Wembedder: Wikidata entity embedding web service

Finn Årup Nielsen
Cognitive Systems, DTU Compute
Technical University of Denmark
Kongens Lyngby, Denmark
faan@dtu.dk

ABSTRACT

I present a web service for querying an embedding of entities in the Wikidata knowledge graph. The embedding is trained on the Wikidata dump using Gensim's Word2Vec implementation and a simple graph walk. A REST API is implemented. Together with the Wikidata API the web service exposes a multilingual resource for over 600'000 Wikidata items and properties.

Keywords

Wikidata, embedding, RDF, web service.

1. INTRODUCTION

The Word2Vec model [7] spawned an interest in dense word representation in a low-dimensional space, and there are now a considerable number of "2vec" models beyond the word level.[1] One recent avenue of research in the "2vec" domain uses knowledge graphs [13]. Such systems can take advantage of the large knowledge graphs, e.g., DBpedia or Freebase, for graph embedding. Graph embedding in the simplest case would map individual nodes of the network to a continuous low-dimensional space, while embedding with knowledge graphs would typically handle the typed links between knowledge items/nodes.

Wikidata https://www.wikidata.org/ [16] is a relatively new knowledge graph resource. It is run by the Wikimedia Foundation that is also behind Wikipedia, thus Wikidata can be regarded as a sister site to Wikipedia. While Wikipedia has been extensively used as a data and text mining resource [6], Wikidata has so far seen less use in machine learning contexts. There are several advantages with Wikidata. Wikidata is not tied to a single language, but can include labels for hundreds of languages for each item in the knowledge graph. As such, an embedding that works from Wikidata items is in principle multilingual (however, there is no guarantee that the item label for a specific language is set). Another advantage with Wikidata is that each item can provide extra contextual data from the Wikidata statements. Search in Wikidata is enabled by string-based search engines in the Wikidata API as well as the SPARQL-based Wikidata Query Service (WDQS). General functions for finding related items or generating fixed-length features for machine learning are to my knowledge not available.

There is some research that combines machine learning and Wikidata [9, 14], e.g., Mousselly-Sergieh and Gurevych have presented a method for aligning Wikidata items with FrameNet based on Wikidata labels and aliases [9].

Property suggestion is running live in the Wikidata editing interface, where it helps editors recall appropriate properties for items during manual editing, and as such a form of recommender system. Researchers have investigated various methods for this process [17].

Scholia at https://tools.wmflabs.org/scholia/ is our web service that presents scholarly profiles based on data in Wikidata extracted with WDQS [11]. Counting co-occurrence patterns with SPARQL queries, Scholia can list related items based on a query item. For instance, Scholia lists related diseases based on overlapping associated genes.[2] Other than these count- and SPARQL-based methods Scholia has limited means to show related items to a Scholia user.

Several research groups provide word embedding web services: GPL-licensed *WebVectors* uses a Flask and Gensim [5, 4], and instances for English and Russian run at http://rusvectores.org/ and for English and Norwegian at http://vectors.nlpl.eu/explore/embeddings/. A Turku BioNLP group provides a Flask-based word embedding web service at http://bionlp-www.utu.fi/wv_demo/ based on the English Google News and Finnish Corpora. A web service for handling multilingual word embeddings has also been announced [1]. Wembedder is distinguished from these services by using the Wikidata entities (items and properties) as the "words" in the embedding (rather than natural language words) and by using the live Wikidata web service to provide multilingual labels for the entities.

2. WEMBEDDER

2.1 Model setup

The Wikimedia Foundation provides the Wikidata RDF dumps for download at https://dumps.wikimedia.org/wikidatawiki/entities/. For the setup of the initial model, I downloaded the so-called truthy dumps available in Notation3 format. The specific file was the 5.2 GB large compressed file `wikidata-20170613-truthy-BETA.nt.bz2`.

[1] https://github.com/MaxwellRebo/awesome-2vec

[2] See, e.g., the page for schizophrenia at https://tools.wmflabs.org/scholia/disease/Q41112.

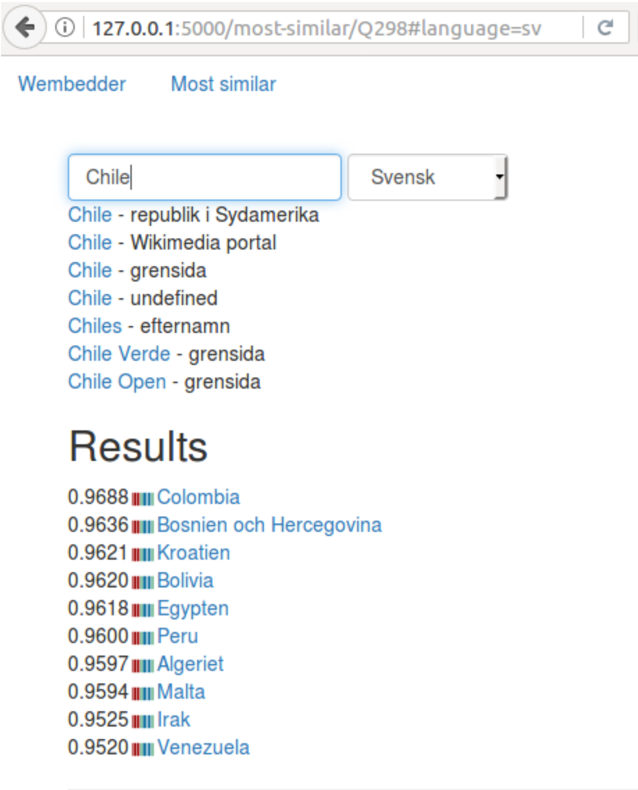

Figure 1: Wembedder's output after a query on Q298 (the country Chile) with the interface set to Swedish.

Figure 2: JSON output from Wembedder's REST API with a query on Q2 (Earth) rendered in a web browser. The first entities in the result list is Q313 (Venus).

The truthy dumps only have a limited set of all the triples in Wikidata: Those that are associated with the `wdt` prefix. From this dump, I extracted the triples where both subject and object were Wikidata items, i.e., leaving out triples where the object is a value such as an identifier, a date, a name, etc. The generated file contains 88'941'173 lines each with a triple. The `http://www.wikidata.org/entity/` and `http://www.wikidata.org/prop/direct/` prefixes were stripped, so the first few lines of the generated file have the following content in a format similar to Magnus Manske's QuickStatements format:

```
Q22 P1546 Q2016568
Q22 P610 Q104674
Q22 P1151 Q8143311
Q22 P31 Q3336843
Q22 P36 Q23436
Q22 P47 Q21
...
```

Each line can be regarded as a very simple graph walk consisting of a single step from one Wikidata item through a typed property to the next Wikidata item. These triple data I now regard as a sentence of three "words" which can be treated by standard word embedding implementations. I use the Word2Vec model in the Gensim program [18]. The initial model trained used the CBOW training algorithm, an embedding dimension on 100, a window of 1 and a minimum count of 20, i.e., any "word" must appear 20 times or more to be included in the model. The rest of the parameters in the Word2Vec model were kept at the Gensim defaults. With this setup, the model ends up with a vocabulary of 609'471. This number includes 738 properties and 608'733 Wikidata items. Gensim can store its model parameters in files with a combined size of 518 megabytes. A permanent version of the model parameters is available in Zenodo under DOI 10.5281/zenodo.823195.

2.2 Web service

The web service was set up with the Python Flask web framework [3] with the Apache-licensed code available at a GitHub repository: `https://github.com/fnielsen/wembedder`. Figure 1 shows the interface. A version of Wembedder runs from `https://tools.wmflabs.org/wembedder/`, i.e., from the cloud service provided by the Wikimedia Foundation.

The Wembedder web service relies on the Wikidata API at `https://www.wikidata.org/w/api.php` and its *wbsearchentities* action for searching for items in multiple languages in an implementation based on the search facility on the Wikidata homepage. Labels for searching and labels for the results are generated via ajax calls to the Wikidata API.

A REST API is implemented as part of Wembedder and returns JSON-formatted results, e.g., `/api/most-similar/Q80` will return the most similar entities for a query on Tim Berners-Lee (Q80), see also Figure 2. Similarity computations are implemented with a URL such as `/api/similarity/Q2/Q313`. Here the Earth and Venus are compared. The human interface of Wembedder uses the REST API in a ajax fashion, returning an HTML page with an empty result list and with JavaScript for the actual fetching of the results.

3. EVALUATION

The embedding in the current version of Wembedder is fairly simple compared to the state of the art embeddings, that uses complex/holographic knowledge graph embedding [10] or multiple knowledge graphs and pre-trained corpora-based resources for building the embedding [15]. One should not expect Wembedder to perform at the state of the art level,[3] and a comparison with the Wordsim-353 dataset for semantic relatedness evaluation [2] shows poor performance with Pearson and Spearman correlations on just 0.13.

When used to evaluate the Wikidata graph embedding, a matching is needed between English Wordsim-353 words and the Wikidata items. It is not straightforward as there usually is a semantic difference between the words and the items. It is often possible to find the word as the English label in a Wikidata item, but for instance, for the Wordsim-353 word "Japanese" one must decide whether it should be linked to Japanese as a language (Q5287), Japanese as a people (Q161652), another item (e.g., the disambiguation page, Q346080) or an average or sum over the items. I attempted to match the words with items, but left several unmatched so only 278 of the word pairs of the 353 were possible in the analysis. The correlations were computed from these 278 word pairs. A skip-gram trained model yielded even lower performance with correlations of just 0.11 and 0.10 for Pearson and Spearman correlations, respectively. A CBOW model trained with a higher number of iterations (DOI 10.5281/zenodo.827339) performed somewhat better with correlations of 0.21.

4. DISCUSSION AND FUTURE WORK

Wembedder—with its 100-dimensional Gensim model query—will usually be able to return results in around one second, while the API call is considerably faster. It means that it could be used for interactive "related items" search. The SPARQL-based related items queries in Scholia usually takes several seconds.

Wikidata at its current state is mostly an encyclopedic source having little lexical information. State of the art relational modeling ConceptNet is setup from both encyclopedic and lexical knowledge graphs as well as corpus-based embeddings [15]. Embeddings based on Wikidata could presumably perform better by using the link to Wikipedia with the different language versions of Wikipedia acting as a corpora. There exist several works describing joint models of words and entities from knowledge bases/graphs, see, e.g., [8] and reference therein. There is work underway to enable Wikidata to represent lexical information [12]. A Wikidata-based embedding may benefit from such data.

5. ACKNOWLEDGMENT

The Danish Innovation Foundation supported this work through Danish Center for Big Data Analytics driven Innovation (DABAI).

6. REFERENCES

[1] AMMAR, W., MULCAIRE, G., TSVETKOV, Y., LAMPLE, G., DYER, C., AND SMITH, N. A. Massively Multilingual Word Embeddings.

[2] FINKELSTEIN, L., GABRILOVICH, E., MATIAS, Y., RIVLIN, E., SOLAN, Z., WOLFMAN, G., AND RUPPIN, E. Placing search in context: the concept revisited. *ACM Transactions on Information Systems 20* (January 2002), 116–131.

[3] GRINBERG, M. Flask Web Development.

[4] KUTUZOV, A., AND KUZMENKO, E. Building Web-Interfaces for Vector Semantic Models with the WebVectors Toolkit. *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics* (April 2017), 99–103.

[5] KUTUZOV, A., AND KUZMENKO, E. WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models. *Analysis of Images, Social Networks and Texts: 5th International Conference, AIST 2016* (December 2017), 155–161.

[6] MEHDI, M., OKOLI, C., MESGARI, M., NIELSEN, F. Å., AND LANAMÄKI, A. Excavating the mother lode of human-generated text: A systematic review of research that uses the wikipedia corpus. *Information Processing & Management 53* (March 2017), 505–529.

[7] MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J. Efficient Estimation of Word Representations in Vector Space.

[8] MORENO, J. G., BESANÇON, R., BEAUMONT, R., D'HONDT, E., LIGOZAT, A.-L., ROSSET, S., TANNIER, X., AND GRAU, B. Combining Word and Entity Embeddings for Entity Linking. *The Semantic Web* (May 2017), 337–352.

[9] MOUSSELLY-SERGIEH, H., AND GUREVYCH, I. Enriching Wikidata with Frame Semantics. *Proceedings of the 5th Workshop on Automated Knowledge Base Construction* (June 2016), 29–34.

[10] NICKEL, M., ROSASCO, L., AND POGGIO, T. Holographic Embeddings of Knowledge Graphs.

[11] NIELSEN, F. Å., MIETCHEN, D., AND WILLIGHAGEN, E. Scholia and scientometrics with Wikidata.

[12] PINTSCHER, L. Let's move forward with support for Wiktionary. *Wikidata mailing list* (September 2016).

[13] RISTOSKI, P., AND PAULHEIM, H. RDF2Vec: RDF Graph Embeddings for Data Mining. *The Semantic Web – ISWC 2016* (December 2016), 498–514.

[14] SOROKIN, D., AND GUREVYCH, I. End-to-end Representation Learning for Question Answering with Weak Supervision.

[15] SPEER, R., CHIN, J., AND HAVASI, C. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (December 2016), 4444–4451.

[16] VRANDEČIĆ, D., AND KRÖTZSCH, M. Wikidata: a free collaborative knowledgebase. *Communications of the ACM 57* (October 2014), 78–85.

[17] ZANGERLE, E., GASSLER, W., PICHL, M., STEINHAUSER, S., AND SPECHT, G. An Empirical Evaluation of Property Recommender Systems for Wikidata and Collaborative Knowledge Bases. *Proceedings of the 12th International Symposium on Open Collaboration* (December 2016).

[18] ŘEHŮŘEK, R., AND SOJKA, P. Software framework for topic modelling with large corpora. *New Challenges For NLP Frameworks Programme* (May 2010), 45–50.

[3] An overview of the state of the art performance in semantic relatedness task, including for the Wordsim-353 task, is available at https://aclweb.org/aclwiki/index.php?title=Similarity_(State_of_the_art)