ВЕКТОРНЫЕ МОДЕЛИ И ВСПОМОГАТЕЛЬНЫЕ МЕТОДЫ ДЛЯ ОПРЕДЕЛЕНИЯ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ СЛОВ РУССКОГО ЯЗЫКА

Лопухин К. А. (kostia.lopuhin@gmail.com)
ЧТД, Москва, Россия

Лопухина А. А. (nastya-merk@yandex.ru)
Институт русского языка им. В. В. Виноградова РАН, Москва, Россия

Носырев Г. В. (grigorij-nosyrev@yandex.ru)
Яндекс, Москва, Россия

Ключевые слова: семантическая близость, ассоциации, машинное обучение, семантические векторы, векторные модели

THE IMPACT OF DIFFERENT VECTOR SPACE MODELS AND SUPPLEMENTARY TECHNIQUES ON RUSSIAN SEMANTIC SIMILARITY TASK[1]

Lopukhin K. A. (kostia.lopuhin@gmail.com)
Chtd, Moscow, Russia

Lopukhina A. A. (nastya-merk@yandex.ru)
V. V. Vinogradov Russian Language Institute, Russian Academy of Sciences, Moscow, Russia

Nosyrev G. V. (grigorij-nosyrev@yandex.ru)
Yandex, Moscow, Russia

This paper presents a system for determining semantic similarity between words that was an entry for the Dialog 2015 Russian semantic similarity competition. The system introduced is primary based on word vector models, supplemented with various other methods, both corpus- and dictionary-based. In this paper we compare performance of two methods for building word vectors (word2vec and GloVe), evaluate how performance

[1] Работа выполнена при частичной финансовой поддержке Программы фундаментальных исследований Президиума РАН «Историческая память и российская идентичность» и гранта РГНФ №13-04-00307а.

varies on different corpus sizes and preprocessing techniques, and measure accuracy gains from supplementary methods. We compare system performance on word relatedness and word association tasks, and it turns out that different methods have varying relative importance for these tasks.

Key words: semantic similarity, associations, machine learning, semantic vectors, vector space model

1. Introduction

Semantic similarity is a measure of closeness of word meanings that can be represented as a number on some scale. The notion of semantic similarity includes different types of semantic relations: synonyms, hyponyms and hypernyms ("*свист*" (*whistle*), "*хрип*" (*wheeze*), "*стрекотня*" (*chirr*) and "*звук*" (*sound*); "*жвачка*" (*chewing gum*) and "*продукт*" (*product*); "*муж*" (*husband*) and "*мужчина*" (*man*)) and semantic associations, that link words by connotations ("*актер*" (*actor*) and "*игра*" (*performance*), "*грим*" (*make-up*); "*Айвазовский*" (*Ayvazovsky*) and "*маринист*" (*painter of seascapes*)). The last term, association, is loosely defined, and can range from pairs that average speaker might consider synonyms, to rather distant concepts.

Semantic similarity is an important building block in more complex natural language processing tasks, such as sentence and text similarity, machine translation [Mikolov et al 2013a], query expansion [Voorhees 1994], etc.

There are several approaches for determining semantic similarity: based on dictionaries, ontologies or machine learning. Synonym dictionaries are compiled manually and reflect human understanding of synonymy, but contain only one type of semantic relations and are deemed to be incomplete. Ontologies include hyponym relations and allow searching for the shortest connection between words or concepts, but also suffer from low recall. Machine learning solves low recall problem by training models on big corpora, but human understanding of semantic similarity is hard to model correctly.

2. Russian Semantic Similarity Evaluation (RUSSE)

Most approaches to semantic similarity were implemented and evaluated primarily in English, and there were no systematic evaluations of semantic similarity models for Russian until the RUSSE competition and workshop, held for Dialogue 2015 conference [Panchenko et al 2015][2]. Semantic similarity was measured on the following tracks:

- Human judgements track (hj): word similarity assessed by Russian native speakers.
- Relatedness track (rt): relations sampled from RuThesLite Tesaurus.
- First association track (ae): relations sampled from Russian Associative Thesaurus.
- Second association track (ae2): relations sampled from Sociation.org online experiment.

[2] http://russe.nlpub.ru

Evaluation metric for human judgements track was Spearman's rank correlation, and AUC under the ROC curve for the other tracks.

In this paper we describe a system that was an entry for RUSSE competition and analyse its performance.

3. Word vector models

One of the most widely used machine learning approaches for determining semantic similarity is building word vector models from large corpora and using distance in this vector space as a measure of semantic similarity. Word vector models represent each word as a low-dimensional (50–1,000 components) vector, built based on words contexts in corpus.

These models are often called semantic vector space models, because components of the vectors exhibit semantic properties [Mikolov et al 2013b]: for example, the difference between vectors for "*king*" and "*queen*" is very close to the difference of "*man*" and "*woman*". The most useful property for our task is that semantically similar words have similar vectors. Word similarity is usually defined as a cosine of the angle between two word vectors (cosine similarity).

We decided to use word vectors for modelling word similarity because they are known to perform well for this task [Mikolov et al 2013c] and are straightforward to implement. Another benefit is that they give continuous similarity measure out of the box, which is useful for hj track and simplifies augmentation with other models.

There are several different algorithms for computing word vectors. In this paper we evaluated word2vec skip-gramm algorithm [Mikolov et al 2013c] using gensim implementation and GloVe [Pennington et al 2014] algorithm using reference implementation. Some studies [Shi at al 2014] suggest that although these two algorithms have quite different numerical formulation, their optimization objectives are similar. But in practice these algorithms produce vectors which quality very much depends on the task at hand. In our case it turns out that word2vec models perform better on all tracks, as we can see in the following table:

Table 1. Comparison of word2vec and GloVe models

	word2vec	**GloVe**	**ratio**
hj	**0.76254**	0.66537	14.60%
rt	**0.92277**	0.90128	2.38%
ae	**0.95525**	0.95427	0.10%
ae2	**0.98354**	0.97723	0.65%

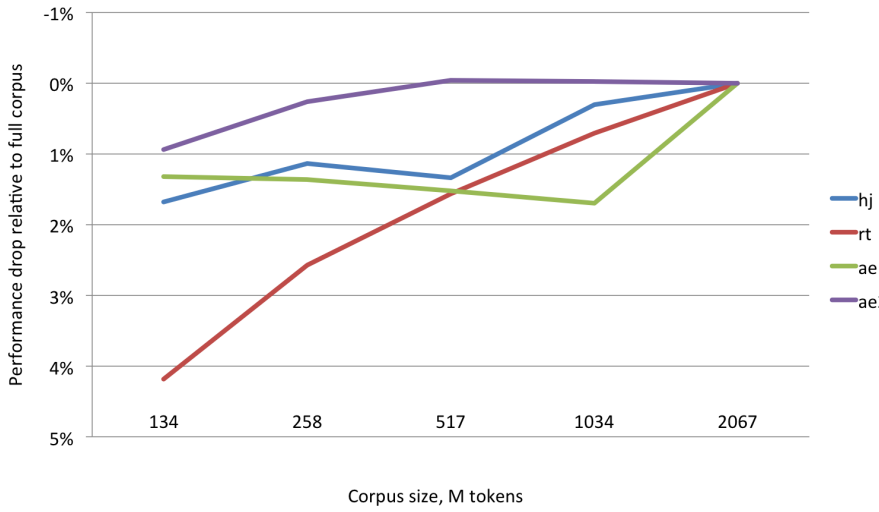
Note that we did not do extensive meta-parameter optimization: we used window size 10, and vector size 300, leaving other parameters at default values. We used cosine similarity for both methods, although there might be a better measure, especially in the case of GloVe.

4. Importance of corpora size and preprocessing

Quality of corpus-based models usually depends on the size and quality of the corpus and preprocessing techniques. Knowing that, we used the biggest corpus we could get at the time, by combining several separate corpora: ruwac[3] (1,268 M tokens), lib.ru (624 M tokens), and Russian Wikipedia[4] (176 M tokens). Even for such a large corpus rare words were still a problem, so we used a rather low frequency cutoff of 10, which gave us vocabulary size 844,530. In order to measure how model quality depends on corpus size, we compared final system performance on randomly sampled sub-corpora of various sizes. Results are represented as a table, that shows performance loss relative to full corpus.

Table 2. Impact of corpus size

rel. size	0.5	0.25	0.125	0.065
hj	0.31%	1.34%	1.13%	1.68%
rt	0.71%	1.57%	2.58%	4.19%
ae	1.70%	1.52%	1.36%	1.32%
ae2	−0.02%	−0.04%	0.27%	0.94%



This suggests that increasing corpus size might be worthwhile for most tracks.

Model and corpus building time should also be considered. We needed 4 hours for corpus preprocessing and 8 hours for model training using 8 cores for the full corpus.

[3] http://corpus.leeds.ac.uk/tools/ru/ruwac-parsed.out.xz

[4] https://s3-eu-west-1.amazonaws.com/dsl-research/wiki/wiki-ru-noxml.txt.bz2

Besides basic preprocessing (getting rid of html markup, short sentences, etc.) we also experimented with using lemmatizer as a preprocessing step. On one hand, we lose valuable grammatical information here, so the quality of the vectors might decrease. On the other hand, lemmatizing helps mitigate low frequency words problem and allows comparing lemmas and not word forms.

As we see in the following table, lemmatizing hugely influences human judgements track performance and is also important for other tracks.

Table 3. Impact of lemmatization

	lem	no lem	ratio
hj	**0.76254**	0.60014	27.06%
rt	**0.92277**	0.86150	7.11%
ae	**0.95525**	0.91079	4.88%
ae2	**0.98354**	0.94570	4.00%

So far we have described the base of our method: word vector model built with word2vec on a large corpus with lemmatization.

5. Supplementary models and sources

The first association track (ae) contained a certain number of high frequency bigrams, like "*человек*" (*man*) and "*амфибия*" (*amphibian*) or "*время*" (*time*) and "*не ждет*" (*does not wait*), so **bigram model** was used to supplement the word vector model. Bigram model was built from the same corpus that was used for word vectors, but with stop words (prepositions, conjunction, etc.) removed. In order to convert bigram score into [0, 1] range, we used ad hoc normalized PMI: $log\,(max(1{,}1 + PMI))\,/\,2$. Bigram model was used only on ae and ae2 tracks, with ae gaining 7.49%, and ae2 just 0.89%. On hj and rt tracks performance with bigram model dropped significantly, up to 7.84% for hj track.

Analysis of errors on training datasets revealed two major sources of errors:

1. Low frequency words: some words, especially in rt training dataset, were never seen in the corpus, for example "*автохтонка*" (*woman-indigene*), "*магометанство*" (*Mohammedanism*).
2. High frequency words having common semantic components, but not synonyms or hyponyms: such words are often used in similar contexts, and thus have high similarity according to word vector model, for example "*собрат*" (*brother*) and "*предшественник*" (*predecessor*) or "*блузочка*" (*blouse*) and "*платьице*" (*dress*).

Such errors are hard to resolve with just word vector and bigram models, so we introduced a number of supplementary models and sources to overcome them:

- synonyms database
- prefix database
- orthographic similarity model
- secondary orthographic similarity model
- hyphen handling

They are described in more detail below.

Synonyms database is a database of synonyms compiled from five dictionaries[5] by students and researchers from the Higher School of Economics. Synonyms are given for 43,679 words, constituting 135,134 pairs, as many words have several synonyms. If word A had synonyms S1..Sn, then pairs (Si, Sk) were also considered synonyms, but with a lower weight—such extension gave 1 556 374 word pairs. Recall on rt train dataset is 7.64%, with 0.63% false positives. False positive ratio is the number of cases where model considered words as similar, divided by the total number of predictions by the model (and not by the total number of pairs in training set). Gain from this model (how much precision dropped when dropping this model from the final method) ranged from 1.53% to -0.03% on different tracks within the test dataset, with maximum gain on human judgements track. Synonym databases should be used if possible, as they are very easy to incorporate into existing models and increase performance without significant drawbacks.

Prefix database is a list of greek and latin prefixes, extracted from "The anatomy of terms. 400 derivation elements from Latin and Greek" [Bykov 2008], that give strong contribution to the word meaning, like "auto", "aero", etc. If two words shared such prefix, they were considered similar. This model was added to overcome low frequency words problem for pairs such as "*авиаконцерн*" (*aviaconcern*) and "*авиаконсорциум*" (*aviaconsortium*). Recall on rt train dataset is 0.82%, with 0.53% false positives. The only track that gained a little from this model was rt track, with 0.15% gain. Despite such a low gain, we still used it in the competition, but generally this model seems to be of little use due to very low recall.

Orthographic similarity model measures similarity in spelling, and improves handling of low frequency word pairs like "*автохтон*" (*indigene*) and "*автохтонка*" (*woman-indigene*). More precisely, it searches for a longest common beginning or ending, and then gives similarity in [0, 1] range based on its length and lengths of compared words. It is especially useful in case of two cognate words of different gender ("*агроном*" (*agriculturist*) and "*агрономша*" (*woman-agriculturist*)), or usage of some rare stem ("*авангардность*" (*vanguardness*) and "*авангардизм*" (*avant-gardism*)). Such cases could also be handled by stemming.

Recall for this model on rt train dataset is 6.40%, with 1.76% false positives. Gain from this model, combined with secondary similarity model is up to 1.55% for rt track. Due to our definition of gain, we can not measure the gain without secondary similarity model, but we can compare the gain against pure word2vec model: it is 0.58% for rt track.

Secondary orthographic similarity model extends the gains in orthographic similarity model to more cases. For example, words "*водитель*" (*driver*) and "*автолюбительница*" (*woman-motorist*) are not considered similar by the model, because "*автолюбительница*" (*woman-motorist*) is absent from the word vector model. But we have a pair "*водитель*" (*driver*) and "*автолюбитель*" (*motorist*), where words are similar according to word vector model, and a pair "*автолюбитель*" (*motorist*) and "*автолюбительница*" (*woman-motorist*), where words are similar according

[5] http://web-corpora.net/synonyms

to orthographic similarity model. Secondary model can thus infer that the original pair "*водитель*" (*driver*) and "*автолюбительница*" (*woman-motorist*) has high similarity, namely the multiplication of two other similarity measures. Recall on rt train dataset is 7.20% (that is, ratio of pairs that gained higher similarity measure). Gain from this model is 1.00% for rt track.

Hyphen handling was added to improve similarity assessment of words like "*компания-монополист*» (*monopolist company*), "*писатель-фантаст*» (*science fiction writer*) that are rather rare by itself, but are composed of high-frequency words. This handling is very primitive: words are split by hyphen, and all possible pairs are compared for similarity, e.g. for pair "*предпринимательство*» (*enterprise*) and "*кибер-коммерция*» (*cyber-commerce*) the resulting pairs would be "*предпринимательство*» (*enterprise*), "*кибер*» (*cyber*) and "*предпринимательство*» (*enterprise*), "*коммерция*» (*commerce*). Obviously, words with hyphens constitute a small fraction of all words, so recall is only 1.11%, and gain from this special handling is only 0.10% for hj and rt tracks.

Models described in this section have low recall and very low false positive rate, and each returns normalized score in [0; 1] range, so we used the **maximum** of model predictions in the combined model. In order to quantify the gains from separate models, we measured system performance with each model removed, and also measured performance of word vector model without any additional models. Overall, we can summarize gains from the models in the following table (each cell contains performance relative to the full model). Note that bigram model is used for all figures in italic (ae and ae2 tracks except the second column).

Table 4. Performance drops when excluding supplementary models

	full with bigrams	**without bigrams**	**without synonyms**	**without prefix**	**without 2nd. orth. sim.**	**without orth. sim.**	**without hyphen**	**only word2vec**
hj	7.84%	**0.00%**	1.53%	0.00%	0.19%	0.26%	0.10%	1.78%
rt	0.47%	**0.00%**	0.64%	0.15%	1.00%	1.55%	0.10%	2.41%
ae	*0.00%*	7.49%	*0.13%*	*0.03%*	*0.00%*	*0.04%*	*0.01%*	***−0.16%***
ae2	*0.00%*	0.89%	***−0.05%***	*0.01%*	*0.00%*	*0.05%*	*0.01%*	*0.08%*

As we can see, apart from bigram model in case of ae track, other models give modest performance gains, especially on ae and ae2 tracks. Still, combining all models gives around 2% of improvement for hj and rt tracks.

In the case of determining synonymy and hyponymy, supplementary models and sources (namely, synonyms database and orthographic similarity) improve overall performance. In the case of associations we did not find any useful additional sources or techniques, and just a combination of word2vec and bigram models gives the best result.

6. Conclusion and future work

We presented a system for determining semantic similarity between Russian words. The system was developed in Python and is free to download and use[6].

We compared two vector models, analysed the importance of lemmatization and corpus size, and measured the gain of supplementary models. It turned out that word vector model gives the main contribution for word similarity task, and it can be successfully enhanced with other techniques tailored to the task at hand.

We think that further development is possible, and improvement of word vector model seems to be the most promising approach. Most obvious things to try would be increasing corpus size, tuning meta-parameters, experimenting with other solutions to different word forms problem (the one we solved with lemmatizing here). It could be also useful to understand the reason for relatively poor performance of GloVe model.

Another area we did not touch here is the nature of the task in which semantic similarity is needed, as it is not the end in itself. Such external context could influence system design. These types of models also seem a promising start for the problem of word sense disambiguation, as an extension of work on the word sense frequency database [Iomdin et al 2014]. The model might serve a basis for computing context vectors and, by clustering them, derive the senses of polysemantic words.

Acknowledgments

We would like to thank our friends and colleagues Anna Vybornova and Maria Kartysheva for valuable advice and resources. Synonyms database was kindly provided by Valentina Apresjan and HSE students.

References

1. *Bykov A. A.* (2008), The anatomy of terms. 400 derivation elements from Latin and Greek [Anatomiya terminov. 400 slovoobrazovatelnykh elementa iz latyni i grecheskogo], ENAS, Moscow.
2. *Iomdin B. L., Lopukhina A. A., Nosyrev G. V.* (2014), Towards a word sense frequency dictionary [K sozdaniyu chastotnogo slovarya znacheniy slov], Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2014" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2014"], Bekasovo, pp. 199–212.
3. *Tomas Mikolov, Quoc V. Le, Ilya Sutskever.* (2013a) Exploiting Similarities among Languages for Machine Translation. arXiv:1309.4168 [cs.CL]

[6] https://bitbucket.org/kostialopuhin/russe

4. *Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Je Dean.* (2013) Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems (NIPS).
5. *Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean.* (2013c) Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR.
6. *Panchenko A., Loukachevitch N. V., Ustalov D., Paperno D., Meyer C. M., Konstantinova N.* (2015) RUSSE: The First Workshop on Russian Semantic Similarity. In Proceeding of the Dialogue 2015 conference. Moscow, Russia (in print).
7. *Jerey Pennington, Richard Socher, and Christopher D Manning.* (2014b) Glove: Global vectors for word representation. In Conference on Empirical Methods on Natural Language Processing (EMNLP).
8. *Tianze Shi, Zhiyuan Liu.* (2014) Linking GloVe with word2vec. arXiv:1411.5595 [cs.CL]
9. *Ellen M. Voorhees.* (1994), Query Expansion using Lexical-Semantic Relations. SIGIR '94.