Robust Word Vectors for Russian Language

Valentin Malykh
Laboratory of Neural Systems and Deep Learning,
Moscow Institute of Physics and Technology
Moscow, Russia
valentin.malykh@phystech.edu

***Abstract*—Many successful machine learning methods are developed to work with inputs in real-valued vector spaces. One way to modify them for working with inherently discrete word data in natural language processing systems is to map the words into such space. Several approaches that do that exist but they assume a fixed vocabulary for the words. This assumption does not generally hold in data collected from unedited conversations due to presence of typos and spelling idiosyncrasies. In this paper we address the problem by developing an deep learning architecture of robust word vectors for Russian Language. The architecture is designed to be robust to typos and deliberate spelling distortions in texts that do not adhere to a fixed vocabulary.**

I. Introduction

The problem of user input data is widely known: typos, errors, incorrect word usage and more. A number of different tools exists to handle this issue – embedded spell-checkers in the input forms on websites is an example – but severity of this problem keeps growing larger, especially with the wide usage of mobile devices with small and/or simplified keyboards.

Meanwhile, in a variety of machine learning tasks for natural language processing (NLP) word vectors had become very popular in recent years, especially in task of text classification, paraphrase detection, sentiment analysis, etc. Unfortunately, to use this word vectors the user input should be cleared from the noise. To address this issue we developed a novel architecture of word vectors that is insensitive to specific type of noise: missing or extra letters. The formal contributions of the paper are:

- A word vector architecture that is robust to typos.
- Testing this architecture on Russian paraphrase corpus.

II. Related work

Our architecture is based on the work of Tomas Mikolov [3, 4] and also [5], where the authors demonstrate a stable recognition of vocabulary words. Both of the mentioned approaches are lacking the support of out of vocabulary words (OOV), which could be an issue with noisy input and/or input that includes rare specific words. To demonstrate the robustness of the proposed architecture we have conducted an experiment on a clean corpus—first described in [19]—with artificial noise (see Section IV-A). To test our hypothesis of an improved handling of OOV, we conducted an experiment on a different corpus with a lot of infrequent words [14].

Also, some ideas in our work were inspired by the Long Short-Term Memory networks [6]. The Begin-Middle-End (BME) representation is related to Begin-Intermediate-End representation [5].

A related model was presented in [15]. The authors used mapping of characters to some embeddings in contrast to above-mentioned BME representation. These embeddings were then used as an input to the bidirectional LSTM. Similarly to Sakaguchi et al. [5] the output of the Bi-LSTM is used as input for SoftMax layer on the vocabulary.

Paraphrase detection is a task from the family of closely related tasks: the paraphrase detection itself, the documents similarity detection, and the plagiarism detection. In this work, we consider work on all of this tasks as related to this paper, since the proposed architecture does not rely on specific restrictions of any of this fields. Some prior literature should be mentioned, but the work has been mainly focused on web-data: an article about web corpus for Russian language [9], [13], detection of near-duplicates for web-documents [11], strings comparison presented in [10]; for the specific corpus used in this work, paraphrase detection was described in [17]. Plagiarism detection task is presented in [12] for students work (but there is no publicly available corpus) and for scientific texts [14].

III. Architecture

To represent a word we use the BME representation. The *B* part of the representation consists of the one-hot encoding for the first three letters, the *E* part consists of the one-hot encoding for the last three letters, and the *M* part is the sum of one-hot encoded vectors for all the letters in the word.

The graphical representation of the architecture is presented in Fig. 1. The model consists of the first fully connected (FC) layer, where BME representation is mapped to vector of LSTM layer width. Next is the LSTM layers themselves. And in the end is again FC layer to produce fixed size vector.

A. The Negative Sampling

For training we use negative sampling as it was shown to capture semantic properties of words. The negative sampling according to [3] is :

$$L(x) = \log(\sum_{i \in C} e^{-s(x,w_i)}) + \log(\sum_{j \notin C} e^{s(x,w_j)}), \qquad (1)$$

where C is the set of indices of words in context for word x. The context is defined as words in predefined window, surrounding the given one. $s(x, w)$ is a similarity scoring function for two words.

ISBN 978-952-68397-8-3 (PDF)

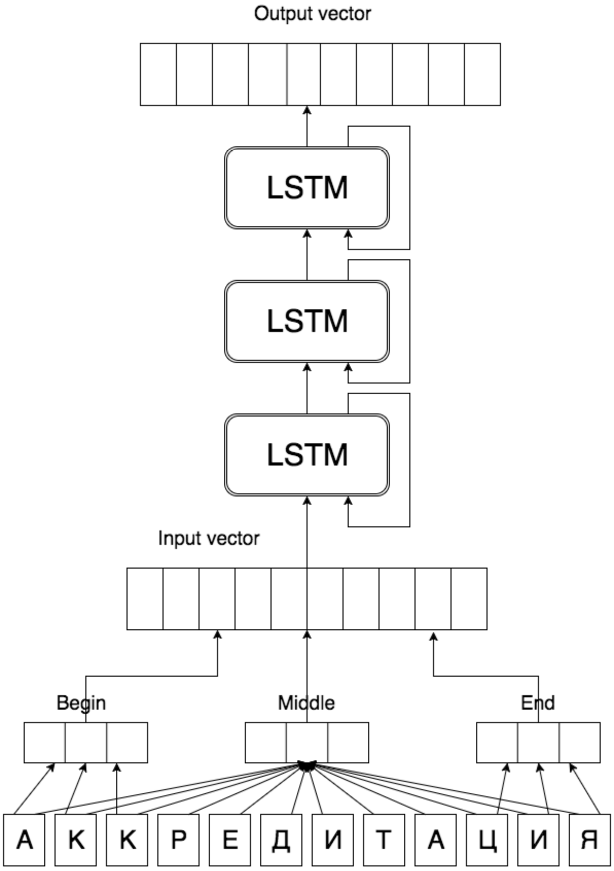

Fig. 1. A graphical representation of the proposed model

B. Technical details

In our model the cosine similarity of word vectors produced by the output layer of the network is used as scoring function.

We have used three LSTM layers (as pictured on the figure) due to gradient vanishing which is typical for LSTM networks. There are some methods to handle this issue, like residual connections introduced in [16], but the application of these methods is a subject of further research.

The model was implemented by the author using Tensorflow framework (https://www.tensorflow.org/).

The training of the model was performed in continuous bag-of-words (CBOW) manner, like in [3]. For our evaluation we choose window size 8.

IV. Experiments

A. Spelling distortions

1) Corpus: Corpus 1 is corpus from ParaPhraser challenge (http://paraphraser.ru/). This corpus consists of news headings from different news agencies, which are supposed (by the means of automatic grading system) be close in terms of semantic meaning. Additionally they all tested to be close in the creation time. The corpus contains about 6000 pairs of phrases, which are labeled as -1 - not paraphrase, 0 - weak paraphrase, and 1 - strong paraphrase. For our evaluation we had taken only -1 & 1 classes, i.e. non-paraphrase and strong paraphrase. There are 4470 such pairs in the corpus.

In published works [17], [18] there are several used measures are proposed. But since the goal of this work is to compare quality of word vectors and not the classifiers, we had chosen ROC AUC score as the most independent from a classifier.

This corpus was chosen due to its known characteristics, like paraphrase type for phrase pairs. Also since these pairs are from news headings they have a few OOV and few or none spelling errors, which is very important for us since we need to be sure for the noise level (see section IV-A5).

2) Random baseline: The random baseline just reporting a random number in $[0, 1]$ interval.

3) Word2Vec baseline: For the baseline we're taken such solution: mean word2vec for known to model words for every phrase and cosine similarity between resulting vectors. The model we'd taken is adopted from RusVectores project (http://ling.go.mail.ru/dsm/en/about), [1]. The word2vec model we'd used was trained on National Corpus of Russian Language (NCRL, http://www.ruscorpora.ru/) firstly described in [7]. This model was trained using gensim software package (https://radimrehurek.com/gensim/). The authors used window size 2. Also for this solution we used the Mystem lemmatization engine (https://tech.yandex.ru/mystem/) described in [8].

4) Our solution: For our solution we also take mean vector for all the words (since in our setup there is no such thing as OOV) and cosine similarity between resulting vectors. To have a fair competition with word2vec baseline, we also trained our model on NCRL. Our solution does not demand any lemmatization or stemming. The training of our model is taking two hours on one GPU.

5) Experiment setup: The setup of the experiment is that: we're adding noise to the input phrases and producing mean vectors by described rules. The noise emulation in this experiment consists of two components:

- The probability of inserting a letter after the current one. The letters are drawn uniformly from the alphabet.
- The probability of the letter to disappear.

The both types of noise emulation are applied at the same time.

This noise setup was chosen to demonstrate the robustness against the random error, not the typical typo (letter shuffle). The robustness against the shuffling was demonstrated in [5].

For random baseline and for every noise level (except zero level) the experiment was conducted 10 times. The standard error is not exceeding 0.003.

The results are presented on the Fig. 2.

We could see that the level of noise is important characteristic of the input. The word2vec solution is highly sensitive to the noise level, and from the level of 0.14 it generates virtually the random results (due to distribution of the test results, some of them are worse than random).

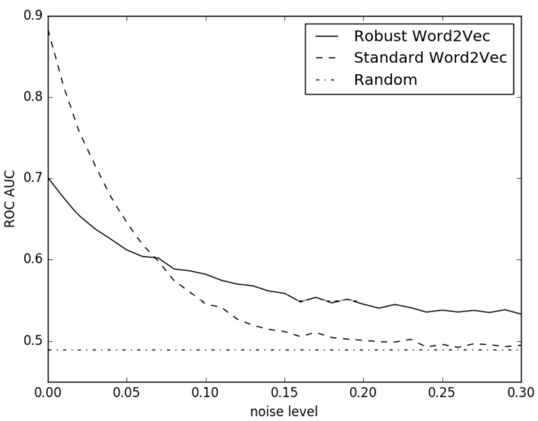

Fig. 2. The results of experiment 1

In contrast our architecture has been demonstrated the robustness to noise up to level of 0.30. The 0.30 was chosen arbitrarily, with additional consideration of that 0.30 noise level is unrealistic and too high for practical use.

It is important that proposed architecture performs better from level of 0.06 and its quality decreases steadily.

B. Out of the vocabulary words

To prove hypothesis of robustness against the OOV was performed additional experiment on the Corpus 2, originated from [14]. The corpus consists of 147 pairs of titles and short descriptions of scientific articles. The task is to determine the plagiarism. The labeling comes from three human experts, there three numbers in $[0, 1]$ interval. The metric for this corpus had been chosen mean square error (MSE) against mean of experts decisions.

This corpus was chosen for the OOV test since is consists of phrases from scientific articles from different fields. It includes a lot of words which are uncommon in the language and not typically included in the vocabularies.

1) Experiment setup: It is follow setup in section IV-A5, excepting the quality measure which is MSE in the current setup.

In the IV-B1 the results of second experiment are presented.

TABLE I. RESULTS OF TESTING ON SCIENTIFIC PLAGIARISM CORPUS

System	Quality
Random Baseline	0.213 ± 0.025
Word2Vec Baseline	0.189
Robust Word2Vec	0.232

As we could see from the table, the experiment is inconclusive, since both of word2vec baseline and our solution are close to random (and inside the margins). This could be due to size of he corpus and also the inconsistency on experts markup.

V. CONCLUSION

The robust word vector model had demonstrated abilities to be indifferent to some levels of noise. It is better from the standard widely used word2vec model with noise levels from 0.06 up to at least 0.30. It seems to be practical level, but for the future work we should try to improve our model to produce better results with less noise or without noise at all. Also we need to test our architecture on some corpus with a lot of OOV, probably Twitter.

REFERENCES

[1] A. Kutuzov and I. Andreev, Texts in, meaning out: neural language models in semantic similarity task for Russian. Proceedings of the Dialog 2015 Conference, Moscow, Russia, 2015.

[2] T. Mikolov, I. Sutskever, A. Deoras, H.S. Le, S. Kombrink, and J. Cernocky, Subword language modeling with neural networks. 2015, preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf)

[3] T. Mikolov and J. Dean. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 2015.

[4] A. Joulin et al. Bag of Tricks for Efficient Text Classification. arXiv preprint arXiv:1607.01759.

[5] K. Sakaguchi, K. Duh, M. Post, and B. Van Durme, Robsut Wrod Reocginiton via semi-Character Recurrent Neural Network. arXiv preprint arXiv:1608.02214.

[6] S. Hochreiter and J. Schmidhuber, Long short-term memory. Neural computation, 9(8):1735-1780, 1997.

[7] V.M. Andryuschenko, Konzepziya i arhitectura Mashinnogo fonda russkogo jazyka (The concept and design of the Computer Fund of Russian Language), Moskva: Nauka, 1989 (in Russian).

[8] I. Segalovich, A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine. In MLMTA (pp. 273-280), 2003, June.

[9] S. Sharoff, Classifying Web corpora into domain and genre using automatic feature identification. In Proc. of Web as Corpus Workshop, Louvain-la-Neuve, 2007, September.

[10] N. V. Neelova, Preliminary processing strings at critical measure Jaccard for improvement computation of webdocuments similarity. ZONT'09, 2009. (in Russian).

[11] Y. A. Zagorulko, N. V. Salomatina, A. S. Sery, E. A. Sidorova, and V. K. Shestakov, Detecting near-duplicates for automatically forming thematical text collections on the basis of web documents // Vestnik NSU Series: Infor-mation Technologies. - Volume 11, Issue No 4. - P. 59-70. - ISSN 1818-7900, 2013. (in Russian).

[12] A. O. Zibert and V. I. Khrustalev, Development of a System for Determining the Existence of Adoption in the Works of the Students. The Search Algorithms of Indistinct Duplicates. Universum: technical sciences 3 (4), 2014 (in Russian).

[13] P. Osipovs and A. Borisovs, Practice of Web Data Mining Methods Application. IT and Management Science. Vol.40, pp.101-107. ISSN 1407-7493, 2009 (in Russian).

[14] N. V. Derbenev, D. A. Kozliuk, V. V. Nikitin, V. O. Tolcheev, Experimental Research of Near-Duplicate De-tection Methods for Scientific Papers. Machine Learning and Data Analysis. Vol. 1 (7), 2014 (in Russian).

[15] W. Ling, T. Lus, L. Marujo, R. Fernandez Astudillo, S. Amir, C. Dyer, A.W. Black, and I. Trancoso, Finding function in form: Compositional character models for open vocabulary word representation. In Proc. of EMNLP2015.

[16] Y. Wu, M. Schuster, Zh. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, Kl. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, . Kaiser, St. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, Cl. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Trans-lation. arXiv preprint arXiv:1609.08144.

[17] E. Pronoza and E. Yagunova, Comparison of sentence similarity measures for Russian paraphrase identification. In: Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT), pp.74-82, 2015.

[18] E. Pronoza and E. Yagunova, Low-level Features for Paraphrase Identification. Proceedings of the 14th Mexican International Conference on Artificial Intelligence: MICAI 2015, Part I, Springer LNAI 9413.

[19] E. Pronoza, E. Yagunova, and A. Pronoza, Construc tion of a Russian Paraphrase Corpus: Unsupervised Para-phrase Extraction. Proceedings of the 9th Russian Sum-mer School in Information Retrieval, August 2428, 2015, Saint-Petersburg, Russia, (RuSSIR 2015, Young Scientist Conference), Springer CCIS.