Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2017”

Moscow, May 31—June 3, 2017

EXPLOITING RUSSIAN WORD EMBEDDINGS FOR AUTOMATED GRAMMEME PREDICTION

Romanov A. V. (Aleksey_Ro@abbyy.com)

ABBYY; Moscow Institute of Physics and Technology (MIPT),
Moscow, Russia

Distributed representations of words are currently used in a variety of linguistic tasks. A specific branch of their possible applications includes automatic extraction of word-level grammatical information by formulating it as a problem of word embedding classification. In this paper, we investigate applicability of this approach to prediction of several particular classifying grammemes. We focus on animacy of Russian nouns and transitivity of Russian verbs. These categories can serve as good examples of classifying grammatical categories in the Russian language since their concrete values can hardly be predicted judging by appearance of words and morphemes that constitute them. We conduct experiments on a corpus of Russian texts from the Web with several widely used word-embedding algorithms and different parameter settings. Experimental evaluation includes the comparison of performance of several classifiers, with distributed representations being source of features for classification task. Our findings show feasibility of the approach and its potential to be implemented for solving related tasks.

Key words: natural language processing, distributional semantics, word embeddings, word-level classification, automatic corpus annotation

ПРИМЕНЕНИЕ МОДЕЛЕЙ ДИСТРИБУТИВНОЙ СЕМАНТИКИ ДЛЯ АВТОМАТИЧЕСКОГО ПРЕДСКАЗАНИЯ ГРАММЕМ

Романов А. В. (Aleksey_Ro@abbyy.com),

ABBYY; Московский физико-технический институт
(государственный университет), Москва, Россия

1. Introduction

Distributed word representations, or word embeddings, have already shown their power as a basis for efficient model training within the scope of neural-network approach in various natural language processing tasks. In addition to this primary mission, word embeddings are widely and successfully applied to development of solutions that do not necessarily use neural nets as their core component; e.g. contextual information encoded in the distributed representations can be used for word similarity estimation and in related problems.

Another potential application of word embeddings resides in automated word-level grammatical and semantical information extraction. This set of tasks is itself quite interesting for linguists: measuring the correlation between contexts of the word and its internal sense, and determining the limits of distributional approach are two questions that are still open and should be investigated broader. Moreover, such tasks can be seen as auxiliary for more complex ones. One can consider, for example, the following situation: vast amount of text available on the Web can be exploited in a variety of linguistic studies provided it is properly and fully labeled in accordance with the specific task orientation. Availability of automated word labeling methods for text corpora is thus the condition for future linguistic research.

In this paper, we investigate applicability of distributed word representations to prediction of several classifying grammemes of Russian words. In particular, we consider animacy of nouns and transitivity of verbs in the Russian language, since concrete values of these grammatical categories can hardly be predicted judging by appearance of words and morphemes that constitute them. We expect that good performance of automatic classification in the aforementioned tasks may open the way to extension of the approach into other related problems.

We propose to utilize real-valued vectors obtained from distributed representation models as features in these prediction tasks, which may lead to a scheme of grammeme prediction on a basis of insufficiently labeled corpora. We explore power of several widely used word-embedding algorithms and train models with different sets of parameters in order to achieve better performance. Additional investigation concerns testing the dimensionality reduction technique proposed recently ([12]) for enhancing word embeddings in applied tasks. Based on the experimental results, we argue that our approach is feasible for prediction of grammatical characteristics of Russian words.

2. Related work

The task of automated grammeme prediction is closely related to a more general problem of automated corpus annotation and, more specifically, to automated grammatical tagging. A classic work in the field is [9], where the authors utilize a stochastic algorithm to complete the first stage of two-staged tagging process, the second being manual correction of errors produced by the automatic stage. The method and a number of related ones ([13], [7]) are based on complex models with a large number of parameters to be tuned in order to achieve good performance; this may be an encumbrance in the case of small corpus on annotated data.

State-of-the-art techniques of automatic grammatical tagging mostly focus on overcoming the obstacle of insufficient amount of data available for model training. This is important, among other things, for developing automated tagging systems for languages lacking high-quality text resources and corpora on the Web. The authors of [6] propose to use graph-based label propagation for cross-lingual knowledge transfer and utilize the resulting labels as features in an unsupervised model. The idea is further developed in [14], where ambiguous learning approach enables effective automated transfer of tags from English corpora to corpora in other languages. In [8] several techniques for low-resource tagging are shown to be feasible.

Word embeddings have also been utilized for solving a number of morphological tasks. A work [4] proposes an architecture and an algorithm performing well in POS-tagging task without labeling data beforehand. [5] describes a model of morphologically guided embeddings, which is capable of handling tagging tasks in a semi-supervised manner by adding labeling to the training corpus.

Several works are devoted to automatic animacy prediction ([3], [1]). The methods rely on hand-crafted features obtained from annotated sources of semantic and lexical information, achieving high accuracy over 90%. The key idea of our method is, in contrast, in taking automated grammeme prediction to a competitive level by means of minimal available corpus annotation and limited amount of data. Our approach builds upon and extends the method described in [12], where the authors explore the ability of word embeddings to predict certain grammatical functions including noun animacy. In our work, we try to extend their research into the Russian language and improve the performance studying the influence of various parameters, both on the stage of distributional model training and while generating classification features.

3. Method description

3.1. Formal problem statement

We consider the task of grammeme prediction as a word-level binary classification task. In other words, we train a classifier to predict whether a word has a certain value of a grammatical category or not. In our study, which is designed to give preliminary characterization to feasibility of the approach, we do not focus on homonym disambiguation and treat each unique sequence of characters as a classification object.

Thus, the task is to build is a classifier a: $W \longrightarrow \{0,1\}$ that, on the basis of feature representation $\boldsymbol{w}$ of the word $w \in W$, would predict whether w has the grammeme g (1 class) or not (0 class). The optimal classifier a^* is found by training on the set of labeled precedents $W = \{(\boldsymbol{w}_1, y_1), \dots, (\boldsymbol{w}_n, y_n)\}$, $y_1 \in \{0, 1\}$, $i = 1 \dots n$, i.e. the process of minimization of the empirical risk $Q(a, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}[a(\boldsymbol{w_i}) \neq y_i]$:

$$a^* = \arg\min_{a} Q(a, D)$$

3.2. Grammeme choice

In order to verify the assumption that word embeddings may potentially be applied for grammeme prediction, we have chosen two of *classifying* grammemes in the Russian language, i.e. those that are intrinsically fixed for a lexeme and constant across its derived forms. Noun animacy and verb transitivity are good examples of grammatical categories whose values cannot be easily predicted judging by appearance of words and morphemes that constitute them; therefore, it is particularly interesting if distributional models of morphologically rich languages, with Russian being an example, can be a source of ready-to-use classification features.

Noun animacy basically provides distinction between nouns referring to humans (and some other biological creatures) and those referring to various inanimate objects and phenomena. In the Russian language, it is often necessary to have information about the noun animacy in order to inflect the noun correctly. Consider, for example, the plural accusative of an animate noun *мальчик* (“boy”) — *мальчиков*, matching the plural genitive, and the plural accusative of an inanimate noun *пальчик* (“little finger”) — *пальчики*, matching the plural nominative. This rule generalizes to other animate and inanimate nouns. Automated animacy/inanimacy prediction is thus useful for morphological analysis and phrase generation as well.

Verb transitivity is a property of a verb to take direct objects, a special case of a more general notion of valency. In Russian, like in English, this category is expressed syntactically, i.e. it is possible to identify an intransitive verb by attempting to supply it with an appropriate direct object but not by judging by its morphological markers. Transitivity used to be believed to be a binary characteristics of a verb; now, no verbs are mainly seen as “absolutely transitive” but rather “more often occurring in texts with a transitive role”. Intransitive verbs, however, never appear in phrases with direct objects, and this fact enables the task of transitivity prediction to be considered as a binary classification task.

3.3. Features

The main idea of the method is to use pre-trained word embeddings “as is” as features for classification. The advantages of this approach are its simplicity and scalability onto related problems. We tested different parameter configuration sets of distributional model training to study the effect of the configuration choice on the overall performance.

Additional experiments were devoted to:

- enriching feature space with auxiliary per-word information provided in distributional models;
- transformations of word embeddings aimed at obtaining more informative representations.

4. Experiments

4.1. Text Data

Distributional models were trained on a collection of Wikipedia articles in the Russian language (1.3M articles and 100M tokens on the whole). The text was split into sentences and lowercased. Non-Cyrillic words and punctuation marks were removed. All digits and numbers were replaced by a single special token. We lemmatized the corpus with *pymorphy2,* a Python package.

A list of Russian nouns and verbs labeled respectively with animacy and transitivity tags was obtained from the *pymorphy2* package as well. Overall, 12K verb (7.5K transitive) and 121K noun (47K animated ones, including proper names) lemmas were extracted and prepared for classification.

4.2. Distributional model training

Among frameworks offering opportunities of training distributional models, *gensim* and *fasttext* were chosen, with word2vec and FastText being the models providing word embeddings.

Word2vec continuous-bag-of-words [10] models were trained with a set of default parameters. We tried different (symmetric and asymmetric) configurations of context windows in order to test a hypothesis that smaller context windows induce word embeddings with greater grammeme prediction power. We also varied the dimension of embeddings, as higher dimension leads to better performance in a number of related tasks.

FastText model [2] is a promising extension of word2vec, designed to construct vectors not only for words but also for character N-grams that constitute them. This way, the words that have some N-grams in common get representations that are more similar to each other. Another useful feature of this approach is its ability to predict vectors for unseen words. FastText models of various dimensions were trained as well.

4.3. Experimental results

In our experiments, we compared three types of classifiers: Support Vector Machine (SVM), Random Forest (RF) and Multi-Layer Perceptron (MLP). The metrics to be measured was weighted F1 score (average F1 by classes weighted by support). Hyperparameters of classifiers (regularization constants, number of trees and hidden layers, respectively) were tuned to obtain the best performance on 5-fold stratified cross-validation scheme.

Window size and dimension effect

We selected several word2vec and FastText models to study the effect of different training parameters on classification performance (i.e. on weighted F1 scores):

- word2vec, 250-dimensional vectors, context window: 5 words before + 5 words after the word;

- word2vec, 500-dim, context: 5 + 5;
- word2vec, 500-dim, context: 2 + 2;
- word2vec, 500-dim, context: 0 + 3;
- word2vec, 500-dim, context: 3 + 0;
- FastText, 250-dim, context: 5 + 5;
- FastText, 500-dim, context: 5 + 5;
- FastText, 250-dim, context: 5 + 5, prediction of vectors for unknown words;
- FastText, 500-dim, context: 5 +5, prediction of vectors for unknown words.

It is worth noting that in the last two cases, the support for classification task is much greater in size than that of other cases: in such a setting we can obtain vectors for all the words we are willing. On the contrary, in the first cases, we conduct classification on the set of words that occur in the Wikipedia corpus. Thus, in the last two cases we have 121K nouns (54K in the first cases) and 12K verbs (6.4K in the first cases) for classification. Therefore, classification performance may differ in these two cases, but it resembles the situation of automated corpus tagging to a greater extent. The results are given in Table 1.

Table 1. Performance of different distributional models and classifiers

	Transitivity			**Animacy**		
Model	**SVM**	**RF**	**MLP**	**SVM**	**RF**	**MLP**
word2vec, 250, 5+5	0.818	0.767	0.810	0.880	0.877	0.871
word2vec, 500, 5+5	0.833	0.748	0.831	**0.888**	0.870	0.873
word2vec, 500, 2+2	0.657	0.556	0.644	0.659	0.653	0.626
word2vec, 500, 3+0	0.628	0.550	0.624	0.634	0.621	0.601
word2vec, 500, 0+3	0.631	0.558	0.620	0.631	0.615	0.612
FastText, 250	0.853	0.827	0.862	0.845	0.834	0.825
FastText, 500	0.859	0.834	**0.868**	0.848	0.820	0.830
FastText, 250, prediction	0.828	0.819	0.856	0.790	0.799	0.805
FastText, 500, prediction	0.840	0.825	0.862	0.797	0.789	0.811

The results show that the models that incorporate knowledge about character N-grams of the word are more powerful for transitivity prediction, and their features have non-linear dependencies. However, linear methods performed better on classic word2vec models for animacy prediction task, since noun animacy in the Russian language has weak correlations with the words' appearance. Overall, the results are comparable with those achieved in [12] for tasks in other languages.

Enriching embeddings with auxiliary training information

While training word2vec models, one can access both W_{in} and W_{out} matrices. The former is mainly used as the word embedding source, and the latter rarely comes into use in practical tasks. We investigated the applicability of both types of vectors to our problem in the following manner. Three sets of classification features were extracted from a 500-dimensional word2vec model with the default 5+5 context window:

- W_{in} rows—as in the abovementioned experiment;
- W_{out} columns;
- stacked W_{in} rows and W_{out} columns.

Table 2 shows the results of this study.

Table 2. Performance on various matrix-based features

Features	**Transitivity**			**Animacy**		
	SVM	**RF**	**MLP**	**SVM**	**RF**	**MLP**
W_{in} rows	0.833	0.748	0.831	0.888	0.870	0.873
W_{out} columns	0.848	0.755	0.844	0.886	0.871	0.865
stacked $W_{in} + W_{out}$	**0.852**	0.750	0.841	**0.893**	0.876	0.883

Reducing word embeddings by the main PCA components

We applied the trick proposed in [11], which is said to create more powerful embeddings performing better in a bunch of tasks. The idea of the trick is to center embeddings, reducing them by their average vector, apply PCA and remove *D* most informative components from the vectors afterwards. We studied classification performance with several values of *D* on the same distributional model as in the experiment described in the previous paragraph. The results are available in Table 3.

Table 3. Effect of reduction by PCA components on overall performance

D	**Transitivity**			**Animacy**		
	SVM	**RF**	**MLP**	**SVM**	**RF**	**MLP**
—	**0.833**	0.748	0.831	**0.888**	0.870	0.873
2	0.822	0.735	0.816	0.795	0.822	0.864
3	0.826	0.733	0.822	0.787	0.818	0.862
5	0.821	0.737	0.811	0.739	0.782	0.851
10	0.815	0.720	0.811	0.729	0.749	0.839

4.4. Discussion

Performance achieved in the experiments is sufficient to claim feasibility of the approach. Surprisingly, smaller values of context window size used in word2vec training lead to significant drop in classification performance. This effect allows to assume that grammemes of animacy and transitivity are more closely related to broader, semantic contexts of a word than to narrower, syntactic ones.

FastText showed better performance than word2vec did in transitivity prediction task. This can be attributed to the fact that some transitive (or, probably, intransitive) verbs in the Russian language share certain character N-grams. Thus, a model that assigns closer vectors to similarly looking words is supposed to perform better in this task. At the same time, the problem of animacy prediction is solved better by word2vec models, since animate nouns cannot be distinguished from inanimate ones judging by their appearance. However, FastText predictive power for unseen words still makes it a good choice for automated corpus annotation.

It is worth noting that in some of the task settings non-linear classification models did not achieve higher performance than linear ones (especially on word2vec vectors). Another interesting fact is that W_{out} matrix of word2vec models is sometimes even more informative in classification tasks than W_{in} containing input word embeddings. Removing main PCA components did not drop significantly the quality of classification, but there was no increase as well.

We have also analyzed errors produced by classifiers in both tasks and can group them into the following categories:

- polysemantic words (e.g. изменить (transitive "to change" or intransitive "to cuckold")) and homonyms (e.g. везти (transitive "to carry" or intransitive "to be lucky", *барак* ("a barrack") and Барак ("Barack"));
- rare words lacking occurrences in the training corpus (e.g. дефилировать "to sashay", хлебопашец "a sodbuster");
- transitive verbs frequently used in sentences without a direct object (e.g. петь "to sing");
- inanimate proper names that can be seen as human names (e.g. Бредфорд "Bradford").

Overall, the majority of classifying errors appear to arise due to labeling problems and, to a lesser degree, due to the limited amount of data for training the distributional model.

5. Conclusion

We propose a method of automatic grammeme prediction, which is based on word embedding classification and does not rely on corpora annotation. Preliminary findings show performance that is competitive with systems developed on handcrafted features.

As the future work, we plan to achieve better classification quality and extend the method to handle other grammatical categories than those described in this paper. Improvements can be made by preparatory homonym disambiguation or training on unlemmatized text with subsequent pooling on word forms for lemmas (which can be done by several promising schemes) during the classification stage. Another interesting course of future study includes extension of our approach to other languages.

Acknowledgements

We are grateful to Eugene Indenbom and Konstantin Zuev for their valuable comments and suggestions.

References

1. *Bloem J., Bouma G.* (2013), Automatic animacy classification for Dutch, Computational Linguistics in the Netherlands Journal, Vol. 3, pp. 82–102.
2. *Bojanowski P., Grave E., Joulin A., Mikolov T.* (2016), Enriching word vectors with subword information, arXiv preprint arXiv:1607.04606.
3. *Bowman S. R., Chopra H.* (2012), Automatic animacy classification, Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, Montreal, pp. 7–10.
4. *Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K., Kuksa P.* (2011), Natural language processing (almost) from scratch, Journal of Machine Learning Research, Vol. 12, pp. 2493–2537.
5. *Cotterell R., Schütze H.* (2015), Morphological Word-Embeddings, HLT-NAACL, pp. 1287–1292.
6. *Das D., Petrov S.* (2011), Unsupervised part-of-speech tagging with bilingual graph-based projections, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Portland, pp. 600–609.
7. *Federici S., Pirrelli V.* (1994), Context-sensitivity and linguistic structure in analogy-based parallel networks, Current Issues in Mathematical Linguistics, pp. 353–362.
8. *Garrette D., Baldridge J.* (2013), Learning a Part-of-Speech Tagger from Two Hours of Annotation, HLT-NAACL, pp. 138–147.
9. *Marcus M. P., Marcinkiewicz M. A., Santorini B.* (1993), Building a large annotated corpus of English: The Penn Treebank, Computational linguistics, Vol. 19(2), pp. 313–330.
10. *Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J.* (2013), Distributed representations of words and phrases and their compositionality, Advances in neural information processing systems, pp. 3111–3119.
11. *Mu J., Bhat S., Viswanath P.* (2017), All-but-the-Top: Simple and Effective Post-processing for Word Representations, arXiv preprint arXiv:1702.01417.
12. *Qiu P. Q. X., Huang X.* (2016), Investigating language universal and specific properties in word embeddings.
13. *Schmid H.* (2013), Probabilistic part-of-speech tagging using decision trees, New methods in language processing, Routledge, p.154.
14. *Wisniewski G., Pécheux N., Gahbiche-Braham S., Yvon F.* (2014), Cross-Lingual Part-of-Speech Tagging through Ambiguous Learning, EMNLP, Vol. 14, pp. 1779–1785.