Russian-English dataset and comparative analysis of algorithms for cross-language embedding-based entity alignment

V A Gnezdilova1 and Z V Apanovich2
1Novosibirsk State University, 1, Pirogova str., Novosibirsk, 630090, Russia
2A.P. Ershov Institute of Informatics Systems, Siberian Branch of the Russian Academy of Sciences, 6, Acad. Lavrentjev pr., Novosibirsk 630090, Russia
E-mail: apanovich_09@mail.ru

Abstract. The problem of data fusion from data bases and knowledge graphs in different languages is becoming increasingly important. The main step of such a fusion is the identification of equivalent entities in different knowledge graphs and merging their descriptions. This problem is known as the identity resolution, or entity alignment problem. Recently, a large group of new entity alignment methods has emerged. They look for the so called “embeddings” of entities and establish the equivalence of entities by comparing their embeddings. This paper presents experiments with embedding-based entity alignment algorithms on a Russian-English dataset. The purpose of this work is to identify language-specific features of the entity alignment algorithms. Also, future directions of research are outlined.

1. Introduction
The knowledge graphs (KGs) are widely used as prior knowledge in applications such as recommender systems, decision-making systems, question-answering, etc. The more powerful an underlying knowledge graph is, the higher is the quality of applications based on it. Therefore, the problem of finding equivalent entities in several knowledge graphs and merging them into a unified knowledge graph is becoming increasingly important. This task has been known as entity alignment, entity deduplication, and identity resolution in the data base context. Various methods to establish the similarity of the symbolic features of entities have been studied extensively [1]. In the past few years, interest in the integration of multilingual knowledge graphs has increased, and the need in data fusion from multilingual knowledge graphs has made the problem of entity alignment (EA) vitally important.

Recently, embedding-based methods for entity alignment in various knowledge graphs have become widespread. These methods, known as knowledge embedding, encode entities and relations of various KGs into a low-dimensional vector space. The advantage of the embedding-based approach is high scalability and little effort in preparing training datasets.

Knowledge graphs represent knowledge in the form of relational and literal triples. A relational triple represents a relationship between two real-world objects and has the format \( tr_r = (subject\ entity, relationship, object\ entity) \). An example of a relational triple is \((dbr:War_and_Peace_film_series, dbo:director, dbr:Sergei_Bondarchuk)\). Sometimes the relational triples are represented as \( tr_r = (head\ entity, relation, tail\ entity) \). Literal triples store information about the attributes of entities in the form \( tr_l = (subject\ entity, attribute, literal\ value) \). An example of a literal triple is \((dbr:War_and_Peace_film_series, dbp:language, "Russian")\).
Two fragments of an English (left) and a Russian (right) knowledge graphs are shown in figure 1. The fragments describe two equivalent entities corresponding to the film “War and Peace” directed by a Soviet film director Sergei Bondarchuk. The equivalent entities are highlighted in blue.

![Diagram of knowledge graphs](image)

**Figure 1.** Fragments of an English (left) and a Russian (right) knowledge graphs. Entities, literals and relations are represented by ellipses, rectangles and directed edges respectively.

The intuition behind all the entity alignment algorithms is that equivalent entities should have equivalent relations and attributes in various knowledge graphs. However, real KGs have different origins and schemas, and entities’ relations and attributes can vary. For example, the English entity `dbr:War_and_Peace_film_series` in figure 1 has an attribute `dbo:distributor`, which the corresponding Russian entity `Война и мир` does not have. Also, the information about the film operator of the English entity is represented by an attribute triple while its Russian counterpart has a relational triple instead. Multiple inconsistencies between knowledge graphs require a variety of methods that should cope with these differences.

Since the creation of new methods is based mainly on the developer’s intuition, it is essential to establish a common basis for understanding and comparing these various methods. Currently, the common basis is formed by testing various algorithms on a unified dataset. For example, an OpenEA library [2] (https://github.com/nju-websoft/OpenEA) collects embedding-based entity alignment algorithms. These algorithms are analysed and compared on the datasets containing alignments between English-German, English-French and English-Chinese entities of dbpedia-2016-10.

However, each language version of a knowledge graph has its specificity, and language-specific features of the EA algorithms need to be studied and interpreted appropriately. This paper presents the results of experiments with several groups of entity alignment algorithms tested on a Russian-English dataset.

2. **The choice of embedding-based algorithms for entity alignment**

The embedding-based entity alignment algorithms differ in the type of triples used to obtain an embedding. All currently known approaches use relational triples to construct *structure embeddings*, and some of them use literal triples to generate *attribute embeddings*. Recent algorithms increasingly try to leverage the names of entities represented by the *rdfs:label* relationships to create *name embeddings*.

There are three main approaches to construct structure embeddings: translational, path-based, and neighbourhood-based embeddings.
**Triple-based, or translational embeddings** [3-5]. Translational models interpret a relation as a translation vector from a head entity to a tail one. The current methods represent different entities as embeddings and look for entity alignment by evaluating the similarity between these embeddings. This approach captures the local semantics of relational triplets. Each relational triple of the form \( hr^r t \) is associated with the three vector representations \( h, r, t \). The vector \( r \) representing the relation \( r \) is considered as a geometric transformation, for example, the shift of the vector \( t \) with respect to \( h \). The "energy" function of each relational triple is calculated as \( \| h + r - t \| \), and stochastic gradient descent is used to optimize it. The strong assumption about the translational nature of relationships makes this approach unsuitable for modelling more complex information about relationships. For example, these methods are not able to handle correctly all the transitive relationships, such as "ancestor" or "descendant".

**Path-based embeddings** [6]. This approach attempts to identify the relationships that can be represented as a combination of relationships corresponding to some directed path in the knowledge graph. For example, the fact that (\( p_1 \) has the son \( p_2 \)) and (\( p_2 \) has the son \( p_3 \)) suggests that (\( p_1 \) has the grandson \( p_3 \)). That is, if there is a sequence of interconnected relational triples of the form \((e_1, r_1, e_2), (e_2, r_2, e_3), \ldots, (e_{n-1}, r_{n-1}, e_n)\), the path-based approach tries to find an embedding of the relation \( r^* = comb(r_1, r_2, \ldots, r_n) \). The operation of addition, multiplication, etc., can be used as a combination method. The latest variations of this approach for the combination of embeddings of relations use various modifications of recurrent neural networks (RNN).

**Neighbourhood-based embeddings** [7, 8]. The embeddings of knowledge graph entities are constructed iteratively based on the embeddings of adjacent entities. These methods are based on the assumption that similar entities from different KGs have similar neighborhood structure. Therefore, equal embeddings should be generated for equivalent entities. However, due to the incompleteness and sparsity of real-life knowledge graphs, these neighborhoods can have multiple disparities of the neighborhood size and topological structure as it is shown in figure 1. This approach uses various versions of graph convolutional networks (GCN) to construct the embeddings of entities.

For experiments with the Russian-English dataset, the following algorithms have been used.

JAPE [3] combines structural and attribute embeddings to map entities from different knowledge graphs. The structural embedding is built on the basis of an "entity overlay graph" created from two knowledge graphs. The attribute embedding is built on top of a Skip-gram model that attempts to capture attribute correlations. JAPE uses information about the types of attributes but does not use their specific values.

BootEA [4] exploits a triple-based algorithm to obtain embeddings. This algorithm utilizes a special way of obtaining "negative" triples, the so-called "truncated homogeneous negative sample", which replaces the head or relation of a given positive triple with a random entity from \( s \) nearest neighbours (\( s \) is a hyperparameter). To use the existing interlanguage links for training, new triples are created: for each pair of entities, all triples that include one of the pairs of entities are duplicated with a replacement for an entity from another knowledge graph. An important property of BootEA is the iterative labelling of plausible alignments, which are then used as training data. Moreover, on subsequent iterations, entities can change their label or become unlabelled if the newly generated alignments lead to conflicts.

MultiKE [5] builds three types of embeddings for each entity, using different "views": a name view, a relational view, and an attribute view. Each of the "views" is built according to its own algorithm. For example, for each word from the entity name, there is a vector obtained using fastText [9]; if such a vector does not exist, then the word vector is obtained by summing the vectors of characters obtained using the character embedding algorithm. The word embeddings are summed up and a name view embedding is obtained, which is directly involved in training the model. A relation view embedding is constructed as standard translational structure embedding. An attribute view embedding is generated using a convolutional neural network (CNN). The final embedding of an entity can be obtained using different methods of combining the three views described above.

RSN4EA [6] exploits "relational paths", alternating the chains of entities and relationships, to build the embeddings of relationships and entities. The head and tail of the relational path must be entities. A
conventional way to model relational paths is recurrent neural networks (RNNs). However, regular recurrent neural networks do not distinguish relationships from entities in a relational path. Therefore, RSN4EA uses a "skip" recurrent network architecture (RSN). This architecture allows the relational path entities to participate in predicting not only the relationship, but the object entity as well.

GCN-Align [7] leverages graph convolutional networks that build the embeddings of entities (graph vertices) based on vertex adjacency information. The algorithm uses two two-layer GCNs, each of which constructs an embedding for one knowledge graph in a unified vector space. This approach assigns two feature vectors to each entity in the GCN layers, a structure feature vector and an attribute feature vector. The final outputs of two GCNs are further used to discover entity alignments. Correspondence between entities is established based on the distances between their structure and attribute embeddings.

RDGCN [8]. Graph convolutional networks are not very well suited for dealing with directed graphs. Therefore, the RDGCN approach uses not only the structure of the original knowledge graphs (primal graph) to construct embeddings, but also auxiliary graphs that are dual to the original graphs (dual relation graph). The vertices of dual relation graphs are the edges of the original graphs. To implement interaction between the original knowledge graphs and dual relational graphs, the Graph Attention Network (GAT)[10] mechanism is used. The resulting embeddings of the original graphs are then fed into graph convolutional networks to extract information about the structure of a vertex neighborhood.

3. A Russian-English data set for the EA experiments

A Russian-English dataset comprehensive for a Russian-speaking user has been created using the IDS algorithm [2]. This dataset contains 15 K and 100 K pre-aligned entities from the English and Russian versions of DBpedia 2016-10 (https://wiki.dbpedia.org/downloads-2016-10/). Similar to the already existing dataset DBP [X] K [2] containing the German-English, French-English and Chinese-English data sets, version 1 (V1) is obtained by directly using the IDS algorithm. To construct version 2 (V2), low degree objects (with the degree not exceeding five) in the source KG are randomly removed to double the average degree, and then IDS is performed to match the new KG.

4. Metrics for assessing the quality of embedding-based entity alignment algorithms

The Hits @ k metric is used to analyse the results of the EA algorithms. The metric Hits @ k = n% means that for n percent of objects from one knowledge graph, the equivalent object from the second knowledge graph is among the nearest k neighbours in the embedding space. Obviously, the Hits @ 1 metric is considered as the most indicative since this metric is equivalent to a precision. If Hits @ 1 = 100%, it means that such an algorithm can find the exact counterparts for all the entities. Table 1 and table 2 show the final metrics Hits @ k, where k = 1, 5, 10, 50, 100, and the running time of each of the selected algorithms on sparse and dense Russian-English dataset.

| RuEn15K_V1 | Hits@1 | Hits@5 | Hits@10 | Hits@50 | Hits@100 | Total time (s.) |
|-----------|--------|--------|---------|---------|----------|----------------|
| Jape      | 42.656 | 61.744 | 76.844  | 82.222  | 89.067   | 440.468 s.     |
| BootEA    | **48.156** | **68.922** | 75.911  | **87.167** | **90.589** | 2178.948       |
| MultiKE   | 37.344 | 49.233 | 55.222  | 70.844  | 76.689   | 1639.755       |
| GCN       | 41.944 | 64.678 | 71.933  | 83.844  | 87.3     | 76.023         |
| RDGCN     | 43.256 | 56.833 | 62.378  | 72.167  | 75.022   | 5020.332       |
| RSN4EA    | 45.589 | 61.822 | 67.5    | 80.378  | 84.633   | 6194.072       |

It is possible to see that BootEA has the best Hits@k metrics, however, on the sparse dataset, its results are quite close to the results of RSN4EA, GCN-Align and Jape.
Table 2. Assessment of the quality of the EA algorithms on the Russian-English dense dataset.

| RuEn15K_V2  | Hits@1 | Hits@5 | Hits@10 | Hits@50 | Hits@100 | Total time |
|-------------|--------|--------|----------|---------|----------|------------|
| Jape        | 51.322 | 75.578 | 82.444   | 91.256  | 93.489   | 367.196    |
| BootEA      | 66.486 | **88.156** | **93.4** | **98.156** | **98.822** | 2704.788   |
| MultiKE     | 47.333 | 59.911 | 65.078   | 76.6    | 81.8     | 3383.600   |
| GCN         | 57.256 | 83.844 | 89.333   | 95.356  | 96.611   | 95.480     |
| RDGCN       | 56.7   | 68.8   | 74.322   | 82.622  | 85.533   | 1214.713   |
| RSN4EA      | **69.383** | 82.789 | 86.033   | 91.8    | 92.12    | 13732.360  |

On the dense Russian-English data set, the superiority in BootEA and RSN4EA metrics over other solutions becomes obvious. However, these results differ from those obtained on the original DBP15K and DBP100K datasets [2], which demonstrate very good results for the MultiKE and RDGCN algorithms. The reason for this difference is that the literal embeddings of MultiKE exploit pretrained word embeddings for the respective languages. Another algorithm that has demonstrated essential quality degradation on the Russian-English dataset, compared to standard datasets, is RDGCN. Again, RDGCN leveraged a technique that translated the Chinese, German and French entity names into English and then utilised pretrained English word vectors to construct an input entity representation for the primal graph. Similar adjustments for Russian will be a subject of further investigation.

To gain more insight into the EA algorithms results, we used a program which takes as input the id of an English entity and outputs its ten nearest neighbours in the embedding space. The program has options allowing the users to choose a graph from which the nearest neighbours of a given entity are taken.

For example, the ten nearest neighbours of the entity id4 (http://dbpedia.org/resource/Hard_rock), produced by the GCN_Aligh algorithm, are shown in figure 2(a). The first line contains the object to which the elements nearest in vector space are searched, and the next lines contain its nearest ten neighbours. It is possible to see that all the nearest neighbours are related to music but their semantic closeness is difficult to estimate for a non-musician. Another EA result produced by the GCN_Algih algorithm is shown in figure 2(b). The nearest neighbours of the entity id12550 (http://dbpedia.org/resource/Vladimir_Korotkov,_born_1941) turned out to be football players and football clubs. Note that the Russian equivalent of the English entity is situated in the 4th line. It is not necessary to be a specialist in football to see that nobody except http://ru.dbpedia.org/resource/Коротков,_Владимир_Петрович can be a counterpart of this English entity. To understand this, it would be easier to compare the string similarity of the entities’ names. Thus, we can suppose that it is necessary to investigate hybrid methods exploiting embedding-based methods in combination with conventional methods. Moreover, these hybrid combinations can depend on particular language pairs and the types of entities.
Figure 2. English entities and their ten nearest neighbours produced by the GCN_Align algorithm.

5. Visualization of embeddings for the Russian-English dataset

To gain a global insight into the quality of the entity embeddings obtained, all the embeddings have been visualized using the t-SNE method (t-distributed Stochastic Neighbour Embedding [11]). The main advantage of the t-SNE is that it tends to preserve the distances between the objects of multidimensional space on a plane.

Since the Russian and English knowledge graphs contain the same objects, visualization allows one to estimate the EA quality visually. We can see the clusters of entities, whether the resulting clusters of objects coincide in size, and how the clusters in different language sets are situated with respect to one another. Therefore, the Russian entities were coloured red, and the English clusters were painted blue. A t-SNE visualization of the entity embeddings produced by the RSN4EA algorithm on the Russian-English 15K_V2 dataset is shown in figure 3. It can be seen that the entities and clusters in the two knowledge graphs "overlap" with each other, that indicates a globally good quality of results. At the same time, t-SNE visualization of the entity embeddings produced by the RDGCN algorithm on the Russian-English 15K_V2 dataset is shown in figure 4. As we can see, there are large pure blue and pure red clusters, that indicates a poor quality of the embedding produced on this Russian-English dataset.

This visualization complements and confirms the estimations provided by the Hits@k metric and by the local verification of nearest neighbours. To make the visualization more intuitive, we are going to combine the global visualization of two KGs with the possibility of interactive entity selection in one knowledge graph with its nearest neighbors in the second graph.
However, this partial superposition makes us think that the name embedding should be investigated more attentively with respect to various types of entities and various language pairs. The name embeddings should be especially useful for low-degree entities, where structure embeddings are useless.

6. Conclusion
Experiments with the EA algorithms applied to the Russian-English dataset have shown their superior quality with respect to the structure embeddings. However, the usefulness of attribute embeddings is not completely clear. The initial motivation behind using attribute embeddings was that they outnumbered relational embeddings three times. However, the real file with Russian DBpedia triples
mappingbased_literals_ru.ttl, extracted from Russian Wikipedia infoboxes, contained only 20% more attribute triples than relational ones. Moreover, its English analogue, mappingbased_literals_en.ttl, contained 20% less attribute triples than relational ones. However, the resulting Russian-English data set generated by the IDS algorithm contained about twice more attribute triples than relational ones. First, this “biased” triple distribution does not correspond to the distribution in the original knowledge graphs, and second, even this large amount of the attribute triples did not essentially improve the EA results. The best metrics are obtained by those algorithms that ignore the attribute embeddings.

On the other hand, the entity name embeddings seem essential to obtain a good EA. It is especially important because our experiments have shown that about 15 percent of Russian entities are represented by literals in the English DBpedia. We plan to investigate different strategies for the name embeddings with respect to various types of entities and various language pairs. Also, hybrid methods combining embedding-based methods with the conventional ones will be further investigated.

Acknowledgements
The authors are grateful to Zequn Sun for his assistance in the preparation of the Russian-English dataset

References

[1] Apanovich Z and Marchuk A 2015 Experiments on Russian-English identity resolution Proc. ICADL-2015 Conference Seul, South Korea, 9-13 December 2015 Springer International Publishing Switzerland 2015 R.B. Allen et al. (Eds.): ICADL 2015 LNCS 9469 pp 12–21.
[2] Sun Z, Zhang Q, Hu W, Wang C, Chen M, Akrami F, and Li C 2020 A Benchmarking Study of Embedding-based Entity Alignment for Know Proc. VLDB Endowment 13 pp 2326–40
[3] Sun Z, Hu W and Li C 2017 Cross-Lingual Entity Alignment via Joint Attribute-Preserving Embedding. d’Amato C. et al. (eds) The Semantic Web – ISWC 2017. ISWC 2017. LNCS 10587. Springer, Cham.
[4] Sun Z, Hu W, Zhang Q and Qu Y 2018 Bootstrapping Entity Alignment with Knowledge Graph Embedding / - c.7. Proc. 27th International Joint Conference on Artificial Intelligence (IJCAI-18) pp 4396–4402
[5] Zhang Q, Sun Z, Hu W, Chen M, Guo L and Qu Y 2019 Multi-view Knowledge Graph Embedding for Entity Alignment Proc. 28th International Joint Conference on Artificial Intelligence (IJCAI-19) pp 5424-35.
[6] Wu Y, Liu X, Feng Y, Wang Z, Yan R and Zhao D 2019 Relation-Aware Entity Alignment for Heterogeneous Knowledge Graphs Proc. IJCAI-2019 pp 5278-84.
[7] Wang Z, Lv Q, Lan X and Zhang Y 2018 Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing
[8] Guo L, Sun Z and Hu W 2019 Learning to Exploit Long-term Relational Dependencies in Knowledge Graphs, 36th International Conference on Machine Learning (ICML 2019)
[9] Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics 2017 5 pp 135–146.
[10] Veličković P, Cucurull G, Casanova A, Romero A, Liò P and Bengio Y 2018 Graph Attention Networks ICLR 2018 p 12
[11] Maaten van der L and Hinton G 2008 Visualizing data using t-SNE Journal of Machine Learning Research 9(2605) pp 2579-2605