KGvec2go – Knowledge Graph Embeddings as a Service

Jan Portisch (1,2), Michael Hladik (2), Heiko Paulheim (1)

(1) University of Mannheim - Data and Web Science Group, (2) SAP SE

(1) B 6, 26 68159 Mannheim, Germany (2) Dietmar-Hopp Allee 16, 60190, Walldorf, Germany

jan@informatik.uni-mannheim.de, michael.hladik@sap.com, heiko@informatik.uni-mannheim.de

Abstract

In this paper, we present KGvec2go, a Web API for accessing and consuming graph embeddings in a light-weight fashion in downstream applications. Currently, we serve pre-trained embeddings for four knowledge graphs. We introduce the service and its usage, and we show further that the trained models have semantic value by evaluating them on multiple semantic benchmarks. The evaluation also reveals that the combination of multiple models can lead to a better outcome than the best individual model.

Keywords: RDF2Vec, knowledge graph embeddings, knowledge graphs, background knowledge resources

1. Introduction

A *knowledge graph* (KG) stores factual information in the form of triples. Today, many such graphs exist for various domains, are publicly available, and are being interlinked. As of 2019, the *linked open data cloud* (Schmachtenberg et al., 2014) counts more than 1,000 data sets with multiple billions of unique triples.[1] Knowledge graphs are typically consumed using factual queries for downstream tasks such as question answering. Recently, knowledge graph embedding models are explored as a new way of knowledge graph exploitation. *KG embeddings* (KGEs) represent nodes and (depending on the approach) also edges as continuous vectors. One such approach is *RDF2Vec* (Ristoski and Paulheim, 2016). It has been used and evaluated for machine learning, entity and document modeling, and for recommender systems (Ristoski et al., 2019). RDF2Vec vectors trained on a large knowledge graph have also been used as background knowledge source for ontology matching (Portisch and Paulheim, 2018).

While it has been shown that KGEs are helpful in many applications, embeddings on larger knowledge graphs can be expensive to train and to use for downstream applications. `kgvec2go.org`, therefore, allows to easily access and consume concept embeddings through simple Web APIs. Since most downstream applications only require embedding vectors for a small subset of all concepts, computing a complete embedding model or downloading a complete pre-computed one is often not desirable.

With *KGvec2go*, rather than having to download the complete embedding model, a Web query can be used to obtain only the desired concept in vector representation or even a derived statistic such as the similarity between two concepts. This facilitates downstream applications on less powerful devices, such as smartphones, as well as the application of knowledge graph embeddings in machine learning scenarios where the data scientists do not want to train the models themselves or do not have the means to perform the computations.

Models for four knowledge graphs were learned, namely: *DBpedia* (Lehmann et al., 2015), *WebIsALOD* (Hertling and Paulheim, 2017), *Wiktionary* (Sérasset, 2015), and *WordNet* (Fellbaum, 1998).

The data set presented here allows to compare the performance of different knowledge graph embeddings on different application tasks. It further allows to combine embeddings from different knowledge graphs in downstream applications. We evaluated the embeddings on three semantic gold standards and also explored the combination of embeddings.

This paper is structured as follows: In the next section, related work will be presented. Section 3. outlines the approach, Section 4. presents the data sets for which an embedding has been trained, Section 5. introduces the Web API that is provided to consume the learned embedding models, and Section 6. evaluates the models on three semantic gold standards. The paper closes with a summary and an outlook on future work.

2. Related Work

For data mining applications, propositional feature vectors are required, i.e., vectors with either binary, nominal, or numerical elements. An RDF knowledge graph does not come with such properties and has to be translated into several feature vectors if it shall be exploited in data mining applications. This process is known as *propositionalization* (Kramer et al., 2001; Ristoski and Paulheim, 2014). Two basic approaches for knowledge graph propositionalization can be distinguished: (i) Supervised propositionalization where the user has to manually craft features such as multiple ASK queries for nodes of interest and (ii) unsupervised approaches where the user does not have to know the structure of the graph. (Paulheim and Fümkranz, 2012) In order to exploit knowledge graphs in data mining applications, embedding models have gained traction over the last years. (Wang et al., 2017) distinguish two families of approaches: distance based and semantic matching based approaches. The best known representatives of the first family are translation-based approaches. Given a set of entities E and a set of edges L as well as triples in the form $(head, label, tail)$, usually stated as (h, l, t) where $h, t \in E$ and $l \in L$, *TransE* (Bordes et al., 2013) trains vectors with the learning objective $h + l = t$ given that (h, l, t) holds. Many similar approaches based on TransE have been proposed such as *TransH* (Wang et al., 2014) or *TransA* (Jia et al., 2016). In the second family, the most well known approaches are *RESCAL* (Nickel et al., 2011),

---

[1] `https://lod-cloud.net/`

DistMult (Yang et al., 2014), and *HolE* (Nickel et al., 2016). Another group of approaches exploits language models such as *node2vec* (Grover and Leskovec, 2016) and *RDF2Vec* (Ristoski et al., 2019). This work is based on the latter algorithm. Given a (knowledge) graph $G = (V, E)$ where V is the set of vertices and E is the set of directed edges, the RDF2Vec approach generates multiple sentences per vertex $v \in V$. An RDF2Vec sentence resembles a walk through the graph starting at a specified vertex v. Datatype properties are excluded from the walk generation. After the sentence generation, the *word2vec* algorithm (Mikolov et al., 2013a; Mikolov et al., 2013b) is applied to train a vector representation for each element $v \in V$ and $e \in E$. word2vec is a neural language model. Given the context k of a word w, where k is a set of preceding and succeeding words of w, the learning objective of word2vec is to predict w. This is known as *continuous bag of words* model (CBOW). The *skip-gram* (SG) model is trained the other way around: Given w, k has to be predicted. Within this training process, c defines the size of k and is also known as *window* or *window size*.

RDF2Vec is different from a pure language model in that it uses a knowledge graph as training corpus. Knowledge graphs are typically more structured than human language and can contain named entities that do not have to be explicitly detected.

While there is an ever-growing number of knowledge graph embeddings, few works have addressed the software infrastructure aspect so far. The *OpenKE* toolkit (Han et al., 2018) facilitates a unified framework for efficiently *training* KGEs, but does not address the light-weight exploitation. The closest project to our work is *WEmbedder* (Nielsen, 2017), which, however, only serves embeddings for one single KG, i.e., Wikidata. This makes KGvec2go the first resource serving multiple embedding models simultaneously.

3. Approach

For this work, the RDF2Vec approach has been re-implemented in Java and Python with a more efficient walk generation process. The implementation of the walk generator is publicly available on GitHub[2].

For the sentence generation, duplicate free random walks with *depth = 8* have been generated whereat edges within the sentences are also counted. For *WordNet* and *Wiktionary*, 500 walks have been calculated per entity. For *WebIsALOD* and *DBpedia*, 100 walks have been created in order to account for the comparatively large size of the knowledge graphs.

The models were trained with the following configuration: *skip-gram vectors*, *window size = 5*, *number of iterations = 5*, *negative sampling for optimization*, *negative samples = 25*. Apart from walk-generation adaptations due to the size of the knowledge graphs, the configuration parameters to train the models have been held constant and no data set specific optimizations have been performed in order to allow for comparability.

In addition, a Web API is provided to access the data models in a lightweight way. This allows for easy access to embedding models and to bring powerful embedding models to devices with restrictions in CPU and RAM, such as smart phones. The APIs are introduced in Section 5. The server has been implemented in Python using *flask*[3] and *gensim* (Řehůřek and Sojka, 2010) and can be run using *Apache HTTP Server*. Its code is publicly available on GitHub.[4]

4. The Data Sets

For this work, four data sets have been embedded which are quickly introduced in the following.

4.1. DBnary/Wiktionary

Wiktionary is "[a] collaborative project run by the Wikimedia Foundation to produce a free and complete dictionary in every language"[5]. The project is organized similarly to Wikipedia: Everybody can contribute and edit the dictionary. The content is reviewed in a community process. Like Wikipedia, Wiktionary is available in many languages. *DBnary* (Sérasset, 2015) is an RDF version of Wiktionary that is publicly available[6]. The DBnary data set makes use of an extended *LEMON* model (McCrae et al., 2012) to describe the data. For this work, a recent download from July 2019 of the English Wiktionary has been used.

4.2. DBpedia

DBpedia is a well-known linked data set created by extracting structured knowledge from Wikipedia and other Wikimedia projects. The data is publicly available. For this work, the 2016-10 download has been used.[7] Compared to the other knowledge graphs exploited here, DBpedia contains mainly instances such as the industrial rock band *Nine Inch Nails* (which cannot be found in WordNet or Wiktionary). Therefore, DBpedia is with its instance data complementary to the other, lemma-focused, knowledge graphs.

4.3. WebIsALOD

The *WebIsA* database (Seitner et al., 2016) is a data set which consists of hypernymy relations extracted from the *Common Crawl*[8], a downloadable copy of the Web. The extraction was performed in an automatic manner through Hearst-like lexico-syntactic patterns. For example, from the sentence "[...] added that the country has favourable economic agreements with major economic powers, including the European Union.", the fact isA(european_union,

---

[2]https://github.com/janothan/kgvec2go-walks/

[3]https://flask.palletsprojects.com/en/1.1.x/

[4]https://github.com/janothan/kgvec2go-server/

[5]https://web.archive.org/web/20190806080601/https://en.wiktionary.org/wiki/Wiktionary

[6]http://kaiko.getalp.org/about-dbnary/download/

[7]https://wiki.dbpedia.org/downloads-2016-10

[8]https://commoncrawl.org/

`major_economic_power`) is extracted[9].

WebIsALOD (Hertling and Paulheim, 2017) is the Linked Open Data endpoint which allows to query the data in SPARQL.[10] In addition to the endpoint, machine learning was used to assign confidence scores to the extracted triples. The data set of the endpoint is filtered, i.e. it contains a subset of the original WebIsA database, to ensure a higher data quality. The knowledge graph contains instances (like DBpedia) as well as more abstract concepts that can also be found in a dictionary.

4.4. WordNet

WordNet (Fellbaum, 1998) is a well-known and heavily used database of English word that are grouped in sets which represent one particular meaning, so-called *synsets*. The resource is strictly authored. WordNet is publicly available, included in many natural language processing frameworks, and often used in research. An RDF version of the framework is also available for download and was used for this work.[11]

5. API

`kgvec2go.org` offers a simple Web API to retrieve: (i) individual vectors for concepts in different data sets, (ii) the cosine similarity between concepts directly, and (iii) the top n most related concepts for any given concept. Alternatively, the full models can be downloaded from the Web site directly.[12] The API is accessed through HTTP GET calls and will provide answers in the form of a JSON string. This allows for a simple usage on any device that has Internet access. In addition, natural words can be used to access the data rather than long URIs that follow their own idiosyncratic pattern as it is common for RDF2Vec embedded models. In the following, we will quickly describe the services that are offered. For a full description of the services as well as a graphical user interface to explore the embeddings, we refer to the Web page `kgvec2go.org`.

5.1. Get Vector

`kgvec2go.org` allows to download an individual vector, i.e. a 200 dimensional floating point number array representation of a concept on a particular data set. The HTTP GET call follows the pattern below:
`/rest/get-vector/<data_set>/`
`/<concept_name>`
where `data_set` refers to the data set that shall be used (i.e. one of `alod`, `dbpedia`, `wiktionary`, `wordnet`) and `concept_name` to the natural language identifier of the concept (e.g. *bed*). This call can be used in machine learning scenarios, for instance, where a numerical representation of a concept is required.

For data sets that learn an embedding based on the part-of-speech (POS) of the term, such as WordNet, multiple vectors are returned for one key word if the latter is available
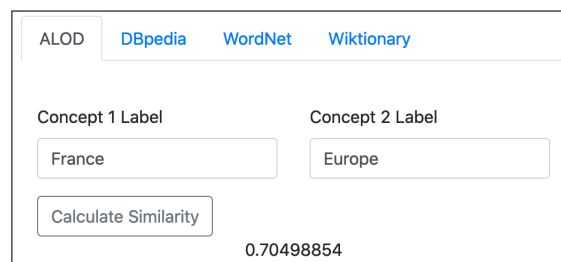

Figure 1: UI to query the similarity of two concepts online. Depicted is the similarity between *France* and *Europe* using the WebIsALOD embeddings.

in multiple POS such as *laugh* which occurs as noun and as verb.

5.2. Get Similarity

Given two concepts, `kgvec2go.org` allows to query a specified data set for the similarity score $s \in [-1.0, 1.0]$ where 1.0 refers to perfect similarity. The HTTP GET call follows the pattern below:
`/rest/get-similarity/<data_set>/`
`<concept_name_1>/<concept_name_2>`
where `data_set` refers to the set that shall be used and the two concept names refer to the concept labels for which the similarity shall be calculated. This call can be used wherever the similarity or relatedness of two concepts needs to be judged such as in recommender systems or matching tasks. A Web UI is available to try out this call in a Web browser.[13] A screenshot is shown in Figure 1 for the terms *France* and *Europe* for the model learned on WebIsALOD.

5.3. Get Closest Concepts

The API is also capable of determining the closest n concepts given a concept and a data set. The given concept is mapped to the vector space and compared with all other vectors. Therefore, the call is expensive on large data sets and should rather be used to explore the data set. The HTTP GET call follows the pattern below:
`/rest/closest-concepts/<data_set>/`
`<top_n>/<concept_name>`
where `data_set` refers to the set that shall be used, `top_n` refers to the number of closest concepts that shall be obtained, and `concept_name` refers to the written representation of the concept. For data sets that learn an embedding based on the part-of-speech of the term, such as WordNet, all closest concepts are determined for all POS of the term and their scores are summarized. This allows to calculate the n closest concepts for a single term, such as *sleep*, that occurs in multiple POS (in this case as noun and as verb). A Web UI is available to try out this call in a Web browser.[14] A screenshot is shown in Figure 2 for the term *Germany* on the trained DBpedia model.

---

[9]This is a real example, see: `http://webisa.webdata commons.org/417880315`

[10]`http://webisa.webdatacommons.org/`

[11]`http://wordnet-rdf.princeton.edu/about/`

[12]`http://www.kgvec2go.org/download.html`

[13]`http://www.kgvec2go.org/query.html`

[14]`http://www.kgvec2go.org/query.html`

ALOD	DBpedia	WordNet	Wiktionary

Germany		Search

#	Concept	Similarity Score
1	dbr:Germany	1
2	dbr:Angela_Merkel	0.87241894
3	dbr:Berlin	0.84634197
4	dbr:Joachim_Gauck	0.828507
5	dbr:Norbert_Lammert	0.80837417
6	dbr:Christian_Wulff	0.72106206
7	dbr:Stanislaw_Tillich	0.6846404
8	dbr:Winfried_Hassemer	0.679457
9	dbr:Marianne_Birthler	0.65852183
10	dbr:Detmold	0.6565967

Figure 2: UI to query the data set online. Shown is the result for query term *Germany* on data set DBpedia. Note that the underlying DBpedia version for the training is that of 2016. In that year, Angela Merkel was the Chancellor of Germany, Berlin the capital of the country, Joachim Gauck the President of Germany, and Norbert Lammert the President of the Bundestag.

6. Evaluation

6.1. Evaluation Gold Standards

In order to test whether there is semantic value in the trained vectors, we evaluate them on three data sets: *WordSim-353* (Finkelstein et al., 2002), *SimLex-999* (Hill et al., 2015), and *MEN* (Bruni et al., 2012). The principle of evaluation is the same for all gold standards used: The system is presented with two words and has to determine their relatedness or similarity; then, the rank correlation (also known as *Spearman's Rho*) with the scores in the gold standards is calculated. Higher correlations between the gold standards' scores and the system's scores are regarded as better. Pairs with an out of vocabulary term are handled here by returning a similarity of 0. As the goal of this data set are comparable general purpose embeddings, it is important to note that the embeddings were not specifically trained to perform well on the given tasks. On similarity tasks, for instance, the results would likely improve when antonymy relations were dropped. With other configuration settings, it is also possible to improve the results further on the given evaluation sets; this has, for instance, been done in (Portisch, 2018) where better relatedness/similarity results on WebIsALOD could be achieved with other RDF2Vec configurations.

6.2. Evaluation Mode

The learned models were evaluated on their own on each of the evaluation data sets. In addition, a combination of all data sets was evaluated. Therefore, the individual similarity scores were added. Hence, $s_{combined}(c_1, c_2) =$

$s_{DBpedia}(c_1, c_2)$ + $s_{WebIsALOD}(c_1, c_2)$ + $s_{Wiktionary}(c_1, c_2) + s_{WordNet}(c_1, c_2)$ where $s_{combined}$ is the final similarity score assigned to the concept pair c_1 and c_2 and $s_{dataset}$ describes the individual score of a model trained on a single *data set* for the same concept pair. This can be done without normalization because (i) all scores are in the same value range ($[-1, 1]$), (ii) out of vocabulary terms receive a score of 0 (so they do not influence the final results), and (iii) because Spearman's rank correlation is used which is independent of the absolute values – only the rank is considered.

6.3. Evaluation Results

The rank correlations on the three gold standards are summarized in Table 1. It can be seen that the results vary depending on the gold standard used. The Wiktionary data set performs best when it comes to relatedness. The WebIsALOD data set performs similarly well on WS-353 and performs best on MEN. On the SimLex-999 gold standard, WordNet outperforms the other data sets. The performance of DBpedia is significantly worse which is due to many out of vocabulary terms: This particular data set is focused on instance data rather than lexical forms such as *angry*. The evaluation performed here is, therefore, not optimal for the data set. This can also be observed in the example results depicted in Table 2: While DBpedia and WebIsALOD work well for entities such as *Germany*, Wiktionary performs better for general words such as *loud*.

Interestingly, the combined evaluation mode outlined in subsection 6.2. is able to outperform the best individual results on WS-353 ($\rho = 0.678$ vs. $\rho = 0.571$) as well as on MEN ($\rho = 0.230$ vs. $\rho = 0.207$). On SimLex, the combination of all similarity scores is very close to the best individual score (WordNet). This shows that it can be beneficial to combine several embedding spaces on different data sets.

It is important to note that the vectors were not trained for the specific task at hand. Nonetheless, the combined embeddings perform well on WS-353 albeit top-notch systems for each data set cannot be outperformed. By the lower performance on SimLex-999 and MEN it can be seen that relatedness is better represented in the embedding spaces than actual similarity. This is an intuitive result given that there was no training objective towards similarity.

When looking at the different properties of the knowledge graphs, it can be reasoned that the level of authoring is not important for the performance on the tasks at hand: WebIsALOD embeddings, which are derived from an automatically generated knowledge graph, easily outperform WordNet embeddings, which are derived from a highly authored knowledge base, on WS-353 and MEN.

6.4. Further Remarks

It is also possible to find typical analogies in the data. In this case, two concepts are presented to the model together with a third one for which the system shall determine an analogous concept. In the following examples, the underlined concept is the best concept that the system found given the three non-underlined concepts.

	WS-353	SimLex-999	MEN
Wiktionary	0.5708	0.2265	0.1513
DBpedia	0.1430	-0.0097	0.0804
WebIsALOD	0.5598	0.1509	0.2066
WordNet	0.4074	0.2870	0.1086
Combined	0.6784	0.2815	0.2304

Table 1: Rank correlation of the data sets with three gold standards.

For example, on Wiktionary:

- *girl* is to *boy* like *man* is to <u>*woman*</u>
- *big* is to *small* like *fake* is to <u>*original*</u>
- *beautiful* is to *attractive* like *quick* is to <u>*rapid*</u>

Similar results can be found on instance level. For example, on DBpedia:

- *Germany* is to *Angela Merkel* like *France* is to <u>*François Hollande*</u>[15]

7. Summary and Future Work

In this paper, we presented KGvec2go, a resource consisting of trained embedding models on four knowledge graphs. The models were evaluated on three different gold standards. It could be shown, that the trained vectors carry semantic meaning and that a combination of different knowledge graph embeddings can be beneficial in some tasks. Furthermore, a lightweight API was presented which allows to consume the models in a computationally cheap, memory-efficient, and easy way through Web APIs. We are confident that our work eases the usage of knowledge graph embeddings in real-world applications.

For the future, we plan to extend the data set by adding more different embedding models of knowledge graphs to the resource presented, as well as including other knowledge graphs, and to extend the capabilities of the current API. Furthermore, we plan to exploit the trained models for downstream application tasks that profit from the inclusion of background knowledge such as ontology matching and domain specific data integration tasks.

8. Bibliographical References

Bordes, A., Usunier, N., García-Durán, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In Christopher J. C. Burges, et al., editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Bruni, E., Boleda, G., Baroni, M., and Tran, N. (2012). Distributional semantics in technicolor. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pages 136–145. The Association for Computer Linguistics.

Christiane Fellbaum, editor. (1998). *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, Massachusetts.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2002). Placing search in context: the concept revisited. *ACM Trans. Inf. Syst.*, 20(1):116–131.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In Balaji Krishnapuram, et al., editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 855–864. ACM.

Han, X., Cao, S., Xin, L., Lin, Y., Liu, Z., Sun, M., and Li, J. (2018). Openke: An open toolkit for knowledge embedding. In *Proceedings of EMNLP*.

Hertling, S. and Paulheim, H. (2017). Webisalod: Providing hypernymy relations extracted from the web as linked open data. In Claudia d'Amato, et al., editors, *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II*, volume 10588 of *Lecture Notes in Computer Science*, pages 111–119. Springer.

Hill, F., Reichart, R., and Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Jia, Y., Wang, Y., Lin, H., Jin, X., and Cheng, X. (2016). Locally adaptive translation for knowledge graph embedding. In Dale Schuurmans et al., editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 992–998. AAAI Press.

Kramer, S., Lavrač, N., and Flach, P. (2001). Propositionalization Approaches to Relational Data Mining. In Sašo Džeroski et al., editors, *Relational Data Mining*, pages 262–291. Springer Berlin Heidelberg, Berlin, Heidelberg.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., and Bizer, C. (2015). Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

McCrae, J., Aguado-de Cea, G., Buitelaar, P., Cimiano, P., Declerck, T., Gómez-Pérez, A., Gracia, J., Hollink, L., Montiel-Ponsoda, E., Spohr, D., and Wunner, T. (2012). Interchanging Lexical Resources on the Semantic Web. *Language Resources and Evaluation*, 46(4):701–719, December.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In Yoshua Bengio et al., editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, et al., editors, *Advances in Neural Informa-*

---

[15] Note that François Hollande is indeed the president of France as of 2016.

	Wiktionary	DBpedia	WebIsALOD	WordNet
1	Germany	Germany	europe	Germany
2	snazziness	Angela Merkel	uk	FRG
3	West Germany	Berlin	france	skillet
4	these islands	Joachim Gauck	canada	Federal Republic of Germany
5	cobbler	Norbert Lammert	japan	Deutschland
6	German Empire	Christian Wulff	italy	High German
7	derisive	Stanislaw Tillich	australia	German
8	who shot John	Winfried Hassemer	usa	Pietism
9	glute	Marianne Birthler	england	Bavaria
10	Okla.	Detmold	asia	ingrained
1	loud	Loud	cons fan	loud (s)
2	silent	Loli	scream	secondly
3	noiseless	Cometa (HVDC)	weird noise	loud (r)
4	rackety	Looc	of noise	aright
5	noisy	Loob	history of 20th century	loud (a)
6	unsilent	Python Server Pages	collective sigh of relief	fruticulose
7	piercing	Louk	thwack	red-handed
8	quiet	Juan Llort	undesired signal	deep down
9	clamorous	Lojo	grinning	every bit
10	blasting	Lone	complaint of office worker	rhymeless

Table 2: Example results for the search terms *Germany* (upper part) and *loud* (lower part). WordNet returns *loud* multiple times with different part-of-speech tags. On DBpedia, results for *Loud* are given as there is no vector for *loud*.

tion Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., pages 3111–3119.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *ICML*, volume 11, pages 809–816.

Nickel, M., Rosasco, L., and Poggio, T. (2016). Holographic embeddings of knowledge graphs. In *Thirtieth Aaai conference on artificial intelligence*.

Nielsen, F. Å. (2017). Wembedder: Wikidata entity embedding web service. *CoRR*, abs/1710.04099.

Paulheim, H. and Fümkranz, J. (2012). Unsupervised generation of data mining features from linked open data. In *Proceedings of the 2nd international conference on web intelligence, mining and semantics*, page 31. ACM.

Portisch, J. and Paulheim, H. (2018). ALOD2Vec Matcher. In Pavel Shvaiko, et al., editors, *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018.*, volume 2288 of *CEUR Workshop Proceedings*, pages 132–137. CEUR-WS.org.

Portisch, J. P. (2018). Automatic schema matching utilizing hypernymy relations extracted from the web.

Řehůřek, R. and Sojka, P. (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. http://is.muni.cz/publicati on/884893/en.

Ristoski, P. and Paulheim, H. (2014). A comparison of propositionalization strategies for creating features from linked open data. *Linked Data for Knowledge Discovery*,

6.

Ristoski, P. and Paulheim, H. (2016). Rdf2vec: RDF graph embeddings for data mining. In Paul T. Groth, et al., editors, *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I*, volume 9981 of *Lecture Notes in Computer Science*, pages 498–514.

Ristoski, P., Rosati, J., Noia, T. D., Leone, R. D., and Paulheim, H. (2019). Rdf2vec: RDF graph embeddings and their applications. *Semantic Web*, 10(4):721–752.

Schmachtenberg, M., Bizer, C., and Paulheim, H. (2014). Adoption of the linked data best practices in different topical domains. In *International Semantic Web Conference*, pages 245–260. Springer.

Seitner, J., Bizer, C., Eckert, K., Faralli, S., Meusel, R., Paulheim, H., and Ponzetto, S. P. (2016). A large database of hypernymy relations extracted from the web. In Nicoletta Calzolari, et al., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Sérasset, G. (2015). Dbnary: Wiktionary as a lemon-based multilingual lexical resource in RDF. *Semantic Web*, 6(4):355–361.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In Carla E. Brodley et al., editors, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 1112–1119. AAAI Press.

Wang, Q., Mao, Z., Wang, B., and Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data*

Engineering, 29(12):2724–2743.

Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. (2014). Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.