Generalizable Architecture for Robust Word Vectors Tested by Noisy Paraphrases

Valentin Malykh

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

Abstract. This paper is devoted to present language independent architecture of robust word vectors. The robustness for typos is burning demand of current industry, regarding the social networks, instant messaging, etc. This architecture is designed to be indifferent to typos like switching, extra letters, and missing letters. The experiments on paraphrase corpora for three different languages are demonstrating the applicability of the proposed approach in noisy environments.

Keywords: word vectors, noise-resilient, char-aware, neural nets

1 Introduction

The problem of user input data is widely known: the typos, the errors, wrong word usage, etc. To handle this issue there exist bunch of different tools like embedded spell-checkers on websites' input forms, but severity of this problem just growing larger, especially with wide usage of mobile devices with small and/or simplified keyboards.

On the other hand word vectors had become very popular past years in variety of tasks, like text classification, paraphrase detection, sentiment analysis, etc. But to use this word vectors the user input should be cleared from the noise. To address this issue we present the novel architecture of word vectors proof to specific type of noise - the missing or surplus letters in words.

To demonstrate generality of proposed approach there were chosen languages from different types: English as (almost) analytical, Russian as synthetic flexive, and Turkish as synthetic agglutinative. The task to test the approach against was chosen to be paraphrase identification, since in this task there is a natural metric - paraphrase or not for every pair of sentences. The addition of some noise to these pairs is enabling us to compare different architectures on noise robustness.

The formal contribution of the paper is:

- Introduction of robust to typos word vector architecture.
- Results of testing on Russian Paraphrase Corpus.
- Results of testing on Microsoft Research Paraphrase Corpus.
- Results of testing on Turkish Paraphrase Corpus.

2 Related work

Some of results presented in this work have previously been presented as workshop paper [Malykh16]. This paper is broadening the previously presented results by additional language (Turkish) and additional experiments on other languages.

The presented architecture is based on previous works of Tomas Mikolov [Mikolov13], [Joulin16]. Both of the mentioned works are lacking the support of out of vocabulary (OOV) words, which could be the issue with noisy input. The group where belongs authors of previously mentioned works also proposed another approach [Bojanowski16], where the issue of OOV words has been resolved by composing word vector by summation of vectors for n-grams, from which word letter representation consists of. Our approach differs in a that way, that our model creates embeddings of the words on-the-fly, basing only on their letter representation, so we have no explicit vocabulary in our model.

Close idea was presented in corrupted word reconstruction task for english language in the work [Sakaguchi16], where the authors demonstrate stable recognition of vocabulary words. Our approach is using related initial word representation BME.

2.1 BME

The Begin-Middle-End (BME) representation is related to Begin-Intermediate-End (BIE) representation from work [Sakaguchi16]. The BME representation is broadening BIE representation by addition of three instead of one initial and ending characters. This is more suitable for languages with rich morphology, like Russian or Turkish: for example, in Russian language an affix has average length of 2.54 [Polikarpov07].

2.2 LSTM

Our approach is based on Long Short-Term Memory cells described in original paper [Hochreiter97].

$$
\begin{aligned}
\mathbf{g}^u &= \sigma(\mathbf{W}^u * \mathbf{h}_{t-1} + \mathbf{I}^u * x_t) \\
\mathbf{g}^f &= \sigma(\mathbf{W}^f * \mathbf{h}_{t-1} + \mathbf{I}^f * x_t) \\
\mathbf{g}^o &= \sigma(\mathbf{W}^o * \mathbf{h}_{t-1} + \mathbf{I}^o * x_t) \\
\mathbf{g}^c &= \tanh(\mathbf{W}^c * \mathbf{h}_{t-1} + \mathbf{I}^c * x_t) \\
\mathbf{m}_t &= \mathbf{g}^f \odot + \mathbf{g}^u \odot \mathbf{g}^c \\
\mathbf{h}_t &= \tanh(\mathbf{g}^o \odot \mathbf{m}_{t-1})
\end{aligned}
\tag{1}
$$

here σ is the logistic sigmoid function, $\mathbf{W}^u, \mathbf{W}^f, \mathbf{W}^o, \mathbf{W}^c$ are recurrent weight matrices and $\mathbf{I}^u, \mathbf{I}^f, \mathbf{I}^o, \mathbf{I}^c$ are projection matrices. The $\mathbf{u}$, $\mathbf{f}$, and $\mathbf{o}$ are denoting update, forget, and output gates in LSTM cell respectively. And $\mathbf{c}$ denotes memory cell related variables.

The main idea of usage of recurrent neural nets is to exploit their memorising ability for the context, since we are supposing that word meaning is highly correlated

with meanings of the surrounding words, following so called distributional hypothesis, which for the best of author's knowledge firstly appeared in [Rubenstein65]. This relates our model to word2vec approach, which is also heavily relies on context in creation of word vectors.

2.3 Corpora

For the Microsoft Research Paraphrase Corpus has a lot of previous work published. Few of them should be mentioned due to their word vector usage: additive composition of vectors and cosine distance achieved 0.73 accuracy in 2014 [Milajevs14] and recursive neural nets using syntax-aware multi-sense word embeddings achieved 0.78 accuracy in 2015 [Cheng15]. For the relatively full list of works on this corpus we're referring the reader to ACL website[1].

For the Russian Paraphrase Corpus there are two available works: [Loukashevich17] and [Pronoza16]. The latter paper is devoted to construction the corpus and has no presented baseline method for the pask of paraphrase identification itself. And the former work is using SVM methods for the task of paraphrase detection with a result in comparable (two class, non-standard) track of 0.81 of F1 measure.

Surprisingly there is no previously published works on the Turkish Paraphrase Corpus, despite that it is partially available for quite a time to this day.

Also it should be explicitly stated that proposed model does not pretend to be compared in the paraphrase detection task, this task is used to demonstrate the robustness to noise property of presented word vectors.

3 Architecture

A few words should be spoken about the BME representation. B part of the representation consists of one-hot encoding for first three letters, E part consists of one-hot encoding for last three letters, and M part is the sum of one-hot encoded vectors for all the letters in the word. Finally this representation is used as initial input for our model. The graphical representation of BME representation is presented in the bottom of figure 2, where the while architecture is also presented.

The model itself consists of two Fully-Connected (FC) layers and three LSTM layers. The first FC layer is used as mapping from input BME representation to fixed-size vector. This vector is fed to LSTM layers, which are responsible for handling the context in the training. The top layer of the model is also FC, which produces final fixed-size vector used as embedding for target word.

The training is followed the procedure for continuous bag-of-words (CBOW) proposed in [Mikolov13]. I.e. the model is fed by window size of surrounding words, and is supposed to produce a vector for target word. The target word vector is then compared to context words vectors and some distant word vectors by the means of cosine similarity. The visual representation of training process is given at figure 1.[2]

[1] `https://aclweb.org/aclwiki/index.php?title=Paraphrase_Identification_(State_of_the_art)`

[2] This figure is taken from `Tensorflow.org`.

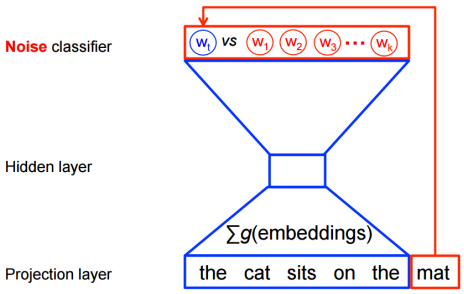

Fig. 1. Negative Sampling

3.1 The Negative Sampling

Negative Sampling technique is used in the CBOW training process for speedup and we also use it. Originally, the negative sampling comes from work [Smith05], but we are using definition of negative sampling according to [Mikolov13]:

$$L(x) = log(\sum_{i \in C} e^{-s(x,w_i)}) + log(\sum_{j \notin C} e^{s(x,w_j)}) \tag{2}$$

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. In our model it is cosine similarity of word vectors produced by the output layer of the network.

For our evaluation we chose window size as 8 following [Pennington14]. The three layers architecture shown best results in our experiments, which is typical for language modelling tasks, where number of layers is varied from one [Sundermeyer12] to four [Sutskever14]. The model was implemented on Tensorflow framework [Abadi16].

4 Experiment Setup

The conducted experiments are supposed to demonstrate the robustness to noise of proposed architecture in contrast to standard approach presented in [Mikolov13]. To achieve this goal the following setup was created:

- we measure ROC AUC of prediction of paraphrase (the true class consists of true paraphrases);
- prediction is based directly on cosine similarity between vectors for phrases;
- the vectors for phrases are computed by averaging vectors for (known) words of this phrase;
- we compare the standard word vectors for particular language and the proposed architecture trained on the same corpus;
- we are adding the noise to the input data and comparing the sensitivity to the noise level.

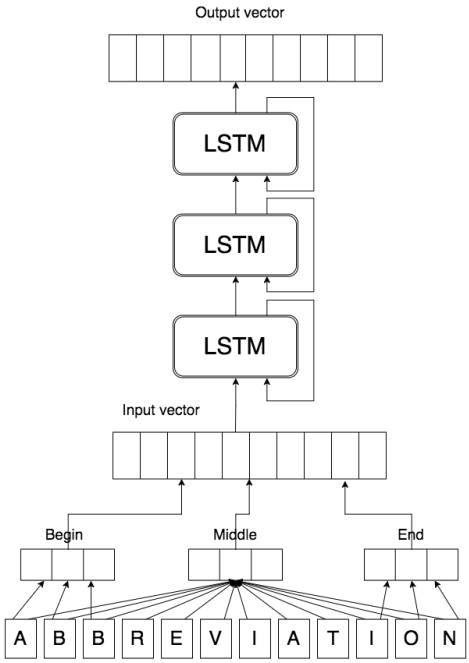

Fig. 2. Model graphical representation

Also we provide random baseline, which for the chosen measure is always close to 0.5.

The noise emulation in this experiment setup 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. The noise level mentioned below is always meant to have both probabilities to be at specified value.

This noise setup was chosen to demonstrate the robustness against the random error, i.e. unintentionally added extra letter or missed letter in a word, not the typical typo (letter shuffle), since the robustness against the shuffling was demonstrated in [Sakaguchi16].

We are comparing the solutions in range $[0.0, 0.30]$. The 0.30 was chosen arbitrarily, with additional consideration of that 0.30 noise level is unrealistic and too high for practical use. 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.

5 Experiment on Russian language

5.1 Corpus

The corpus is described in [Pronoza16]. 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.

5.2 Random baseline

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

5.3 Word2Vec baseline

For the standard word2vec baseline we're taking the model adopted from RusVectores project[3], [Kutuzov15]. The word2vec model we've used was trained on Russian National Corpus (RNC)[4] firstly described in [Andryuschenko89]. Also for this solution we'd used the Mystem lemmatization engine[5] described in [Segalovich03]. We are averaging all and only the known to model lemmatized words vectors, i.e. the unknown words are ignored.

5.4 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. By design our solution does not demand any lemmatization or stemming. We also trained our model on RNC.

5.5 Results on Russian language

The results are presented on the figure 3.

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). In contrast our architecture has been demonstrated the robustness to noise up to level of 0.30. It is important that proposed architecture performs better from level of 0.06 and its quality decreases steadily.

[3] http://rusvectores.org/

[4] http://www.ruscorpora.ru/

[5] https://tech.yandex.ru/mystem/

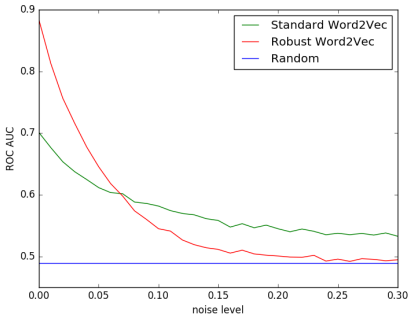

Fig. 3. The results of the experiment on Russian language

6 Experiment on English language

6.1 Corpus

The corpus is Microsoft Research Paraphrase Corpus[6]. This corpus consists of 5800 pairs of sentences which have been extracted from news sources on the web and provided with human annotations indicating whether each pair captures a paraphrase/semantic equivalence relationship.

6.2 Random baseline

The random baseline again just reporting a random number in $[0, 1]$ interval.

6.3 Word2Vec baseline

For the standard word2vec baseline here we're taking the model trained on Reuters corpus [Lewis04] by the gensim software package[7]. For this solution we'd used the Snowball stemmer described in [Porter01]. The model was trained for 500 iterations with min count value set to 2. In the testing stage we are averaging all the known to model stemmed words vectors.

6.4 Word2Vec baseline 2

For the reference also we're providing the result of testing in this setup the other word2vec model. It is Google News word vectors model, available online[8]. To the best of our knowledge it is the largest available model for English language. It was trained

[6] It is available from here: `https://www.microsoft.com/en-us/download/details.aspx?id=52398`

[7] `https://radimrehurek.com/gensim/models/word2vec.html`

[8] `https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing`

on corpus of 3 billion words, and has 3 million tokens. Unfortunately, the corpus on which is was trained is not publicly available so we could not compare to it directly. For this model we do not use lemmatization, since it contains the word forms for the majority of the words.

6.5 Our solution

For our solution we also take mean vector for all the words and cosine similarity between resulting vectors. We also trained our model on Reuters corpus.

6.6 Results on English language

The results are presented on the figure 4.

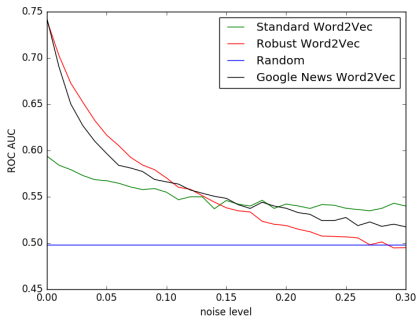

Fig. 4. The results of experiment on English language

Here we also could see that the level of noise is important characteristic of the input. But for the English language the effects are ”postpone” to the higher noise levels. The word2vec solutions for English language are not so sensitive to the noise level, and for the Reuters trained model the random level[9] comes from only 0.27. For the Google News model the random level is to the right of the 0.30 border of our plot.

The proposed architecture is performing better starting from 0.14 level for Reuters trained model and 0.16 for Google news model. Also it is important to mention that our model not only more robust than Google News one, but it is also contains far less parameters: for Google News we have 3 million by 300 vector length - about 1 billion parameters. And for the proposed architecture it is by design only squared layer width, which is 1024 for three layers in our experiments. That gives us 3 million parameters for the whole model.

[9] The level of noise in input data where the produced results are indistinguishable from random baseline results.

7 Experiment on Turkish language

7.1 Corpus

The corpus is Turkish Paraphrase Corpus (TPC), described in [Demir12]. As for this day only news part of this corpus is available[10]. It contains of 846 pairs of sentences from news sources on the web and provided with human annotations indicating whether each pair captures a paraphrase/semantic equivalence relationship.

7.2 Random baseline

The random baseline like in the previous experiments reporting a random number in $[0, 1]$ interval.

7.3 Word2Vec baseline

For the standard word2vec baseline here we're taking the model trained on "42 bin haber" (42 thousand news) corpus described in [Yildirim03]. We again used gensim software package. For this solution we'd used the Snowball stemmer for Turkish language described in [Eryigit04]. The model was trained for 500 iterations with min count value set to 2. In the testing stage we are averaging all the known to model stemmed words vectors.

7.4 Our solution

For our solution we also take mean vector for all the words and cosine similarity between resulting vectors. We also trained our model on "42 bin haber" corpus.

7.5 Results on Turkish language

The results are presented on the figure 5.

As we can see the results on TPC is not very impressing, but the main feature of noise-robustness could be noticed nevertheless.

8 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 for Russian language and 0.14 for English language 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. The difference in the excess level could be explained by the fact that Russian is flexive language with rich morphology, which is on the one hand is stable for the most words and easy to learn for

[10] It is available from here: `https://osf.io/wp83a/`

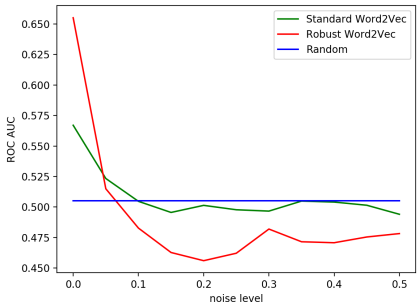

Fig. 5. The results of experiment on Turkish language

the model, and on the other hard the disruption in the flexion could lead the lemmatizer and consequently the standard word2vec model to be unable to produce vector for it. And English is language with strong analytic tendency, so the morphology of it poor, and the letters of the word are meaningful even an the end of a word. Tor Turkish language critical noise level is as low as 0.05. This seems to be unreasonable low. The possible explanations could be that for agglutinative languages the whole structure of the word is important, but more likely that available corpora is not enough for proposed approaches to demonstrate reasonable quality.

For the future work we are considering improving the architecture to achieve higher scores on small noise levels and conduct more experiments on different architecture variations.

References

Kutuzov15. Kutuzov, A. and Andreev, I., 2015. Texts in, meaning out: neural language models in semantic similarity task for Russian. Proceedings of the Dialog 2015 Conference, Moscow, Russia.

Mikolov12. Mikolov, T., Sutskever, I., Deoras, A., Le, H. S., Kombrink, S., & Cernocky, J., 2012. Subword language modeling with neural networks. preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf).

Mikolov13. Mikolov, T., and J. Dean., 2013. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems.

Joulin16. Joulin, Armand, et al., 2016. Bag of Tricks for Efficient Text Classification. arXiv preprint arXiv:1607.01759.

Sakaguchi16. Sakaguchi, K., Duh, K., Post, M. and Van Durme, B., 2016. Robsut Wrod Reocginiton via semi-Character Recurrent Neural Network. arXiv preprint arXiv:1608.02214.

Hochreiter97. Hochreiter, Sepp and Schmidhuber, Jurgen, 1997. Long short-term memory. Neural computation, 9(8):17351780.

Andryuschenko89. Andryuschenko, V.M., 1989. Konzepziya i arhitectura Mashinnogo fonda russkogo jazyka (The concept and design of the Computer Fund of Russian Language), Moskva: Nauka (in Russian).

Segalovich03. Segalovich, I., 2003, June. A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine. In MLMTA (pp. 273-280).

Smith05. Smith, Noah A., and Jason Eisner, 2005. Contrastive estimation: Training log-linear models on unlabeled data. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics.

Lewis04. Lewis, D. D.; Yang, Y.; Rose, T.; and Li, F., 2004. RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, 5:361-397

Porter01. Porter, M.F., 2001. Snowball: A language for stemming algorithms.

Milajevs14. Milajevs, D., Kartsaklis, D., Sadrzadeh, M. and Purver, M., 2014. Evaluating Neural Word Representations in Tensor-Based Compositional Settings, Proceedings of EMNLP 2014, Doha, Qatar, pp. 708-719.

Cheng15. Cheng, J. and Kartsaklis, D., 2015. Syntax-Aware Multi-Sense Word Embeddings for Deep Compositional Models of Meaning, Proceedings of EMNLP 2015, Lisbon, Portugal, pp. 1531-1542.

Demir12. Demir, S., El-Kahlout, I.D., Unal, E. and Kaya, H., 2012. Turkish Paraphrase Corpus. In LREC (pp. 4087-4091).

Yildirim03. Yildirim, O., Atik, F., Amasyali, M. F., 2003. 42 Bin Haber Veri Kumesi, Yildiz Teknik Universitesi, Bilgisayar Muh. Bolumu.

Eryigit04. Eryigit, G. and Adali, E., 2004. An Affix Stripping Morphological Analyzer for Turkish. Proceedings of the IAESTED International Conference Artificial Intelligence and Applications 2004, Innsbruck, Austria.

Pronoza16. Pronoza, E., Yagunova, E. and Pronoza, A., 2016. Construction of a Russian paraphrase corpus: unsupervised paraphrase extraction. In Information Retrieval (pp. 146-157). Springer International Publishing.

Malykh16. Malykh V., 2016. Robust Word Vectors for Russian Language. In Proceedings of AINL FRUCT Conference 2016.

Bojanowski16. Bojanowski P., Grave E., Joulin A., Mikolov T., 2016. Enriching Word Vectors with Subword Information. arXiv:1607.04606

Rubenstein65. Rubenstein, H., & Goodenough, J. (1965). Contextual correlates of synonymy. Communications of the ACM, 8 (10), 627-633.

Loukashevich17. Loukachevitch N. V., Shevelev A. S., Mozharova V. A., 2017. Testing Features and Methods in Russian Paraphrasing Task. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2017".

Polikarpov07. Polikarpov A.A., 2007. Towards the Foundations of Menzerath?s Law. In: Grzybek P. (eds) Contributions to the Science of Text and Language. Text, Speech and Language Technology, vol 31. Springer, Dordrecht

Pennington14. Pennington, J., Socher, R. and Manning, C.D., 2014, October. Glove: Global vectors for word representation. In EMNLP (Vol. 14, pp. 1532-1543).

Sutskever14. Sutskever, I., Vinyals, O. and Le, Q.V., 2014. Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).

Sundermeyer12. Sundermeyer, M., Schlueter, R. and Ney, H., 2012. LSTM neural networks for language modeling. In Thirteenth Annual Conference of the International Speech Communication Association.

Abadi16. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. and Ghemawat, S., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.