WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models

Andrey Kutuzov[1] and Elizaveta Kuzmenko[2(✉)]

[1] University of Oslo, Oslo, Norway
andreku@ifi.uio.no
[2] National Research University Higher School of Economics, Moscow, Russia
eakuzmenko_2@edu.hse.ru

Abstract. The paper presents a free and open source toolkit which aim is to quickly deploy web services handling distributed vector models of semantics. It fills in the gap between training such models (many tools are already available for this) and dissemination of the results to general public. Our toolkit, *WebVectors*, provides all the necessary routines for organizing online access to querying trained models via modern web interface. We also describe two demo installations of the toolkit, featuring several efficient models for English, Russian and Norwegian.

Keywords: Distributional semantics · Neural embeddings · Word2vec · Machine learning · Visualization

1 Introduction

In this paper we present *WebVectors*,[1] a free and open-source toolkit to deploy web services implementing vector semantic models, primarily word embeddings.

Vector models of distributional semantics are well established in the field of computational linguistics and have been here for decades (see [1] for an extensive review). However, recently they received substantially growing attention. The main reason for this is a possibility to employ artificial neural networks trained on large corpora to learn low-dimensional distributional vectors for words (word embeddings). The most well-known tool in this field now is *word2vec* and its Skip-Gram and CBOW algorithms, which allow fast training on huge amounts of raw linguistic data [2].

Word embeddings represent meaning of words, and can be of use in almost any linguistic task: named entity recognition [3], sentiment analysis [4], machine translation [5], corpora comparison [6], etc. Approaches implemented in *word2vec* and other similar tools are being extensively studied and tested in application to the English language: see [7] and many others. However, for many other languages the surface is barely scratched. Thus, it is important to facilitate research in this field and to provide access to relevant tools for various linguistic communities.

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

D.I. Ignatov et al. (Eds.): AIST 2016, CCIS 661, pp. 155–161, 2017.
DOI: 10.1007/978-3-319-52920-2_15

With this in mind, we release the *WebVectors* toolkit. It allows to quickly deploy a stable and robust web service for operations on vector semantic models, including querying, visualization and comparison, all available to users of any computer literacy level. It 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 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 Basics

Technically, the toolkit is a web interface between distributional semantic models and a user. Under the hood, we use *Gensim* library [8] which deals with models' operations. The user interface is implemented in Python (*Flask* framework) and runs on top of a regular Apache HTTP server or as a standalone service (using, for example, *Gunicorn*). It communicates with *Gensim* (functioning as a daemon with our wrapper) via sockets, sending user queries and receiving back models' answers.

Such 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 hard drive every time there is a need to process a query.

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 one's results to general public. The setup process then is as follows:

1. install project and its dependencies at your Linux server according to the user guide (it is basically installing *Flask* and *Gensim* and copying *WebVectors* files to your web directory);
2. put your model(s) in '*models*' sub-directory of *WebVectors*;
3. change configuration files, stating the paths to your models;[2]
4. optionally change other settings;
5. run Python script to load models into memory and start daemon listening to queries via sockets;
6. run Apache or other web server you use, to start user interface listening to HTTP queries.

3 Main Features of WebVectors

Immediately after that you can interact with the loaded model via web browser. From a user's point of view, *WebVectors* is a semantic calculator which operates on relations between words in distributional models. In particular, users are able to:

[2] As of now, *WebVectors* supports models in generic *Word2vec* format (which is essentially a simple list of word vectors, in text or binary form) and *gensim* format (it is always binary and retains much more technical data, including output vectors).

1. find **semantic associates**: words semantically closest to the query word (results are returned as lists of words with corresponding similarity values);
2. calculate exact **semantic similarity** between pairs of words (results are returned as similarity values, in the range between −1 and 1);
3. apply simple **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 reasoning, widely known as one of the most interesting features of word embeddings [2];
4. **visualize** word vectors and their geometrical relations;
5. get the **raw vector** (array of real values) for the query word.

All these operations can optionally employ part-of-speech filters. Of course, to this end the model must be trained on a PoS-tagged corpus and must differentiate between homonyms belonging to different parts of speech. Also, in this case *WebVectors* can use an external tagger to detect PoS of the query words, if not stated explicitly by the user. By default, Freeling suite of linguistic analyzers [9] is employed for morphological processing: main reasons for choosing Freeling is that it is open-source, supports multiple languages and provides thread-safe parallel query processing, at the same time featuring sufficient accuracy (about 98% for English). However, one can easily adapt *WebVectors* to use any PoS-tagger of one's own choice.

Another feature of the toolkit is the possibility to use several models 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 featured models or several at once. The results (for example, lists of nearest semantic associates) for different models are then presented to user side-by-side. Thus, it is convenient to conduct research related to comparing several distributional semantic models (trained on different corpora or with different hyperparameters).

For some categories of users it may be important that *WebVectors* web GUI is HTML5-compliant and fully supports mobile devices. It is also inherently multilingual: extending the interface with another language is pretty straightforward, demanding only to add an entry for the new language in the configuration file and to add new translations for text strings in the localization file (we provide English and Russian translations).

In the spirit of the Semantic Web paradigm, each word in each model has its own unique URI (Uniform Resource Identifier) explicitly stating lemma, model and PoS: for example, http://example.com/webvectors/your_model/boot_N. In response to requests for these addresses, we return a special page for this word in this model, providing lists of the nearest semantic associates which belong to the same PoS as the lemma itself (if PoS-aware model is used). Additionally, the word vector and its visualization are shown, complete with links to search for the word on the Internet or in the Wiktionary.

Web interface and neat HTML5 pages can be good for initial exploration into the data or for live demos, but real studies often demand large-scale querying of

models. That is why *WebVectors* returns not only human-readable results, but also provides simple API. Using this, one can query the service from one's own application and receive results in the form of tab-separated text file or JSON.

3.1 Visualization Possibilities

WebVectors provides two kinds of visualizations: for vectors of single words and for inter-relations between several words in the model. Single word visualizations are simple plots of corresponding n-dimensional vectors. They can be useful for explaining how distributed semantic models work, for the audience which is not math-savvy. The Fig. 1 demonstrates such a plot for the 300-dimensional vector of the word 'лингви стика' *linguistics* from the model trained on Russian National Corpus.

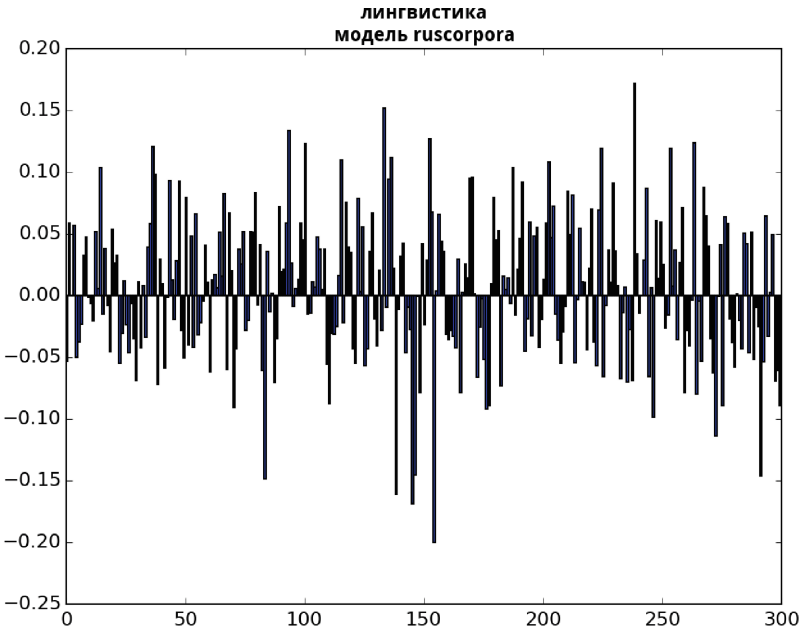

Fig. 1. Visualization of a single word vector

Multiple words visualizations are implemented using the well-known t-SNE algorithm [10] and project complex semantic relationships into the 2-dimensional space, possibly providing useful insights into the data structure. The algorithm tries to keep as much information about high-dimensional geometry as possible. These plots are shown for queries consisting of 7 or more words (with less words visualizations usually being not informative).

An example of such a plot for words '*mouse, keyboard, computer, laptop, aircraft, vehicle, car, tank, wine, beer, whisky*' in a model trained on Google

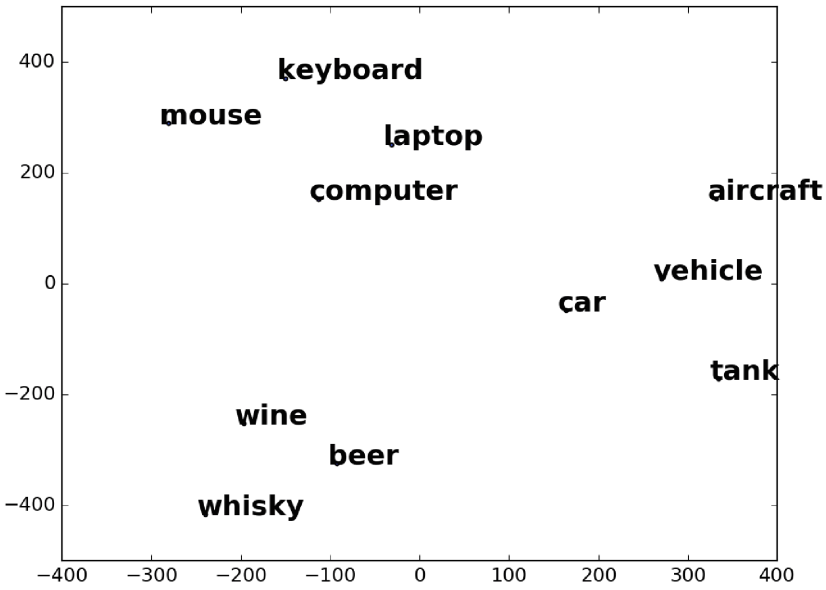

Fig. 2. Words relations visualization using t-SNE

News data set is shown in the Fig. 2. One can see how three groups of words are clustered in different parts of the plot: computer hardware, alcoholic beverages and transportation means.

Note that the layout of points in t-SNE plots is an approximate projection of real word layout in the multidimensional vector space of the models. Thus, distances between these points also only approximately reflect cosine similarities between high-dimensional word vectors.

4 Live Demos

As stated above, *WebVectors* can facilitate semantic research for languages which are less subject to computational linguists' attention. One running installation of our toolkit proves this: this is *RusVectores* service available at http://ling.go.mail.ru/dsm/. The service is already being employed in academic studies in computational linguistics [11] and digital humanities (several research projects in process as of now).

It features four models for Russian trained on different corpora. Note that prior to training, each word token in the corpora was not only lemmatized, but also augmented with a marker denoting its part of speech (for example, 'печь_V' for the verb 'to bake' which in Russian is homonymous to the noun 'furnace'). This linguistic preprocessing lends the models the ability to better handle rich morphology of Russian. In addition, it allows issuing PoS-aware queries like 'What are the most semantically similar verbs to this noun?'.

All the regularities described in English-related publications can be shown to retain in Russian models, including analogical inference through algebraic operations on vectors. For example, the model trained on concatenated Russian Wikipedia and Russian National Corpus returns быт 'daily round' if subtracting смысл 'meaning' from жизнь 'life'. The existing research in English models has shown that such relationships can be useful for many applications, including machine translation [12].

RusVectores allows users to upload their own corpora and train models on them server-side. After the corpus is uploaded, the user is provided with a unique identifier, allowing access to the trained model, which can be then downloaded. It is possible to train models either on raw texts or on PoS-tagged corpora. Training hyperparameters (context window length, vector dimensionality, etc.) are defined by a user via web interface.

Another service running on our code base is *Semantic Vectors* available at http://ltr.uio.no/semvec. It allows queries to 3 English models: the widely known Google News model distributed with *word2vec* tool, and the models we trained on British National Corpus (BNC) and English Wikipedia. One can study semantic differences between modern English language featured in Wikipedia and more diverse but a bit outdated variety represented by BNC. Google News model is trained on a very large corpus (100 billion words), but lacks linguistic preprocessing, which in turn leads to more ways of performing interesting comparisons. Additionally, *Semantic Vectors* features a model trained on the corpus of Norwegian news texts, *Norsk aviskorpus* [13]. To our knowledge, this is the first neural embedding model for Norwegian made available online.

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 vector semantic models, independently of a particular language. It allows to make complex linguistic resources available to wide audience in almost no time. The authors plan to continue adding 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 are striving to lower the entry threshold for the distributional semantics field. Neural word embeddings seem to be a very promising approach to many NLP tasks that can be widely used. At the same time, it is important that researchers (especially linguists) with no solid programming background were able to use these powerful approaches: both in their studies and in disseminating their results. It is particularly true for a very popular discipline of digital humanities, where convenient web access to data is paramount. All these aims can be achieved with *WebVectors*.

Finally, we believe that the presented toolkit can popularize distributional semantics and computational linguistics among general public. Services based

on it can promote interest among present and future students and help to make the field more compelling and attractive.

References

1. Turney, P.D., Pantel, P., et al.: From frequency to meaning: vector space models of semantics. J. Artif. Intell. Res. **37**(1), 141–188 (2010)
2. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Adv. Neural Inf. Process. Syst. **26**, 3111–3119 (2013)
3. Siencnik, S.K.: Adapting word2vec to named entity recognition. In: Nordic Conference of Computational Linguistics, NODALIDA 2015, p. 239 (2015)
4. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 142–150. Association for Computational Linguistics (2011)
5. Zou, W.Y., Socher, R., Cer, D.M., Manning, C.D.: Bilingual word embeddings for phrase-based machine translation. In: EMNLP, pp. 1393–1398 (2013)
6. Kutuzov, A., Kuzmenko, E.: Comparing neural lexical models of a classic national corpus and a web corpus: the case for Russian. In: Gelbukh, A. (ed.) CICLing 2015. LNCS, vol. 9041, pp. 47–58. Springer, Heidelberg (2015). doi:10.1007/978-3-319-18111-0_4
7. Baroni, M., Dinu, G., Kruszewski, G.: 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, vol. 1 (2014)
8. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, ELRA, pp. 45–50, May 2010
9. Padró, L., Stanilovsky, E.: Freeling 3.0: towards wider multilinguality. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey, European Language Resources Association (ELRA), May 2012
10. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(2579–2605), 85 (2008)
11. Kutuzov, A., Andreev, I.: Texts in, meaning out: neural language models in semantic similarity task for Russian. In: Proceedings of the Dialog Conference, Moscow, RGGU (2015)
12. Mikolov, T., Le, Q., Sutskever, I.: Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168 (2013)
13. Hofland, K.: A self-expanding corpus based on newspapers on the web. In: LREC (2000)