# Using Word Embeddings for Search in Linked Data with Ontodia

Gerhard Wohlgenannt[1], Nikolay Klimov[1], Dmitry Mouromtsev[1], Daniil Razdyakonov[2], Dmitry Pavlov[2], and Yury Emelyanov[2]

[1] Intern. Lab. of Information Science and Semantic Technologies, ITMO University, St. Petersburg, Russia `http://en.ifmo.ru/en`
[2] Vismart Ltd., St. Petersburg, Russia `https://vismart.biz`

**Abstract.** Ontodia is an open-source diagramming and visual exploration tool for linked data and ontologies. Here, we present an extension of the Ontodia data query functionalities. We evaluate different types and configurations of word embeddings for improving recall and flexibility of the Ontodia natural language interface. The demonstration will focus especially on the new query functionalities, where Ontodia will be applied to Wikidata as underlying dataset.

**Keywords:** Ontodia, word embeddings, Wikidata, Linked Data, visual interface

## 1 Introduction

One of the key challenges of the Semantic Web and Linked Data is to make the contents of datasets available to users which lack the skills to write semantic queries and understand the underlying data schemata [1]. Ontodia [4] provides visual data exploration for Linked Data in a step-by-step diagrammatic approach. In this demonstration, we combine Ontodia with Wikidata as a knowledge graph. Wikidata[3] is a free and open knowledge base and the central data storage for projects like Wikipedia.

Ontodia allows users to understand and explore datasets, and to answer information needs in a diagrammatic way. It combines its visual interface with textual search for entities and properties in the dataset. Currently the retrieval of entities and properties is limited to exact lexical matches with the labels defined in the dataset. As a simple example, if a user searches for persons *married with* a given person, they might not get any results if they did not use the correct label *spouse* for their query.

In this work, we describe and evaluate improvements to the natural language (NL) interface of Ontodia. The system finds and ranks properties related to a user query using distributional semantics. We evaluate various types of word embeddings against the pre-defined set of aliases for the Wikidata properties. In the demonstration, we plan to present Ontodia and its improved search features, and to discuss our experiences of using word embeddings in querying linked data.

[3] `https://www.wikidata.org`

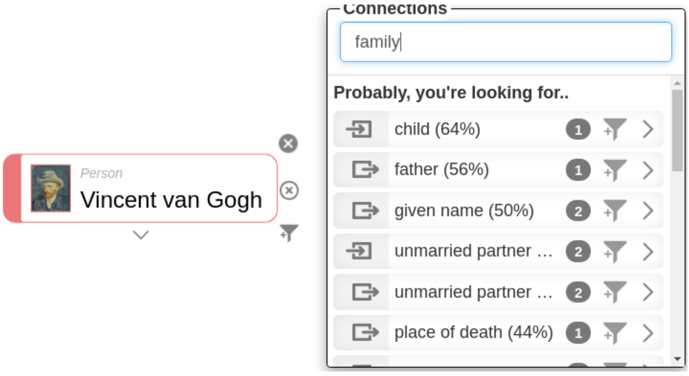
## 2 Related Work

We utilize distributional semantics in the form of word embeddings to enrich the Ontodia NL interface. Word embeddings are language modeling techniques that transform the vocabulary of a given corpus into a continuous and low-dimensional vector space representation. Word embeddings have been applied successfully to many NL processing problems, from word similarity estimations to more complex tasks [3]. Shekarpour et al. [6] describe the challenges in Question Answering on Linked Data. Word embeddings and deep learning techniques are prominently listed as promising techniques for future investigation.

## 3 System Description

Ontodia[4] is a free online RDF and OWL diagramming tool. One of its main use-cases is the visual exploration of linked data sets, and the sharing of information found. In the presented application, Ontodia is used inside the metaphacts[5] platform to explore the Wikidata dataset.

The application described in this demonstration paper is found at: `http://ontodia-prop-suggest.apps.vismart.biz/wikidata.html`. To use the system, a user can search eg. for "Van Gogh" among the *Instances*, and then pull the entity onto the canvas in the center of the window.



Connections

family

Probably, you're looking for..

**Fig. 1.** Searching the properties related to "family" of entity *Van Gogh*

In this demo, we focus on improving the NL interface of Ontodia. After searching instances and classes in the left-hand side of the interface, users can filter the list of properties of the entities displayed on the canvas. Previously, only properties with labels exactly matching the search term were found. For this demo, we experimented with different word embedding models to find and rank properties related to the user input. In a nutshell, using models trained

[4] `http://www.ontodia.org`

[5] `http://www.metaphacts.com`

on a Wikipedia corpus, a representation of every Wikidata property is created by the vectorial sum of the words in its label (and description text). Then we compute the similarity of the user input (represented by the vectorial sum of its words) with the entity properties, and rank the properties by similarity to the user input. The goal is to make the NL interface more powerful and improve user experience and ease-of-use. Figure 1 shows the result of searching for “family” relations for entity *Van Gogh*.

## 4 Evaluation

In this section we give an overview of the evaluation results for various word embedding models for the task of property suggestion for user input terms.

### 4.1 Evaluation Setup

Wikidata includes 3323 different properties and 4603 property aliases. Although the data quality of the aliases is sometimes questionable, the aliases are a sufficiently large gold standard dataset to evaluate our property suggestion modules.

Like for user input, for any of the aliases, first we extract the contained words, and remove stopwords. Then the tool creates the vectorial sum of the alias words using the pre-trained word embedding models (see below), and ranks all 3323 Wikidata properties by cosine similarity with the alias vector. We apply various evaluation metrics. Due to space limitations, here we include only the ratio of correct property suggestions for aliases in the *top-N* of the property ranking, and the mean reciprocal rank (*MRR*).

Also, we conduct experiments with many types, settings and different training corpora for the word embeddings. The best results are encountered with fastText [2] embeddings trained on a Wikipedia corpus, and LexVec [5] embeddings trained on Wikipedia and a news corpus, both with vectors of 300 dimensions.

### 4.2 Results

In Table 1 we present the evaluation results for the fastText and LexVec embeddings. We distinguish models which use only the words from the property label to create the property vector (*not using description*) and property representations that build the vectors by making vectorial sums which include both the words from the property labels and the property descriptions.

Using the property description text to create the property representations improves service quality. Over all metrics, the fastText models perform best. This may result from fastText using word morphology together with the word2vec skipgram model, which is well suited for the task at hand.

In the real world application in visual data exploration in Ontodia the task is much easier as compared to the evaluation in Table 1. When we randomly pick entities from Wikidata – we evaluated with 1154 random entities, these have on average 29.5 properties per entity. When evaluating randomly picked

| | Top 1 | Top 3 | Top 10 | MRR |
|---|---|---|---|---|
| **fastText** using description | **38.12** % | **55.13%** | **70.22%** | **0.493** |
| **LexVec** using description | 37.21% | 53.45% | 67.98% | 0.464 |
| **fastText** not using description | 36.10% | 51.94% | 65.25% | 0.478 |
| **LexVec** not using description | 34.74% | 50.03% | 62.91% | 0.447 |

**Table 1.** Ratio of *test terms* (aliases) with the correct property suggestion in the top-N of the results, depending on word embedding method used and settings; and MRR.

entities using the same gold standard data, 70.5% of first ranked properties are the correct ones, the Top-3 score is 85%, and the MRR is 0.80. The runtime of queries is below 0.01 seconds, fast enough for interactive systems.

## 5 Conclusions

In this demonstration paper we present and evaluate the improved query functionalities of Ontodia, which apply vector similarity with word embeddings between query terms and entity properties. Extensive evaluations show the method is sufficiently accurate for integration in Linked Data visualization tools. In future work we apply the method to other datasets, and the search for entities.

## 6 Acknowledgments

This work was supported by the Government of the Russian Federation (Grant 074-U01) through the ITMO Fellowship and Professorship Program.

## References

1. Augenstein, I., Gentile, A.L., Norton, B., Zhang, Z., Ciravegna, F.: Mapping keywords to linked data resources for automatic query expansion. In: Cimiano, P.e.a. (ed.) The Semantic Web: ESWC 2013 Satellite Events, Montpellier, France. pp. 101–112. Springer LNCS, Berlin, Heidelberg (2013)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
3. Ghannay, S., Favre, B., Estve, Y., Camelin, N.: Word embedding evaluation and combination. In: et al., N.C. (ed.) Proc. of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). ELRA, Paris, France (May 2016)
4. Mouromtsev, D., Pavlov, D., Emelyanov, Y., Morozov, A., Razdyakonov, D., Galkin, M.: The simple web-based tool for visualization and sharing of semantic data and ontologies. In: International Semantic Web Conference (Posters & Demos) (2015)
5. Salle, A., Idiart, M., Villavicencio, A.: Enhancing the lexvec distributed word representation model using positional contexts and external memory. CoRR abs/1606.01283 (2016), `http://arxiv.org/abs/1606.01283`
6. Shekarpour, S., Lukovnikov, D., Kumar, A.J., Endris, K.M., Singh, K., Thakkar, H., Lange, C.: Question answering on linked data: Challenges and future directions. CoRR abs/1601.03541 (2016), `http://arxiv.org/abs/1601.03541`