Building Web-Interfaces for Vector Semantic Models with the WebVectors Toolkit

Andrey Kutuzov
University of Oslo
Oslo, Norway
andreku@ifi.uio.no

Elizaveta Kuzmenko
Higher School of Economics
Moscow, Russia
eakuzmenko_2@edu.hse.ru

Abstract

We present WebVectors, a toolkit that facilitates using distributional semantic models in everyday research. Our toolkit has two main features: it allows to build web interfaces to query models using a web browser, and it provides the API to query models automatically. Our system is easy to use and can be tuned according to individual demands. This software can be of use to those who need to work with vector semantic models but do not want to develop their own interfaces, or to those who need to deliver their trained models to a large audience. WebVectors features visualizations for various kinds of semantic queries. For the present moment, the web services with Russian, English and Norwegian models are available, built using WebVectors.

1 Introduction

In this demo we present *WebVectors*, a free and open-source toolkit[1] helping to deploy web services which demonstrate and visualize distributional semantic models (widely known as word embeddings). We show its abilities on the example of the living web service featuring distributional models for English and Norwegian[2].

Vector space models, popular in the field of distributional semantics, have recently become a buzzword in natural language processing. In fact, they were known for decades, and an extensive review of their development can be found in (Turney et al., 2010). Their increased popularity is mostly due to the new prediction-based approaches, which allowed to train distributional models with large amounts of raw linguistic data very fast. The most established word embedding algorithms in the field are highly efficient *Continuous Skip-Gram* and *Continuous Bag-of-Words*, implemented in the famous *word2vec* tool (Mikolov et al., 2013b; Baroni et al., 2014), and *GloVe* introduced in (Pennington et al., 2014).

Word embeddings represent the meaning of words with dense real-valued vectors derived from word co-occurrences in large text corpora. They can be of use in almost any linguistic task: named entity recognition (Siencnik, 2015), sentiment analysis (Maas et al., 2011), machine translation (Zou et al., 2013; Mikolov et al., 2013a), corpora comparison (Kutuzov and Kuzmenko, 2015), word sense frequency estimation for lexicographers (Iomdin et al., 2016), etc.

Unfortunately, the learning curve to master word embedding methods and how to present the results to general public may be steep, especially for people in (digital) humanities. Thus, it is important to facilitate research in this field and to provide access to relevant tools for various linguistic communities.

With this in mind, we are developing the *WebVectors* toolkit. It allows to quickly deploy a stable and robust web service for operations on word embedding models, including querying, visualization and comparison, all available even to users who are not computer-savvy.

WebVectors can be useful in a very common situation when one has trained a distributional semantics model for one's particular corpus or language (tools for this are now widespread and simple to use), but then there is a need to demonstrate the results to the general public. The toolkit can be installed on any Linux server with a small set of standard tools as prerequisites, and generally works out-of-the-box. The administrator needs

[1] https://github.com/akutuzov/webvectors

[2] http://ltr.uio.no/semvec

Proceedings of the EACL 2017 Software Demonstrations, Valencia, Spain, April 3-7 2017, pages 99–103

only to supply a trained model or models for one's particular language or research goal. The toolkit can be easily adapted for specific needs.

2 Deployment

The toolkit serves as a web interface between distributional semantic models and users. Under the hood it uses the following software:

- *Gensim* library (Řehůřek and Sojka, 2010) which is responsible for actual interaction with models[3];
- Python *Flask* framework responsible for the user interface. It runs either on top of a regular *Apache* HTTP server or as a standalone service (using *Gunicorn* or other standalone WSGI server).

Flask communicates with *Gensim* (functioning as a daemon with our wrapper) via sockets, sending user queries and receiving answers from models.

This architecture allows fast simultaneous processing of multiple users querying multiple models over network. Models themselves are permanently stored in memory, eliminating time-consuming stage of loading them from permanent storage every time there is a need to process a query.

The setup process is extensively covered by the installation instructions available at `https://github.com/akutuzov/webvectors`.

3 Main features of WebVectors

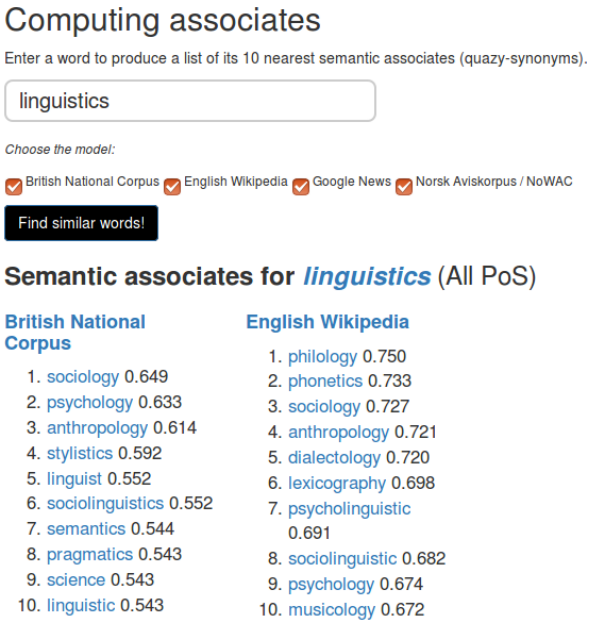
Once *WebVectors* is installed, one can interact with the loaded model(s) via a web browser. Users are able to:

1. find **semantic associates**: words semantically closest to the query word (results are returned as lists of words with corresponding similarity values); an illustration of how it looks like in our demo web service is in Figure 1;

2. calculate exact **semantic similarity** between pairs of words (results are returned as cosine similarity values, in the range between -1 and 1);

[3]Can be any distributional model represented as a list of vectors for words: *word2vec*, *GloVe*, etc.

Figure 1: Computing associates for the word *'linguistics'* based on the models trained on English Wikipedia and BNC.



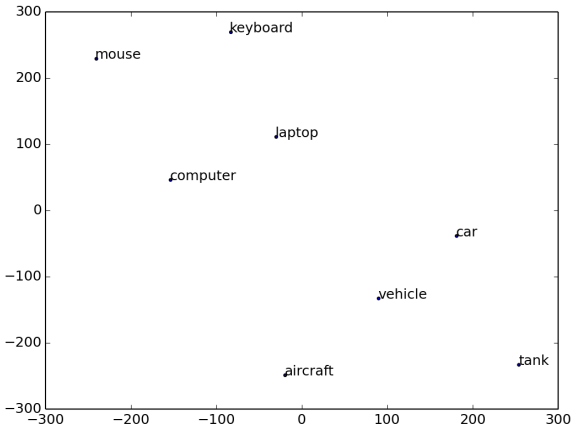
3. apply **algebraic operations** to word vectors: addition, subtraction, finding average vector for a group of words (results are returned as lists of words nearest to the product of the operation and their corresponding similarity values); this can be used for analogical inference, widely known as one of the most interesting features of word embeddings (Mikolov et al., 2013b);

4. **visualize** semantic relations between words. As a user enters a set of words, the service builds a map of their inter-relations in the chosen model, and then returns a 2-dimensional version of this map, projected from the high-dimensional vector space, using *t-SNE* (Van der Maaten and Hinton, 2008). An example of such visualization is shown in Figure 2;

5. get the **raw vector** (array of real values) for the query word.

One can use part-of-speech filters in all of these operations. It is important to note that this is possible only if a model was trained on a PoS-tagged corpus and the tags were added to the resulting lemmas or tokens. Obviously, the tagger should

Figure 2: Visualizing the positions of several words in the semantic space



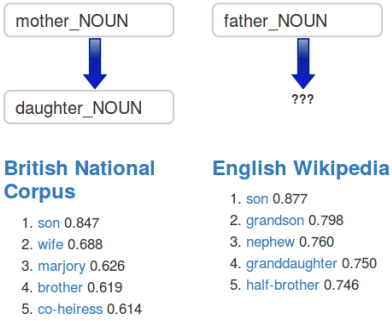
differentiate between homonyms belonging to different parts of speech. *WebVectors* can also use an external tagger to detect PoS for query words, if not stated explicitly by the user. By default, *Stanford CoreNLP* (Manning et al., 2014) is used for morphological processing and lemmatization, but one can easily adapt *WebVectors* to any other PoS-tagger.

In fact, one can use not only PoS tags, but any other set of labels relevant for a particular research project: time stamps, style markers, etc. *WebVectors* will provide the users with the possibility to filter the models' output with respect to these tags.

Another feature of the toolkit is the possibility to display results from more than one model simultaneously. If several models are enumerated in the configuration file, the *WebVectors* daemon loads all of them. At the same time, the user interface allows to choose one of the featured models or several at once. The results (for example, lists of nearest semantic associates) for different models are then presented to the user side-by-side, as in the Figure 3. This can be convenient for research related to comparing several distributional semantic models (trained on different corpora or with different hyperparameters).

Last but not least, *WebVectors* features a simple API that allows to query the service automatically. It is possible to get the list of semantic associates for a given word in a given model or to compute semantic similarity for a word pair. The user performs GET requests to URLs following a

Figure 3: Analogical inference with several models



specific pattern described in the documentation; in response, a file with the first 10 associates or the semantic similarity score is returned. There are two formats available at the present moment: *json* and tab-separated text files.

4 Live demos

The reference web service running on our code base is at `http://ltr.uio.no/semvec`. It allows queries to 4 English models trained with the *Continuous Skipgram* algorithm (Mikolov et al., 2013b): the widely known Google News model published together with the *word2vec* tool, and the models we trained on Gigaword, British National Corpus (BNC) and English Wikipedia dump from September 2016 (we plan to regularly update this last one). Additionally, it features a model trained on the corpus of Norwegian news texts, *Norsk aviskorpus* (Hofland, 2000). To our knowledge, this is the first neural embedding model trained on the Norwegian news corpus made available online; (Al-Rfou et al., 2013) published distributional models for Norwegian, but they were trained on the Wikipedia only, and did not use the current state-of-the-art algorithms.

Prior to training, each word token in the training corpora was not only lemmatized, but also augmented with a Universal PoS tag (Petrov et al., 2012) (for example, *boot_VERB*). Also, some amount of strongly related bigram collocations like '*Saudi::Arabia_PROPN*' was extracted, so that they receive their own embeddings after the training. The Google News model already features ngrams, but lacks PoS tags. To make it more comparable with other models, we assigned each word in this model a PoS tag with *Stanford CoreNLP*.

Another running installation of our toolkit is the *RusVectores* service available at `http://`

rusvectores.org (Kutuzov and Kuzmenko, 2016). It features 4 *Continuous Skipgram* and *Continuous Bag-of-Words* models for Russian trained on different corpora: the Russian National Corpus (RNC)[4], the RNC concatenated with the Russian Wikipedia dump from November 2016, the corpus of 9 million random Russian web pages collected in 2015, and the Russian news corpus (spanning time period from September 2013 to November 2016). The corpora were linguistically pre-processed in the same way, lending the models the ability to better handle rich morphology of Russian. The *RusVectores* is already being employed in academic studies in computational linguistics and digital humanities (Kutuzov and Andreev, 2015; Kirillov and Krizhanovskij, 2016; Loukachevitch and Alekseev, 2016) (several other research projects are in progress as of now).

One can use the aforementioned services as live demos to evaluate the *WebVectors* toolkit before actually employing it in one's own workflow.

5 Conclusion

The main aim of *WebVectors* is to quickly deploy web services processing queries to word embedding models, independently of the nature of the underlying training corpora. It allows to make complex linguistic resources available to wide audience in almost no time. We continue to add new features aiming at better understanding of embedding models, including sentence similarities, text classification and analysis of correlations between different models for different languages. We also plan to add models trained using other algorithms, like *GloVe* (Pennington et al., 2014) and *fastText* (Bojanowski et al., 2016).

We believe that the presented open source toolkit and the live demos can popularize distributional semantics and computational linguistics among general public. Services based on it can also promote interest among present and future students and help to make the field more compelling and attractive.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria, August. Association for Computational Linguistics.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Knut Hofland. 2000. A self-expanding corpus based on newspapers on the web. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000)*.

B. L. Iomdin, A. A. Lopukhina, K. A. Lopukhin, and G. V. Nosyrev. 2016. Word sense frequency of similar polysemous words in different languages. In *Computational Linguistics and Intellectual Technologies. Dialogue*, pages 214–225.

A. N. Kirillov and A. A. Krizhanovskij. 2016. Model' geometricheskoj struktury sinseta [the model of geometrical structure of a synset]. *Trudy Karel'skogo nauchnogo centra Rossijskoj akademii nauk*, (8).

Andrey Kutuzov and Igor Andreev. 2015. Texts in, meaning out: neural language models in semantic similarity task for Russian. In *Computational Linguistics and Intellectual Technologies. Dialogue*, Moscow. RGGU.

Andrey Kutuzov and Elizaveta Kuzmenko. 2015. Comparing neural lexical models of a classic national corpus and a web corpus: The case for Russian. *Lecture Notes in Computer Science*, 9041:47–58.

Andrey Kutuzov and Elizaveta Kuzmenko. 2016. Webvectors: a toolkit for building web interfaces for vector semantic models (in press).

Natalia Loukachevitch and Aleksei Alekseev. 2016. Gathering information about word similarity from neighbor sentences. In *International Conference on Text, Speech, and Dialogue*, pages 134–141. Springer.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

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

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. European Language Resources Association (ELRA).

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA.

Scharolta Katharina Sienčnik. 2015. Adapting word2vec to named entity recognition. In *Nordic Conference of Computational Linguistics NODALIDA 2015*, page 239.

Peter D. Turney, Patrick Pantel, et al. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85.

Will Y. Zou, Richard Socher, Daniel M. Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398.