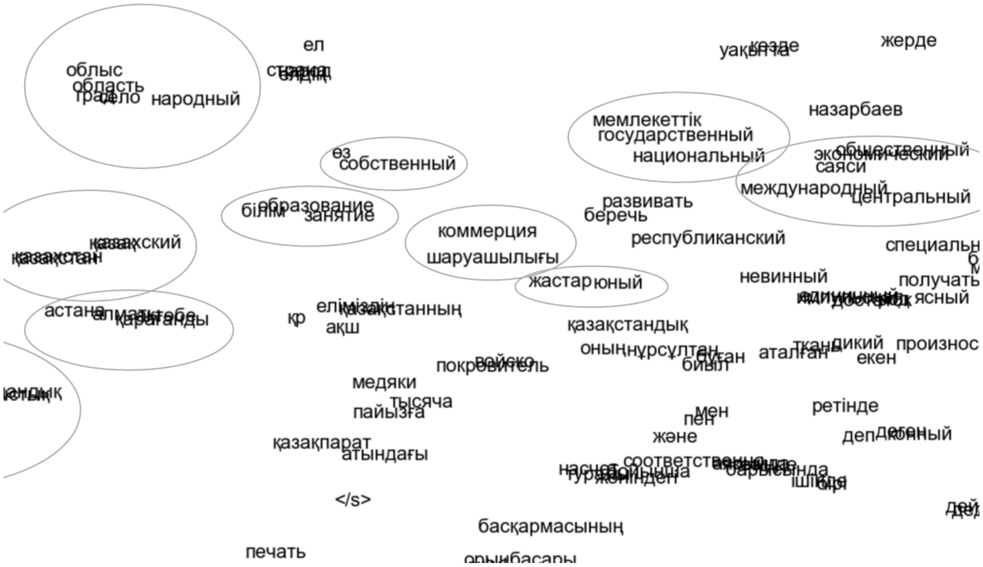

Fig. 3. Visualization of bilingual word embeddings. Circled positions shows a semantic proximity between Russian and Kazakh words. For example, 'мемлекеттік' - adj, state (from Kazakh), 'государственный' - adj, state (from Russian), 'национальный' - adj, national (from Russian)

TABLE II. Neural network parameters

Parameter	Parameter description	Value
d^E	Fraction of the embeddings to drop.	0.25
n^R	Number of hidden units in LSTM(GRU) layer.	64
d^W	Fraction of the input units to drop for input gates for LSTM(GRU) layer.	0.2
d^W	Fraction of the input units to drop for recurrent connections for LSTM(GRU) layer.	0.2
λ	Learning rate	0.001

TABLE III. Evaluating English-Russian pair model

Training data	Accuracy English	ROC AUC English	Accuracy Russian	ROC AUC Russian
Our approach(to building a bilingual word embeddings)				
100% English	0.73	0.80	0.59	0.62
75% English and 25% Russian	0.73	0.81	0.67	0.76
50% English and 50% Russian	0.74	0.81	0.70	0.78
Linear transformation approach				
100% English	0.69	0.74	0.55	0.60
75% English and 25% Russian	0.70	0.77	0.59	0.60
50% English and 50% Russian	0.71	0.77	0.60	0.64
100% English, SVM (Go et al. [22])	0.82	-	-	-
100% English, NB (Go et al. [22])	0.83	-	-	-

too small and we use also a small concentration of Kazakh training samples in dataset. In Table IV we show results for Russian-Kazakh pair. Again we see growth of quality for using the "language mixed" dataset.

TABLE IV. Evaluating Russian-Kazakh pair model

Training data	Accuracy Russian	ROC AUC Russian	Accuracy Kazakh	ROC AUC Kazakh
100% Russian	0.71	0.79	0.55	0.58
98% Russian and 2% Kazakh	0.72	0.80	0.56	0.64
95% Russian and 5% Kazakh	0.72	0.79	0.58	0.67

F. *Code and Data Sets*

This section describes the network architectures and training details for the experimental results reported in this paper. The code for reproducing these results can be obtained from https://github.com/eabdullin/nlp_mthesis. The implementation based on Keras library and Theano as back-end using CPU. But also there is a possibility to use GPU. More detailed description of using GPU with Keras may be found on [30].

V. Conclusion

In this work we present an approach to performing bilingual sentiment analysis. Also we propose a new relatively simple approach to building word embeddings. The main contributions of the paper are: (1) the new approach to building bilingual word embeddings; (2) the idea of using pre-trained bilingual word embeddings in neural network architecture; (3) experimental results for Kazakh sentiment analysis.

Proposed method may be used to sentiment analysis in different language which does not have enough labeled corpora. For this purpose researches need to have only dictionaries to translate words. As a future work, we would like to build Kazakh sentiment labeled corpus using our classification model. Additionally, we would like to check the impact of performing the semi-supervised learning.

Acknowledgment

This work is supported by Russian Science Foundation (project 15-11-10019 ”Text mining models and methods for analysis of the needs, preferences and consumer behavior”). Authors thank the Everware team (https://github.com/orgs/everware/people) for access to their platform. Also we would like to thank Yerlan Seitkazinov, Zarina Sadykova and Alfiya Sitdikova for manual annotation of the Kazakh sentiment corpus.

References

[1] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury, “Twitter power: Tweets as electronic word of mouth,” *Journal of the American society for information science and technology*, vol. 60, no. 11, pp. 2169–2188, 2009.

[2] C. Chew and G. Eysenbach, “Pandemics in the age of twitter: content analysis of tweets during the 2009 h1n1 outbreak,” *PloS one*, vol. 5, no. 11, p. e14118, 2010.

[3] M. J. Paul and M. Dredze, “You are what you tweet: Analyzing twitter for public health.” *ICWSM*, vol. 20, pp. 265–272, 2011.

[4] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain, “Neural probabilistic language models,” in *Innovations in Machine Learning*. Springer, 2006, pp. 137–186.

[5] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.

[7] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation.” in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[8] W. Y. Zou, R. Socher, D. M. Cer, and C. D. Manning, “Bilingual word embeddings for phrase-based machine translation.” in *EMNLP*, 2013, pp. 1393–1398.

[9] C. D. Manning, P. Raghavan, H. Schütze *et al.*, *Introduction to information retrieval*. Cambridge university press Cambridge, 2008, vol. 1, no. 1.

[10] C. N. dos Santos and M. Gatti, “Deep convolutional neural networks for sentiment analysis of short texts.” in *COLING*, 2014, pp. 69–78.

[11] I. Vulic and M.-F. Moens, “Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*. ACL, 2015.

[12] A. Lu, W. Wang, M. Bansal, K. Gimpel, and K. Livescu, “Deep multilingual correlation for improved word embeddings,” in *Proceedings of NAACL*, 2015.

[13] S. M. Mohammad, S. Kiritchenko, and X. Zhu, “Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets,” *arXiv preprint arXiv:1308.6242*, 2013.

[14] T. Mikolov, Q. V. Le, and I. Sutskever, “Exploiting similarities among languages for machine translation,” *arXiv preprint arXiv:1309.4168*, 2013.

[15] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] A. Graves, *Supervised sequence labelling*. Springer, 2012.

[17] C. Olah. (2015, aug) Understanding lstm networks. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs

[18] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” *arXiv preprint arXiv:1409.1259*, 2014.

[19] T. Tieleman and G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural Networks for Machine Learning*, vol. 4, p. 2, 2012.

[20] Y. N. Dauphin, H. de Vries, J. Chung, and Y. Bengio, “Rmsprop and equilibrated adaptive learning rates for non-convex optimization,” *arXiv preprint arXiv:1502.04390*, 2015.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” *CS224N Project Report, Stanford*, vol. 1, p. 12, 2009.

[23] Y. V. Rubtsova and Y. A. Zagorulko, “An approach to construction and analysis of a corpus of short russian texts intended to train a sentiment classifier,” *The Bulletin of NCC*, vol. 37, pp. 107–116, 2014.

[24] Google. Tool for computing continuous distributed representations of words. [Online]. Available: https://code.google.com/p/word2vec/

[25] O. Makhambetov, A. Makazhanov, Z. Yessenbayev, B. Matkarimov, I. Sabyrgaliyev, and A. Sharafudinov, “Assembling the kazakh language corpus.” in *EMNLP*, 2013, pp. 1022–1031.

[26] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.

[27] F. Chollet, “Keras: Theano-based deep learning library,” *Code: https://github. com/fchollet. Documentation: http://keras. io*, 2015.

[28] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, “Theano: a cpu and gpu math expression compiler,” in *Proceedings of the Python for scientific computing conference (SciPy)*, vol. 4. Austin, TX, 2010, p. 3.

[29] Machine learning in python. [Online]. Available: http://scikit-learn.org/

[30] F. Chollet. Keras: Deep learning library for theano and tensorflow. [Online]. Available: http://keras.io