Bull. Nov. Comp. Center, Comp. Science, 45 (2021), 49--58

Methods of processing textual information in entity alignment algorithms

Daniil Gusev, Zinaida Apanovich

Abstract. Entity alignment algorithms aim to find equivalent entities in cross-lingual knowledge graphs, which is important for the task of obtaining information about real-world objects. Recently, several studies have been conducted on entity alignment algorithms on various datasets. Algorithms using information about entity names have shown a wide range of results. In this paper, we have conducted a study of this phenomenon. Work has been done to improve the quality of matching cross-language entity names in vector space. Also, experiments with the modern models of processing natural languages have been carried out. The information obtained has led to a significant increase in the accuracy of entity alignment on the English-Russian dataset.

Keywords: entity alignment, knowledge graphs, vector representation, matrix ordering, language models, multilingual knowledge bases

Introduction

Knowledge graphs are a modern form of representing information about the world. They consist of unique entities and relationships between them. Triples (subject entity, relation, object entity) or (subject entity, attribute, literal value) are used to represent facts. The first type is called relational and is used to describe relationships between entities. Its example for the fact "Novosibirsk is part of Russia" is (dbr:Novosibirsk, dbo:country, dbr:Russia). The second type is called attribute and is used to describe the properties of an entity. Its example for the fact "Novosibirsk was founded in 1893" is (dbr:Novosibirsk, dbp:establishedDate, 1893^^xsd:integer), where "dbp:establishedDate" is an attribute (property) and "1893^^xsd:integer" is a literal (value).

The examples of the applications of knowledge graphs (KGs) include content recommendation systems, drug discovery, investment market analysis, and semantic search. Moreover, the more powerful the basic knowledge graph, the higher the quality of applications based on it.

The knowledge graph can be supplemented by combining it with others. This is achieved by searching for entities in knowledge graphs that refer to the same object of the real world. An example is the entity "Austria" in the English graph and "А в с т р и я" in the Russian one. This direction is called entity alignment (EA). In some literature, it may be referred to as entity matching.

Recently, embedding-based entity alignment algorithms have become widespread. The idea is to obtain the descriptions of KG in the form of low-dimensional vectors in a way

such that the semantic relatedness of entities is captured by their position in the vector space [1]. Potentially, this can mitigate the linguistic and schematic heterogeneity between independently created knowledge graphs.

Hits@K and MRR metrics are used to analyze the results of entity alignment algorithms. Hits@K means that entities from the first knowledge graph and equivalent entities from the second knowledge graph are among the nearest k neighbors. At the same time, the Hits@1 metric is considered the most indicative, since it is equivalent to precision. The MRR (Mean Reciprocal Rank) represents the average of the reciprocals of the numbers of the correct answers in the list of supposed entities. It can be considered as a soft version of Hits@1, which is less sensitive to outliers [2]. For both metrics, the values range from 0 to 1, where a higher number indicates better accuracy.

An extensive study of entity alignment algorithms was carried out on the English-French dataset [3]. Good results were shown by MultiKE and RDGCN. However, on the English-Russian dataset, they have a significant decrease in accuracy [4]. To study this pattern, entity alignment algorithms were studied and a number of experiments were carried out. The application of the obtained information led to an increase in the accuracy of the algorithms.

1. Embedding-based algorithms for entity alignment

Most embedding-based entity alignment algorithms boil down to two steps:

1. Generation of embeddings for entities and relations.
2. Mapping of these embeddings into a single vector space [5] or into different vector spaces [6].

In the first case, the question of whether two entities from different graphs are equivalent (corresponding to the same real-world object) is solved by comparing their vectors, for example, by calculating the Euclidean distance or cosine proximity. When mapping the entities of two knowledge graphs to different vector spaces, it is also necessary to find the correspondence matrix between the vectors of these two spaces.

Modern EA algorithms rely mainly on structural information in knowledge graphs, that is, relational triplets. They are based on the assumption that equivalent entities must have similar graph neighborhoods. Initially, the translational approach prevailed. It considered the relation vector as a shift vector from the vector of one entity to the vector of the second entity. One of the best representatives of the translational approach is MultiKE [7]. MultiKE builds three types of embeddings for each entity, using different "views": name view, relational view, and attribute view. Each of the "views" is built according to its own algorithm. The final vector representation of an entity can be obtained by combining the three views mentioned in various ways.

In recent years, algorithms for constructing entities embeddings based on graph convolutional networks (GCNs) have become extremely popular. These methods give very good results, but their main disadvantage is their extreme complexity, significant computation time, and poor interpretability. The representative of this algorithm is RDGCN [8]. To build embeddings, RDGCN uses not only the structure of the original knowledge graphs (primal entity graph), but also auxiliary graphs that are dual with respect to the original graphs (dual relation graph), whose vertices are the edges of the original graphs. To implement the interaction between the primal knowledge graphs and dual relational graphs,

the mechanism of graph attention networks (GAT) is used. The resulting embeddings are then fed into graph convolutional networks to extract information about the vertex structure.

More recently, an extremely simple algorithm for entity alignment called SEU [9] (Simple but Effective Unsupervised EA method) has appeared. It does not use neural networks. The main idea of SEU is to reduce the entity alignment problem to the well-known assignment problem. There are many ways to solve it. The main assumption of this algorithm is that the adjacency matrices of two knowledge graphs are isomorphic. In this case, the adjacency matrix of the original graph can be converted to the adjacency matrix of the second graph by rearranging rows or columns.

However, most recent research indicates that current EA algorithms are not capable of producing satisfactory results from relational triples alone. This is especially noticeable if the dataset has a distribution of entity degrees which is close to real KG. In particular, it is known that approximately half of the entities in real KGs are associated with less than three other entities [10].

This observation makes it important to use additional information such as entity names and to combine entity name information with structural information. The names of entities must be brought to a common language, and then compared. There are two basic approaches for comparing entity names: based on string similarity and based on semantic similarity. Semantic similarity methods can be divided into two groups: generation of vector representations based on sentences or individual words (word2vec, glove models). However, due to the limitations of the dictionaries used, the situation often arises that the desired word is missing. In this case, the vector representation of the word is built on the basis of the characters included in its composition (fastText, name-BERT models).

The previously mentioned entity alignment algorithms also have their own methods for processing textual information. Their main stages are: reading the pre-trained model, data tokenization and the formation of vector representations. However, there are also differences in the ways of processing unrecognized words and combining vectors.

The following are the main features of the processing methods from the entity alignment algorithms. For further reference, they have also been numbered.

Method 1 is applied in MultiKE. Wiki-news-300d-1M is used as a pre-trained model. Unrecognized word vectors are generated by summing the character vectors obtained using word2vec. A neural network is used to combine word vectors.

Method 2 is applied in RDGCN. Wiki-news-300d-1M is also used as a pre-trained model. Clears the input data from special characters. Unrecognized words are assigned a null vector. Word vectors are combined by summation.

Method 3 is applied in SEU. Glove.6B.300d is used as a pre-trained model. The input data is reduced to lowercase. The vectors of unrecognized words are set randomly. In addition to the vector of each word, a vector of bigrams is applied. The union is made by calculating the arithmetic mean.

2. Experiments with embeddings of entity names

2.1. Cross-lingual datasets

Modern open knowledge graphs contain a large amount of information. However, this

leads to an increase in the complexity of obtaining the results of entity alignment algorithms. To solve this problem, 15,000 pairs of entities were selected. The IDS algorithm was used for their formation [5]. It simultaneously removes entities in two knowledge graphs with alignment by interlanguage links until the desired size is reached. At the same time, the degree distribution similar to the original KGs is preserved.

The generated datasets are represented by two versions. The result of direct IDS application is marked V1. A twice denser set is marked V2. For its generation, entities with fewer than five connections were removed at random beforehand. After that, IDS was applied.

The sources are multilingual versions of DBpedia [1]. In particular, they contain owl:sameAs relationships necessary to obtain aligned entity pairs. The English-French and English-Russian cross-language datasets were selected as the target ones. DBP-15K EN-FR (V1, V2) is taken from the OpenEA library. The DBP-15K EN-RU (V1, V2) set is generated according to the same principles and is available for free download[2].

According to Table 1, there are some differences in the datasets used. Thus, the Russian-language graph contains a smaller number of unique relationships and attributes. There is also a noticeably smaller number of connections between entities. This is worth considering when analyzing the results.

Table 1. Dataset statistics

Dataset	KG	15K (V1)				15K (V2)			
		Rel.	Att.	Rel tr.	Att tr.	Rel.	Att.	Rel tr.	Att tr.
EN-FR	EN	267	308	47334	73121	193	189	96318	66898
	FR	210	404	40864	67167	166	221	80112	68778
EN-RU	EN	163	173	43796	76959	141	147	76617	75135
	RU	66	52	30489	54517	57	46	56399	56455

2.2. Translation of entity names in EA algorithms

The previously considered entity alignment algorithms use pre-trained language models of words. This makes it much easier to combine similar meanings into a single semantic space. At the same time, there are cases when the desired words are not contained in the model. This problem is hardly noticeable for languages with similar morphology, such as English and French. In the case of combining English and Russian words into a single vector representation, it makes sense to use machine translation.

To solve this problem, we have developed an automatic translation tool based on the Google Translate API. The input is provided with a label of the language from which the translation will be performed and entity names. English is selected as the target language.

1 https://wiki.dbpedia.org/downloads-2016-10/

2 https://www.dropbox.com/sh/4oh3nkzwdr1w4dv/AACZ4v8jCdR7Y4mDtS654Bega?dl=0

Then the input data is divided into packets of 3,500 characters. This is due to a limitation of the Google Translate API. Next, each package is converted into strings, and the names are separated from each other to exclude their getting into the context. After that, by accessing a third-party server, the packets are translated and the original data sequence is restored. The result is passed to the method of generating a vector representation.

The "Difference" column in Table 2 shows the change in accuracy according to the Hits@1 metric, depending on the application of machine translation. The English-French knowledge graph has a slight change in all algorithms.

The English-Russian knowledge graph has a significant increase in accuracy. However, the results obtained are still slightly lower than the English-French ones. Most likely, this is due to the relational structure and fewer connections.

Machine translation has had the greatest impact on SEU, which suggests that this algorithm relies heavily on the textual features of entity names. For RDGCN, it was possible to achieve indicators similar to the English-French knowledge graph. The use of translation has also led to an increase in the accuracy of MultiKE. However, this algorithm has the smallest change among the presented ones.

Table 2. Results of entity alignment algorithms depending on the application of translation

Algorithm	Dataset	Translation	Hits@1	Hits@10	MRR	Difference
MultiKE	EN-FR-15K (V1)	-	0,741	0,836	0,774	
MultiKE	EN-FR-15K (V1)	+	0,806	0,885	0,835	0,065
MultiKE	EN-FR-15K (V2)	-	0,855	0,921	0,878	
MultiKE	EN-FR-15K (V2)	+	0,893	0,956	0,915	0,038
MultiKE	EN-RU-15K (V1)	-	0,315	0,457	0,364	
MultiKE	EN-RU-15K (V1)	+	0,520	0,666	0,570	0,205
MultiKE	EN-RU-15K (V2)	-	0,453	0,623	0,510	
MultiKE	EN-RU-15K (V2)	+	0,617	0,770	0,670	0,164
RDGCN	EN-FR-15K (V1)	-	0,770	0,892	0,813	
RDGCN	EN-FR-15K (V1)	+	0,771	0,893	0,813	0,001
RDGCN	EN-FR-15K (V2)	-	0,862	0,948	0,895	
RDGCN	EN-FR-15K (V2)	+	0,871	0,951	0,903	0,009
RDGCN	EN-RU-15K (V1)	-	0,396	0,597	0,460	
RDGCN	EN-RU-15K (V1)	+	0,744	0,882	0,792	0,347
RDGCN	EN-RU-15K (V2)	-	0,537	0,717	0,599	
RDGCN	EN-RU-15K (V2)	+	0,844	0,923	0,882	0,307
SEU	EN-FR-15K (V1)	-	0,989	0,998	0,992	
SEU	EN-FR-15K (V1)	+	0,995	1,000	0,997	0,006
SEU	EN-FR-15K (V2)	-	0,992	0,999	0,994	
SEU	EN-FR-15K (V2)	+	0,996	1,000	0,997	0,004
SEU	EN-RU-15K (V1)	-	0,301	0,348	0,318	
SEU	EN-RU-15K (V1)	+	0,972	0,995	0,981	0,672
SEU	EN-RU-15K (V2)	-	0,424	0,483	0,445	
SEU	EN-RU-15K (V2)	+	0,990	0,998	0,993	0,566

2.3. Comparison of the results of textual information processing methods

To compare the methods of generating vector representations based on EN-RU-15K (V1) and using preliminary translation, we obtained the visualizations of the results. To obtain a two-dimensional space, the t-SNE tool was used.

In the presented images, the English names of entities are blue, and the Russian ones are red. This allows us to evaluate the effectiveness of the method of generating vector representations. The high degree of color overlap indicates that semantically related data presented in different languages are located together. The presence of single-color clusters indicates that the method could not establish a cross-language correspondence.

According to the results of the 1st processing method (Figure 1a), it is clear that the Russian names of entities are at a distance from the English ones. Machine translation partially solves this problem (Figure 1b). However, this vector representation has pronounced language clusters, which indicates low accuracy.

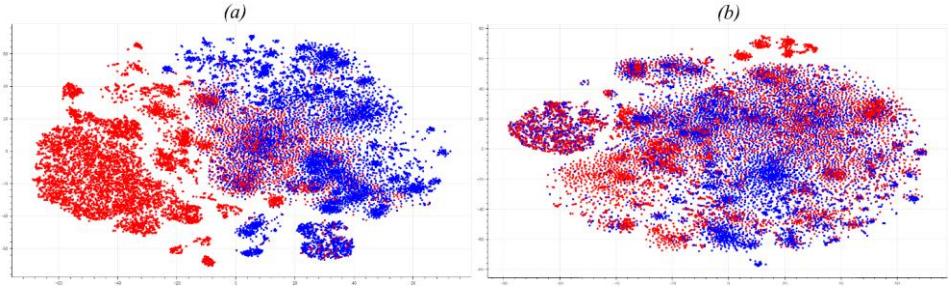

Figure 1. Embeddings of the entity names of method 1:
a is without translation; *b* is with translation

In the vector representation of the entity names of Method 2 (Figure 2a), there is an ellipsoid cluster in the lower left corner. This arose due to the nulling of vectors of words for which the 2nd method did not find values in the pre-trained model. Otherwise, this vector representation has a greater degree of overlap compared to Method 1. The smallest number of language clusters is observed in the result obtained using the 3rd generation method (Figure 2b).

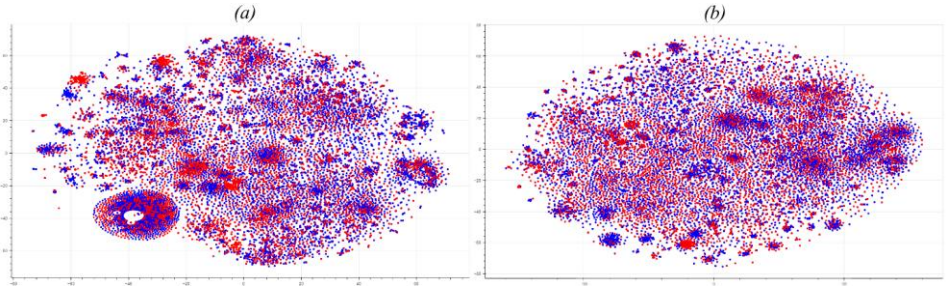

Figure 2. Embeddings of entity names from various methods:
a is the 2nd method; *b* is the 3rd method

Modern XLNet and LaBSE natural language processing models have been chosen as the alternative methods for generating vector representations.

The purpose of the XLNet model is to study distributions for all permutations of words in a given sequence [11]. It has high accuracy in the English DBpedia corpus. Embeddings are formed within the framework of only one language; therefore, to solve our problem, it was necessary to apply machine translation first.

LaBSE generates language-independent embeddings of sentences based on BERT. This is achieved by combining the capabilities of masked and translation language modeling [12]. The authors of the model claimed 90% accuracy in matching English-Russian texts.

The XLNet vector representation (Figure 3a) has an extremely low degree of overlap. The most semantically related entity names are at a distance from each other. The opposite pattern is observed in LaBSE (Figure 3b). The model was able to match entity names without using translation.

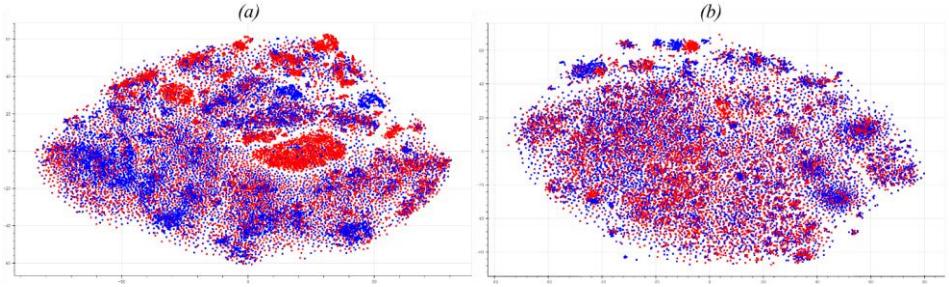

Figure 3. Embeddings of the entity names of natural language processing models: *a* is XLNet; *b* is LaBSE

2.4. The influence of textual information processing methods on EA algorithms

In addition to translating Russian-language entity names in English, experiments were conducted with several methods of generating vector representations of entity names.

The results of entity alignment algorithms based on various processing methods are presented in Table 3. The specified values were obtained on the EN-RU-15K (V1) dataset. They clearly show that the 3rd processing method has turned out to be the most effective. MultiKE and RDGCN based on it exceeded the accuracy values of the original publications.

The results of applying XLNet and LaBSE models to MultiKE are not indicated due to the lack of computing resources for constructing the vector representations of literals. Conclusions about their effectiveness are made based on the values from other algorithms.

The XLNet model has turned out to be unsuitable for the formation of vector representations. The results of algorithms based on it are close to the values obtained without translation. LaBSE has performed well, proving to be more efficient than the processing Methods 1 and 2.

Table 3. Results of EA algorithms depending on the method of processing text information

Algorithm	Method	Hits@1	Hits@10	MRR
MultiKE	1st	0,520	0,666	0,570
MultiKE	2nd	0,699	0,813	0,737
MultiKE	3rd	**0,812**	**0,891**	**0,841**
RDGCN	1st	0,680	0,828	0,733
RDGCN	2nd	0,744	0,882	0,792
RDGCN	3rd	**0,848**	**0,935**	**0,881**
RDGCN	XLNet	0,434	0,530	0,467
RDGCN	LaBSE	0,754	0,859	0,792
SEU	1st	0,881	0,948	0,905
SEU	2nd	0,874	0,954	0,905
SEU	3rd	**0,972**	**0,995**	**0,981**
SEU	XLNet	0,325	0,455	0,369
SEU	LaBSE	0,949	0,984	0,962

Conclusion

In this paper, we have studied the influence of methods for constructing the vector representations of entity names and literals on the results of algorithms. The use of translation has led to a significant improvement in accuracy. For two of the three algorithms studied, we have managed to obtain values comparable to the results on the English-French dataset.

Further study has revealed the most effective method of constructing vector representations. Its application has led to a further increase in accuracy. The results obtained for the English-Russian dataset are not only comparable with the original results for the English-French dataset, but exceed them.

The contribution of the application of two modern models of natural language processing is investigated. At the moment, it is possible to obtain a high-quality vector representation of entity names based on these models.

References

[1] Bordes A., Usunier N., Garcia-Durán A., et al. Translating embeddings for modeling multi-relational data // Proc. of the 26th International Conference on Neural Information Processing Systems. – 2013. – Vol. 2. – P. 2787 –2795. DOI:10.5555/2999792.2999923

[2] Rossi A., Barbosa D., Firmani D., et al. Knowledge graph embedding for link prediction: a comparative analysis // ACM Transactions on Knowledge Discovery from Data. – 2021. – Vol. 15. – P. 1 – 49. DOI:10.1145/3424672

[3] Sun Z., Zhang Q., Hu W., et al. A benchmarking study of embedding-based entity alignment for knowledge graphs // Proc. of the VLDB Endowment. – 2020. – Vol. 13. – P. 2326 –2340. DOI:10.14778/3407790.3407828

[4] Gnezdilova V.A., Apanovich, Z.V. Russian-English dataset and comparative analysis of algorithms for cross-language embedding-based entity alignment // Journal of Physics: Conference Series. – 2021. – Vol. 2099. DOI:10.1088/1742-6596/2099/1/012023

[5] Xu K., Wang L., Yu M., et al. Cross-lingual knowledge graph alignment via graph matching neural network // Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. – 2019. – P. 3156 –3161. DOI:10.18653/v1/P19-1304

[6] Chen M., Tian Y., Chang K., et al. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment // Twenty-Seventh International Joint Conference on Artificial Intelligence. – 2018. – P. 3998 –4004. DOI:10.24963/ijcai.2018/556

[7] Zhang Q., Sun Z., Hu W., et al. Multi-view Knowledge graph embedding for entity alignment // Proc. of the Twenty-Eighth International Joint Conference on Artificial Intelligence. – 2019. – P. 5429 –5435. DOI:10.24963/ijcai.2019/754

[8] Wu Y., Liu X., Feng Y., et al. Relation-Aware entity alignment for heterogeneous knowledge graphs // Proc. of the Twenty-Eighth International Joint Conference on Artificial Intelligence. – 2019. – P. 5278 –5284. DOI:10.24963/ijcai.2019/733

[9] Mao X., Wang W., Wu Y., et al. From alignment to assignment: frustratingly simple unsupervised entity alignment // Proc. of the 2021 Conference on Empirical Methods in Natural Language Processing. – 2021. – P. 2843 –2853. DOI:10.18653/v1/2021.emnlp-main.226

[10] Guo L., Sun Z., Hu W. Learning to exploit long-term relational dependencies in knowledge graphs // Proc. of the 36th International Conference on Machine Learning. – 2019. – Vol. 57. – P. 2505 –2514.

[11] Yang Z., Dai Z., Yang Y., et al. XLNet: Generalized autoregressive pretraining for language understanding // Proc. of the 33rd International Conference on Neural Information Processing

Systems. – 2019. – P. 5753 – 5763. DOI:10.5555/3454287.3454804

[12] Feng F., Yang Y., Cer D., et al. Language-agnostic BERT Sentence Embedding. ArXiv. – 2020. DOI:10.48550/arXiv.2007.01852