Comparing Pretrained Multilingual Word Embeddings on an Ontology Alignment Task

Dagmar Gromann, Thierry Declerck
Technical University Dresden, DFKI GmbH
Nöthnitzer Str. 46, D-01187 Dresden, Germany, Stuhlsatzenhausweg 3, D-66123, Saarbrücken, Germany,
dagmar.gromann@gmail.com, declerck@dfki.de

Abstract

Word embeddings capture a string's semantics and go beyond its surface form. In a multilingual environment, those embeddings need to be trained for each language, either separately or as a joint model. The more languages needed, the more computationally cost- and time-intensive the task of training. As an alternative, pretrained word embeddings can be utilized to compute semantic similarities of strings in different languages. This paper provides a comparison of three different multilingual pretrained word embedding repositories with a string-matching baseline and uses the task of ontology alignment as example scenario. A vast majority of ontology alignment methods rely on string similarity metrics, however, they frequently use string matching techniques that purely rely on syntactic aspects. Semantically oriented word embeddings have much to offer to ontology alignment algorithms, such as the simple Munkres algorithm utilized in this paper. The proposed approach produces a number of correct alignments on a non-standard data set based on embeddings from the three repositories, where FastText embeddings performed best on all four languages and clearly outperformed the string-matching baseline.

Keywords: word embeddings, ontology alignment, multilingual resources, distributional semantics, comparison, evaluation

1. Introduction

Word embeddings constitute a distributed word representation to leverage the semantics of words by mapping them to vectors of real numbers, where each dimension of the embedding represents a latent feature of the word (Turian et al., 2010). They have been shown to be very successful in many NLP tasks (Mikolov et al., 2013; Camacho-Collados et al., 2016) and also ontology alignment (Zhang et al., 2014). Evaluations of embeddings mostly focus on English standard datasets with high frequency single words (Baroni et al., 2014; Riedl and Biemann, 2017) and the few available multilingual comparisons, such as by Camacho-Collados et al. (2016), usually focus on one type of embedding. This paper proposes the use of a non-standard, domain-specific and multilingual dataset with multi-word expressions to compare three different pretrained embedding repositories. Re-use of existing embeddings is attractive since no training time or expertise in learning embeddings is required.

Embeddings abstract away from the string's surface form, which is not the case with syntactic similarity metrics, such as the Levenshtein edit distance (Levenshtein, 1966), conventionally used in ontology alignment (see Cheatham and Hitzler (2013) for a comparison). A semantic representation of words is important in ontology alignment for two major reasons. First, it allows to consider synonyms and terminological variants. For instance, "Schuhe"@de (*shoes*) should have a lower similarity when compared to "Schule"@de (*school*) than compared to its actual synonym "Fußbekleidung"@de (*footwear*). Second, shortened strings, such as abbreviations, co-occur in the same context as their full form but differ strongly in their surface form. However, the computational cost of training word embeddings increases proportionally with a rising number of languages considered.

In this paper, we evaluate three existing embedding libraries, that is, Polyglot (Al-Rfou et al., 2013), a FastText (Bojanowski et al., 2016), and a word2vec embedding repository (Park, 2017) on an ontology alignment task in four languages: English, Spanish, German, and Italian. We compare the embedding libraries to a Jaccard baseline, which is a string-matching technique that has been shown to perform well on multilingual ontology alignment (Cheatham and Hitzler, 2013). For this task, two multilingual ontologies that exist in all four languages with enough overlap to allow for an alignment are needed and the structure of each ontology should be the same in all its languages, whereas the structure of the two multilingual ontologies might not be exactly the same. We test the application of word embeddings to the mapping of ontology labels of two lightweight industry classification ontologies: the GICS (Global Industry Classification Standard[1]) and the ICB (Industry Classification Benchmark[2]) classification systems. To reduce the Out Of Vocabulary (OOV) ratio, we present and utilize a decomposition for German compounds. Our main contribution is, on the one hand, an experiment with existing embedding repositories on a new kind of task, namely multilingual, domain-specific ontology alignment, and, on the other hand, a cost-effective semantic string matching method for multilingual ontology alignment. We also publish the code utilized in this experiment[3].

As regards structure, we first describe the utilized embedding repositories and ontologies before we specify the chosen methodology including the compound decomposition, vector concatenation, and ontology alignment method. Section 5 quantifies and describes the obtained results, which are discussed in Section 6. We conclude by providing some related approaches and concluding remarks.

[1] https://www.msci.com/gics
[2] http://www.icbenchmark.com/
[3] https://github.com/dgromann/OntologyAlignmentWithEmbeddings

2. Embedding Resources

Polyglot (Al-Rfou et al., 2013) represents a repository of word embeddings in more than 100 languages trained on the Wikipedia corpus for each language (Al-Rfou et al., 2013). The used vocabulary consists of the 100,000 most frequent words in each language and the vectors use a dimensionality of 64. The embeddings were trained using a neural network implementation in Theano (Bergstra et al., 2010). Surface forms of words are mostly preserved, that is, a minimum of normalization is applied, which allows for a querying of the resource without major preprocessing for most languages (German is an exception in our dataset, see 4.1. for details).

The FastText embeddings (Bojanowski et al., 2016) are available in 294 languages, use the most common vector dimensionality of 300 and are trained on the Wikipedia corpus using FastText (Bojanowski et al., 2016). In contrast to the other resources, this embedding library considers subword information of the vocabulary in the corpus. By using a skipgram model, a vector representation for each character n-gram is produced and words are the sum of their character representations.

A repository of word2vec embeddings (Park, 2017), called word2vec for short from now on, available in more than 30 languages was used as a comparative library to fastText and trained on the Wikipedia corpus using the popular word2vec approach (Mikolov et al., 2013). The dimensionality of the vectors is also 300, but no subword information was considered here. The cited word2vec embedding library does not contain English, which is why we trained the English word embeddings using the exact same method as for the other languages with the code provided in the repository (Park, 2017) and also compare to the results obtained with the pretrained embeddings trained on the Google News corpus with dimensionality 300 (Mikolov, 2013).

3. Ontology Alignment Data Set

Industry classification systems enable the comparison of companies across borders and languages. They help investors to diversify their asset portfolios by sorting stocks into sectors and industries. Thus, a portfolio manager can choose stocks from different classes to mitigate the risk of centering extensively on one sector or industry. However, due to numerous and often competing classification systems, the resulting resources are inconsistent on a taxonomic and terminological level. We utilize two widely accepted classification systems in our multilingual alignment method in English, German, Spanish, and Italian: the Global Industry Classification Standard (GICS) and the Industry Classification Benchmark (ICB). The English version of GICS has been translated to other languages, which means that all languages of GICS are fully parallel in structure, which is also true for ICB.

3.1. Global Industry Classification Standard (GICS)

The Global Industry Classification Standard (GICS) represents industry sectors in a lightweight ontology developed by MSCI and Standard & Poor's[4]. The GICS structure consists of 10 sectors, 24 industry groups, 68 industries and 154 sub-industries. GICS is offered in 11 different languages. It contains 256 labels for each language, that is, a total of 2,816 labels. For our experiment dealing with 4 languages, we have thus 1,024 labels. Each sub-industry is equipped with a natural language definition.

3.2. Industry Classification Benchmark (ICB)

The Industry Classification Benchmark (ICB) developed by Dow Jones and FTSE[5] consists of four major levels. There are ten main industries which are subcategorized in an increasingly finer classification into 19 supersectors, 41 sectors and 114 subsectors. Each stock is uniquely classified into one of the 114 subsectors, which consequently uniquely classifies it into one of the sectors, supersectors, and industries. ICB is offered in 14 languages and contains 184 labels for each language, that is, 2,576 labels in total for all languages. For our experiment with four of those languages, we have thus 736 labels. Each subsector is equipped with a natural language definition.

3.3. Comparing ICB and GICS

Both systems classify a company according to its primary revenue source, apply four major levels to their structure and have a comparable number of subcategories. We compare the ten top levels of both hierarchies. Apart from the consumer related sector they seem to be very similar, four of them are even exact string matches. One major difference is to be found in the consumers section. GICS differentiates between staples and discretionary containing both goods and services, whereas ICB distinguishes consumer goods from consumer services. As this regards the top-level classification, it is an important aspect to be considered in the alignment strategy. The terms used to designate equivalent categories differ substantially.

Both classifications apply unique integers for indexing the concepts, to which the labels are associated. While GICS and ICB have both four conceptual levels, they use each a different strategy for encoding the taxonomic positions. GICS adds 2 digits per level ($15 > 1510 > 151010 > 15101010$), while ICB increases the numbers for marking each level ($1000 > 1300 > 1350 > 1353$). Both classifications at times use identical strings to label elements at different levels, e.g. "Banks"@en (ICB8300, ICB8350, ICB8355).

4. Methodology

We use an element-level matching algorithm to calculate the semantic similarity between sets of multilingual labels. The embeddings for each word in each label are retrieved individually and then combined. The similarity is calculated based on a cosine function, the most frequently used similarity metric for embeddings, between the combined vectors of each label in each ontology.

[4]See respectively http://www.msci.com/products/indices/sector/gics/ and http://www.standardandpoors.com/indices/gics/en/us

[5]See http://www.ftse.com/Indices/Industry_Classification_Benchmark/index.jsp

4.1. Preprocessing

Our preprocessing focuses on optimizing short textual sequences for word embedding retrieval since we align ontology labels. To this end, we remove stop words, numbers, and punctuation and decompose complex German compounds to reduce the OOV rate. We decided against further preprocessing, such as lemmatization or stemming.

Normalization

To minimize the OOV value, we repeat the retrieval attempt with several case representations of the word, that is, upper case initial letter, lower casing all letters, and representing all characters as upper case (title casing).

German Noun Decomposition

Compounding languages, such as German, allow a potentially infinite creation of new words, which cannot possibly be covered by a single embedding repository. Decomposition of such word constructs does indeed reduce this potential infinity to a nearly finite set of words that are used in compounds, and which have a much higher probability of being covered by the utilized repositories. But decomposition is not a trivial task. Fortunately, we can use two data sources in our experiment: the GermaNet list of 66,059 compounds together with the explicit description of their components[6]. The second resource we use is generated directly from the data of GICS and ICB: a program first traverses all the labels and definitions used in the German version of the classification systems and collects all the words used. In a second run, words that are re-used in larger strings are marked as components of a compound and thus also as independent words. This strategy allows to significantly increase the number of German words covered by the individual repositories and bring them closer to the other languages (see Table 1 for details).

4.2. Vector Composition

In order to obtain the best estimation of the similarity of two labels $sim(label_1, label_2)$ each label is split into its k number of individual words. We then retrieve the vector representation $\vec{v}_i$ for each word from the individual embedding repositories, so we query for the vector $\vec{v}_i : \forall w_k \in label_i$ for all words w_k in each label $label_i$.

Since OOV occurrences are possible, we need to compose the vectors in a way that allows for the indication of missing words in a longer sequence of words representing the label. To this end, we adapt the concept of lexical semantic vectors (Konopik et al., 2016). We create a combined vocabulary $L = unique(label_1 \cup label_2)$ of all unique words in $label_1$ of GICS and $label_2$ of ICB as represented in Figure 1. Similarity of the first word of L and the first word of $label_1$ is calculated as the cosine metric of their respective embeddings, which we retrieve from the embedding repositories (in this example we used Polyglot embeddings). Then, the first word of L is compared to the second word of $label_1$ until the first word of L has been compared to all words of $label_1$. We create a vector $\vec{m}$ that contains the maximum cosine similarities between each word of L and all other words in $label_1$. For instance, the "Renewable" embedding in L obtains the highest similarity (1.0) with the "Renewable" embedding in $label_1$ of GICS. The process is iterated for the second word of L to obtain the second value of vector $\vec{m}$. To obtain $\vec{n}$, the words of L are compared to all words of $label_2$ as described for $label_1$. For instance, the "Renewable" embedding of L obtains the highest similarity value (0.695) with the "Conventional" embedding of $label_2$ of ICB. The dimensionality of the resulting vectors $\vec{m}$ and $\vec{n}$ depends on the number of words in L. The similarity $sim(label_1, label_2)$ is calculated as the cosine value between $\vec{m}$ of GICS and $\vec{n}$ of ICB for each of their labels.

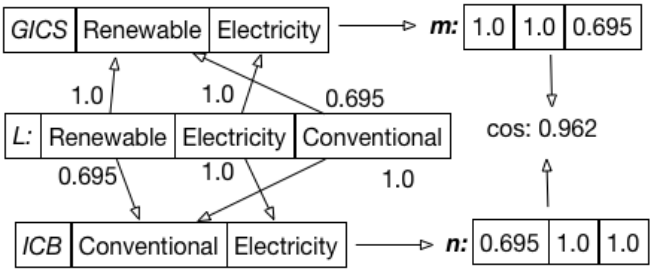

Figure 1: Vector Comparison Method to Measure Semantic Distance

If a word is not in the vocabulary of the embedding repository, we do not have an embedding to calculate the cosine similarity with other word embeddings. To solve this problem, we fill the slot of this word in the final vector $\vec{m}$ respectively $\vec{n}$ with the average of all other values[7]. This means that we sum all calculated cosine similarities for in-vocabulary words of the label with an OOV and divide the sum by the number of in-vocabulary elements of the same label. The resulting value provides us with the cosine-similarity value in vector $\vec{m}$ respectively $\vec{n}$ for the OOV words as depicted Figure 2, where "Nondurable" is not in the embedding library, which in this example case is the Polyglot library.

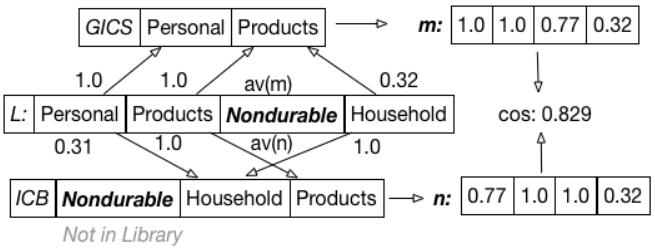

Figure 2: Handling OOV in Vector Composition

4.3. Ontology Alignment Task

Two ontologies modeling the same domain frequently differ due to design choices of the engineers. To enable their

[6] http://www.sfs.uni-tuebingen.de/lsd/compounds.shtml

[7] We tested with values of -1, close to zero, and the average similarity of all other words in the same label for OOV words and found a better performance using the average similarity of all other values in the vector.

re-use, extension, and comparison, ontology alignment is useful. It has been shown that string similarity metrics alone can achieve results competitive to state-of-the-art ontology alignment systems (Cheatham and Hitzler, 2013). For instance, the Jaccard distance, which is based on the number of words two strings have in common divided by the total number of words of the two strings, has been found to be very effective in multilingual ontology alignment scenarios (Cheatham and Hitzler, 2013). For this reason we chose Jaccard as the baseline for measuring the performance of the proposed embedding-based approach. However, linguistic condensation strategies, such as compounding or abbreviations, and the lack of semantic context can pose a serious challenge to string-based ontology alignment methods.

Ontology alignment represents the problem of identifying a set of correspondences $\{e_1, e_2, sim | e_1 \in O_1, e_2 \in O_2\}$ where e_1 is an element in O_1 and e_2 is an element of O_2 and sim represents the confidence of each individual correspondence in the set of alignments. In this first experiment, our entities e correspond to class labels only. The correspondence is represented as an equivalence relation between two entities $e_1 \equiv e_2$, where each entity is represented by its natural language label. This correspondence relation is based on the assumption that a one-to-one matching between entities exists. We repeat the alignment process for each language-specific ontology, since they are provided separately.

To align the two input ontologies in each language, we employ the Munkres assignment algorithm (Munkres, 1957). Given a non-negative `n x n` matrix it optimizes the alignment of the i-th row with the j-th column. Our similarity matrix represents all GICS labels as rows and all ICB labels as columns and the value if the i-th row and j-th column represents their similarity $sim_{ij} = (label_i, label_j)$.

4.4. Cross-Lingual Experiment

Cross-lingual approaches to ontology labels range from benefiting from an alignment across languages to complement elided content in labels (e.g. Gromann and Declerck (2014)) to cross-lingual ontology matching (e.g. Spohr et al. (2011)). Even though our paper focuses on a multilingual alignment process, we conducted a small experiment utilizing cross-lingual information. In order to benefit from the parallel structure across languages, we aggregate each similarity value and related ICB identifier from each language for a given GICS element. In other words, for each GICS element we obtain four ICB ids and four similarity values. We first count the frequency of occurrence of each ICB id in the list of four ids. If the id occurs more than once, it is selected as the chosen alignment target for this specific GICS element. If no id occurs more than once, we select that ICB id with the highest associated similarity value. For instance, the element GICS 15102010 with label "Construction Materials" is mapped to the following ICB elements[8]: "Heavy Construction" (ICB 2357) in English, "Household Goods & Home Construction" (ICB 3720) in Italian, "Building Materials & Fixtures" (ICB 2353) in German and Spanish. The assigned similarity values are $[2357 : 0.812, 3720 : 0.797, 2353 : 0.884, 2353 : 0.924]$ where the last two values correspond to first the German and then the Spanish cosine value. In this example our simple algorithm selects ICB 2353 because it occurs twice, which is a correct mapping that corresponds to the manual alignment. If it had occurred only once, our basic approach would have still selected ICB 2353 since it also has the highest similarity value with 0.924 in Spanish.

[8]For a better understanding of the example we only provide the English labels of the ontology elements here.

4.5. Evaluation

In order to evaluate our method, two experts created a manual alignment of GICS and ICB elements based on their labels and natural language descriptions in English. This monolingual mapping is sufficient, since the other language versions of each standard are translated from English and remain structurally equivalent to the source language. We calculate the inter-annotator agreement to be 0.75. Alignments that reached no agreement were evaluated by a third expert. The automatically created alignment is compared to the manually created alignment, which is how we obtain the metrics presented in Section 5. We only consider elements that can directly be mapped across the two ontologies - elements that require a one-to-many mapping are ignored for this first experiment. For instance, "Marine" (GICS 20303010) in GICS is defined as any maritime transportation of goods and passengers excluding cruise ships, while ICB differentiates between "Marine Transportation" (ICB 2773) for commercial markets, such as container shipping, and "Travel & Tourism" (ICB 5759) providing passenger transportation including ferry companies. Thus, a direct mapping of the GICS concept to one ICB element could not be established in this particular example. Our final dataset contains 155 labels from each ontology in each language, that is, a total of 620 labels per ontology across all languages combined.

5. Results

To quantify our comparison, we first evaluate the coverage of our vocabulary by each embedding library. We continue to quantify the performance of all embedding libraries by F-score on the described dataset and in comparison to a string-matching baseline.

5.1. OOV scores

To provide a better explanation of the F-measure results, Table 1 provides statistics on Out Of Vocabulary (OOV) scores for each set of multilingual embeddings across both ontologies, ICB and GICS, without duplicates. To increase coverage, each word of a label not directly available in the library was queried again with different case settings, i.e., lower, upper, and title case. For German, the compound decomposition described in Section 4.1. was conducted.

Reasons for an inability to find specific words from a label in the embeddings library varied strongly with each library. Polyglot had difficulty finding abbreviations, such as "REITs" (Real Estate Investment Trusts), and unusual compounds, such as "Multi-Utilities". FastText only struggled with unusual compounds in all languages but Italian, where nouns with articles constituted the biggest problem,

	Polyglot	FastText	word2vec
English	21	5	41
Italian	22	17	43
German*	70 (198)	38 (99)	125 (218)
Spanish	21	4	43

*value in brackets is before decomposition

Table 1: Word Coverage Across Resources

e.g. "all'Ingrosso". Articles where no problem for Polyglot and FastText handled abbreviations without any difficulty. So the type of further preprocessing that would be required to further increase coverage differs with the embedding library. The extraordinarily high number of OOVs in German can be attributed to high frequent ellipses in the vocabulary, such as "Abfall-" (waste) in the label " Abfall- und Entsorgungsdienstleister" (ICB 2799 "Waste and Disposal Services"), and complicated compounds, such as "Arzneimitteleinzelhändler" (ICB 5333 "Drug Retailer"). Results for word2vec could be improved for Spanish and Italian by lemmatizing, however, for German the major problem are the compounds that are not considered in the library. Using the proposed decomposition method, we could improve on this situation as shown in Table 1 where the values in brackets represent the OOV words prior to decomposing.

5.2. Alignment Statistics

A generated similarity matrix calculated on the basis of the individual word embeddings is submitted to the simple Munkres algorithm to compare the obtained alignment to the manual gold standard alignment. Table 2 quantifies the comparison across embedding repositories and to a Jaccard baseline. The English embeddings of Polyglot seem to largely outperform all other tested languages of the same repository. The same could be observed for the other repositories in Table 2. All embeddings were trained on the Wikipedia corpus but different methods were utilized in obtaining the embeddings. It can be seen from Table 2 that FastText and its encoded subword information outperforms the other embedding representations. However, the simple string-matching Jaccard similarity baseline outperforms Polyglot in all languages and outperforms the word2vec pretrained embeddings in German.

	Jaccard	Polyglot	FastText	word2vec
English	0.692	0.652	**0.830**	0.760 (0.826*)
Italian	0.488	0.385	**0.686**	0.517
German	0.473	0.434	**0.652**	0.418
Spanish	0.495	0.360	**0.745**	0.582
All	0.678	0.675	**0.812**	0.750

* Using embeddings trained on the Google News Corpus.

Table 2: Embedding Comparison by F-Score on Ontology Alignment Task

The first four rows of Table 2 quantify the results of all four languages, while row five describes a cross-lingual optimization experiment. Results of this first experiment quantified in Table 3 show that FastText has the highest correspondence across all languages, since it has the highest number of recurring ICB ids across languages. Incidentally FastText also has the highest F-measure in all languages, while Polyglot, the repository with the lowest F-measure of all methods also has the lowest correspondence of ICB target ids across languages. This simple cross-lingual comparison will be replaced by a more principled cross-lingual approach in future work.

	Jaccard	Polyglot	FastText	word2vec
most_common	89	71	**138**	107
high_sim	66	84	17	47

Table 3: Cross-Lingual Experiment

Interestingly this cross-lingual experiment leads to quite different F-scores as can be seen from the last row of Table 2. In general, the composition across languages has a positive impact on the F-score in Italian, German, and Spanish in all cases. However, Jaccard, FastText, and word2vec provide better results in the monolingual English setting, while the composition of results across languages outperforms the English results for Polyglot.

6. Discussion

When using pretrained embeddings, the handling of OOV is an important issue. Decomposition and preprocessing for the German data proposed in this paper could still be refined to include for instance ellipses resolution (Gromann and Declerck, 2014). While FastText provides a comparatively high coverage of vocabulary across all languages, the other libraries could benefit from some more refined preprocessing of the GICS and ICB labels. Nevertheless, the performance of the pretrained embeddings on domain-specific multi-word labels of lightweight ontologies is very promising, in particular FastText.

FastText outperforms all the other three methods in all languages. We attribute the success of this embedding library to two main factors: i) the library has fewer OOV words than the other embedding repositories, ii) subword information is considered when training the embeddings and each embedding is the aggregated result of its character n-grams. It seems as if this more morphologically oriented type of embedding in FastText is more adequate for domain-specific multi-word expressions as found in ontologies. FastText is also the one repository with most corresponding results across all languages, as our small cross-lingual comparison shows.

One of the main assumptions for Polyglot's performance below the Jaccard baseline is the high degree of OOV words in the library in all languages. However, word2vec has a higher OOV rate and provides better results than Polyglot. Thus, it can be followed that the settings and parameters of the training method of the embeddings make a difference since all three repositories are trained on the Wikipedia corpus and applied to the same task in this paper but differ in resulting F-score. Those parameters include the chosen dimensionality of the embedding, which in case of Polyglot

is 64 as opposed to 300 in the other libraries. The factor of dimensionality has been shown to have a substantial impact on the accuracy that can be obtained with the vector representations (Mikolov et al., 2013). Of course, also the corpus chosen for training has a large impact as can be seen by the comparison to the word2vec embeddings trained on the Google News Corpus.

Similarity in the vector space is not necessarily semantic similarity, but might be any type of relatedness. Thus, labels such as "Renewable Electricity" and "Conventional Electricity" might obtain a very high cosine value as shown in our vector composition example even though they are not synonymous and in fact almost are opposites. We would expect this problem to be more prominent in this highly domain-specific scenario, however, as can be seen from the good results obtained by FastText, this difference in similarity relation does not seem to be detrimental to the overall application of word embeddings to a label-centric ontology alignment task.

In our results, English embeddings obtain better results than embeddings used in other languages to align ontology labels. Both ontologies were originally produced by English-speaking companies in English and then translated to the other languages. It has been shown that non-native language and translations are closer to each other than they are to the native language (Rabinovich et al., 2016). Thus, this difference in accuracy cannot necessarily be attributed to the embeddings but more likely to the input labels. For this purpose it would be interesting to repeat the experiment on a multilingual and structurally parallel standard dataset.

We believe that this method can also be applied to other interesting scenarios. The similarity between the labels also hints at the similarity between the entire resources. Thus, this method could potentially be used to find similar resources in a repository of ontologies, which is offset in comparison to simple string matching by embeddings and multilinguality. It can also be used to do a lightweight checking of structural problems in ontologies. For instance, it can be considered bad practice to assign identical or almost identical labels to different concepts. Identical surface forms and equivalences can be detected using the proposed method.

7. Related Approaches

Two main lines of research are related to the proposed comparison of word embeddings on the task of multilingual ontology alignment: (i) comparisons of word embeddings in general, (ii) use of word embeddings on ontology alignment tasks. Most embedding comparisons focus on high frequent English words for various tasks (Baroni et al., 2014; Riedl and Biemann, 2017) or if multilingual, evaluate specific embeddings (Camacho-Collados et al., 2016). However, approaches for multilingual, domain-specific multi-word expressions are hard to find for embedding comparisons. The use of word embeddings in ontology-related tasks is a rather recent development. Zhang et al. (Zhang et al., 2014) utilize word embeddings they learn from Wikipedia texts to match the OAEI 2013 benchmark ontologies and three real-world ontologies including Freebase in English. In their embedding comparison, the ones trained solely on Wikipedia performed best but in an overall evaluation a hybrid embedding and edit-distance method outperformed the others.

8. Conclusion

In this initial experiment we evaluate the use of pretrained word embeddings in four languages for the task of real-world domain-specific ontology alignment. We propose a method that is able to handle missing embeddings for individual words applied to a non-standard dataset. One of the reasons for this decision is our interest in domain-specific labels and multilingual, aligned contents. Furthermore, we were interested in a real-world scenario that also has a practical value for industry. Pretrained embedding libraries achieve promising results, particularly FastText with a greater consideration for morphological structures seems very apt for the task of string-based ontology alignment. Future work will consider the integration of taxonomic and axiomatic knowledge from the ontologies with the embeddings to improve the alignment results as well as the utilization of existing knowledge-rich embeddings, e.g. ones that integrate ConceptNet structures into their representation. Furthermore, we are interested in the further utilization of cross-lingual information in the alignment process, such as treating OOV words in one language by using word embeddings available in another language.

Acknowledgement

The DFKI contribution to this work has been partially funded by the BMBF project "DeepLee - Tiefes Lernen für End-to-End-Anwendungen in der Sprachtechnologie" with number 01-W17001, and the project QT21, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 645452.

9. Bibliographical References

Al-Rfou, R., Perozzi, B., and Skiena, S. (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.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL (1)*, pages 238–247.

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*, pages 1–7.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Camacho-Collados, J., Pilehvar, M. T., and Navigli, R. (2016). Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64.

Cheatham, M. and Hitzler, P. (2013). String similarity metrics for ontology alignment. In *International Semantic Web Conference*, pages 294–309. Springer.

Gromann, D. and Declerck, T. (2014). A cross-lingual correcting and completive method for multilingual ontology labels. In *Towards the Multilingual Semantic Web*, pages 227–242. Springer.

Konopik, M., Prazák, O., Steinberger, D., and Brychcín, T. (2016). Uwb at semeval-2016 task 2: Interpretable semantic textual similarity with distributional semantics for chunks. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 803–808.

Levenshtein, V. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38.

Rabinovich, E., Nisioi, S., Ordan, N., and Wintner, S. (2016). On the similarities between native, non-native and translated texts. *arXiv preprint arXiv:1609.03204*.

Riedl, M. and Biemann, C. (2017). There's no 'count or predict' but task-based selection for distributional models. In *12th International Conference on Computational Semantics (IWCS)*. ACL (W17).

Spohr, D., Hollink, L., and Cimiano, P. (2011). A machine learning approach to multilingual and cross-lingual ontology matching. In *International Semantic Web Conference*, pages 665–680. Springer.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Zhang, Y., Wang, X., Lai, S., He, S., Liu, K., Zhao, J., and Lv, X. (2014). Ontology matching with word embeddings. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 34–45. Springer.

10. Language Resource References

Al-Rfou, R., Perozzi, B., and Skiena, S. (2013). Polyglot: Distributed word representations for multilingual nlp. https://sites.google.com/site/rmyeid/projects/polyglot.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Pre-trained word vectors. https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md.

Mikolov, T. (2013). word2vec: Tool for computing continuous distributed representations of words. https://code.google.com/archive/p/word2vec/.

Park, K. (2017). Pre-trained word vectors of 30+ languages. https://github.com/Kyubyong/wordvectors.