A Joint Embedding Method for Entity Alignment of Knowledge Bases

Yanchao Hao(✉), Yuanzhe Zhang, Shizhu He, Kang Liu, and Jun Zhao

National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing 100190, China
{yanchao.hao,yuanzhe.zhang,shizhu.he,kang.liu,jun.zhao}@nlpr.ia.ac.cn

Abstract. We propose a model which jointly learns the embeddings of multiple knowledge bases (KBs) in a uniform vector space to align entities in KBs. Instead of using content similarity based methods, we think the structure information of KBs is also important for KB alignment. When facing the cross-linguistic or different encoding situation, what we can leverage are only the structure information of two KBs. We utilize seed entity alignments whose embeddings are ensured the same in the joint learning process. We perform experiments on two datasets including a subset of Freebase comprising 15 thousand selected entities, and a dataset we construct from real-world large scale KBs – Freebase and DBpedia. The results show that the proposed approach which only utilize the structure information of KBs also works well.

Keywords: Embeddings · Multiple knowledge bases · Structure information · Freebase · DBpedia

1 Introduction

As the amount of knowledge bases (KBs) accumulated rapidly on the web, the problem of how to reuse these KBs has gained more and more attention. In the real-world scenarios, many KBs describe the same entities in different ways, because KBs are distributional heterogeneous resources created by different individuals or organizations. For example, president *Barack Hussein Obama* is denoted by *m.02mjmr* in Freebase [3], while *Barack_Obama* in DBpedia [2]. Aligning such same entities could help people acquire knowledge more conveniently, as they no longer need to look up multiple KBs to obtain the full information of an entity. However, knowledge base alignment is not a trivial task, and the alignment system is often complex [8,15]. Many traditional KB matching pipeline systems including [7,11,20,22] are based on content similarity calculation and propagation.

There are some standard benchmark datasets from the Ontology Alignment Evaluation Initiative (OAEI), on which several alignment systems perform alignment algorithms. The datasets don't contain many relationships and two KBs to be aligned have common relation and property strings, which can be used

H. Chen et al. (Eds.): CCKS 2016, CCIS 650, pp. 3–14, 2016.
DOI: 10.1007/978-981-10-3168-7_1

to compute content similarity to assist instances alignment. The statistics of the author-disambiguation dataset from OAEI2015 Instance Matching are as Table 1. Think about a real case, we have an entity named *m.02mjmr* referring to president *Barack Hussein Obama*, How do we align it with the entity named *Barack_Obama* in another KB with all of the relations and properties in two different encoding system? When facing the cross-linguistic or different encoding situation, what we can leverage are only the structure information of two KBs. Content information is important to KB alignment, but we think the structure information of KBs is also significant. Based on the observation above, we create two datasets including a subset of Freebase comprising 15 thousand selected entities (FB15K) and a dataset we construct from real-world large scale KBs: Freebase and DBpedia. What we try to do is to construct datasets with abundant relations and rich structure information, regardless of the content.

Table 1. Statistics of author-dis sandbox from OAEI2015. The relations and properties are shared in two KBs.

Instance class	Author-instance	Relation	Property
2	854	6	6

In this paper, we perform the KB entity alignment task by leveraging the embeddings of the KBs which are learned via the structure of KBs no matter what the content is. In previous work, KB embeddings [4–6,9,17,21] are learned in order to complete the KB, and they aim at single KB. If the embedding learning method is applied on two KBs, we will obtain two independent embeddings in two different vector spaces. To represent two KBs in a uniform embedding vector space, we give some initial alignments, called seed entity alignments. In the learning process, we ensure the embeddings of the seed entities try to maintain the same. In this way, we could jointly learn the embeddings of the two KBs in a uniform embedding vector space, with two KBs connected by the seed entities "bridge". The seed alignments help learn potential alignments of the two KBs in the uniform expressive vector space via the network of the triplets. Entities with similar learned embeddings could be considered as the same entities. Thus we could find more alignments. The proposed method does not depend on manually designed rules and features, and we do not need to be aware of the content of the KBs. As a result, the proposed approach is more adaptive, could be easily utilized to large scale applications.

We conduct extensive large scale experiments on two datasets including a subset of Freebase comprising 15 thousand selected entities, denoted FB15K [5], and a dataset we construct from real-world large scale KBs – Freebase and DBpedia. The results indicate that the proposed method could achieve promising performance, and the joint embedding method only utilize the structure information of KBs, which may be a efficient supplement for KB alignment pipeline systems.

To the best of our knowledge, this is the first work to deal with the KB alignment problem using an end to end joint embedding model only utilizing the structure information of KBs. In summary, the contributions of this paper are as follows.

(1) We propose a novel model which jointly learns the embeddings of multiple KBs in a uniform vector space to align entities in KBs, only using the structure information of KBs.
(2) We construct two datasets for KB alignment task based on real-world large scale KBs: FB15K datasets and DBpedia-Freebase datasets, which have abundant relationships and rich structure information.
(3) We conduct experiments on the datasets, and the experimental results show that our approach works well.

The remainder of this paper is organized as follows. We first introduce our task in detail and overview of the related work. Then, we present the proposed method in the following section. Finally, we show the experimental results and conclude this paper.

2 Background

2.1 Task Description

Entity alignment on KBs, which is to align the entities that referring to the same real-world things, has been a hot research topic in recent years. For example, we should align the entity *m.02mjmr* in Freebase with the entity *Barack_Obama* in DBpedia. The goal of the KB alignment is to link multiple KBs effectively and create a large scale and unified KB from the top-level to enrich the KBs, which can be used to help machines understand the data and build more intelligent applications.

KBs usually use Resource Description Framework Schema (RDFS) or Ontology Web Language (OWL) or triples to describe ontology, defining elements such as“class”,“relation”,“property”,“instance” and so on. The research of KB alignment starts from ontology matching [23–25], mainly focusing on the semantic similarity at early time.

2.2 Related Work

Over the years, various methods have been proposed for KB alignment. Akbari et al. [1] and Suna et al. [19] utilize string-matching based methods which are quite straightforward but fail when two entity mentions are crossing languages or significantly different in literal. Joslyn et al. [10] consider the aligning problem as a graph homomorphism problem, [14,16] exploit Instance-based techniques to align KBs, and some take the KB alignment as combinatorial optimization problems [13].

In pairs-wise alignment methods, some supervised learning methods compare vectors via property to judge an entity pair whether should be aligned or not.

This kind of technology contains decision tree [26], Support Vector Machine (SVM) [27], ensemble learning [28] and so on. Some clustering based methods [29] learns how to cluster similar entities better.

In collective alignment methods, [18] present a PARIS system based on probabilistic method to align KBs without tuning parameters and training data, but PARIS cannot handle structural heterogeneity. Lacoste et al. [12] propose SiGMa algorithm to propagate similarity via viewing the task of KB alignment as a greedy optimation problem of global match score objective function.

All of them are based on content similarity calculation and propagation, and many ontology matching pipeline systems including [7,11,20,22] which participate in the OAEI 2015 Instance Matching track need to calculate content similarity. Some of them use local structure information to propagate similarity, but from another point of view, we think that the global structure information of KBs is also important. Our proposed models are based on global structure information of KBs, regardless of what the content exactly is.

3 Datasets

Because of the lack of suitable data for our task which is under the cross-linguistic or different encoding situation, we construct two datasets based on real-world large scale datasets. Firstly we present a dataset generated from FB15K, which is extracted from Freebase comprising 15 thousand selected entities. Then we illustrate the DBpedia-Freebase dataset (DB-FB), which are extracted from DBpedia and Freebase.

3.1 FB15K Dataset

FB15K. There are more than 2.4 billion triplets and 80 million entities in Freebase[1]. The base dataset we choose should not be too small to acquire enough overlapping part, and should not be too large to cause computational bottlenecks. As a tradeoff, we choose FB15K containing 592,213 triplets with 14,951 entities and 1,345 relationships. We randomly split them into two KBs, i.e., $kb1$ and $kb2$, with a large amount of overlapping part. Given a ratio number, i.e., the parameter *splitRatio*, we split the intersecting entities into two parts. The first part remains identical entity mention forms in two KBs, denoted as remaining part (seed alignment part). The second part keeps the entity mention forms unchanged in $kb1$, and changes the entity mention forms in $kb2$ by suffixing a certain string like "_#NEW#" to create the different entities, denoted by changing part (target alignment part), which is used for evaluation. Figure 1 indicates the splitting process of our datasets. There are two advantages of our proposed dataset. First, since they origin from the same FB15K dataset, we can control the overlapping part conveniently. Second, the gold entity alignment is known, so the evaluation is more accurate.

[1] https://developers.google.com/freebase/.

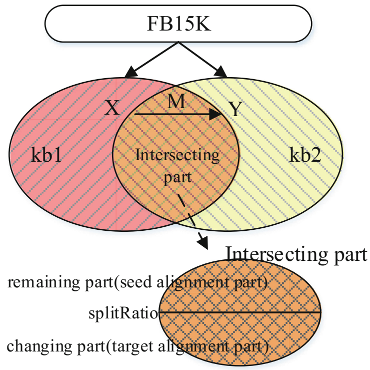

Fig. 1. The process of splitting FB15K.

3.2 DB-FB Dataset

DB-FB. There are more than 3 billion factual triples in DBpedia[2] and 2.4 billion in Freebase. DBpedia also provide datasets which contain triples linking DBpedia to many other datasets. Based on the given entity alignments with Freebase released on the DBpedia website[3], we can build a DBpedia-Freebase alignment dataset. Following the original intention, we intend to construct a dataset with abundant relationships and rich structure information. The dataset we construct should not be too small to contain enough structure information, and too large to cause computational bottlenecks. The steps of constructing DB-FB dataset are as follows.

Step1. As we know, Freebase triples have some Compound Value Types (CVTs) to represent data where each entry consists of multiple fields. Firstly, we need to convert the triples in Freebase which contain CVT to factual triples by reducing the CVT in the preprocessing step.

Step2. Then we find the triples in DBpedia and Freebase whose head and tail entity both show up in the given alignments.

Step3. In the selected triples, we count the frequencies of the entity alignment pairs (take the Napierian logarithm of the product of each entity's frequency in a pair) and rank the frequencies of the entity pairs.

Step4. Based on the top 10 thousand most frequently showing up entity alignment pairs, we select the triples whose head entity or tail entity are among the top 10 thousand entity alignment pairs in the picked out triples in step2.

Step5. Then we make a filter to reduce the triples whose entity frequency are less than 7 in DBpedia and 35 in Freebase.[4]

[2] http://wiki.dbpedia.org/Downloads2015-10.

[3] http://downloads.dbpedia.org/2015-10/links/freebase_links.nt.bz2.

[4] In step5, 7 and 35 are empirical values chosen in experiments.

The statistics of the DB-FB dataset are as Table 2.

Table 2. Statistics of DB-FB dataset.

	Triples	Entities	Relations	Align_pairs
DB	515,937	57,076	373	13,932
FB	724,894	19,166	1,219	

4 Methodology

Given two KBs, denoted by $kb1$ and $kb2$ respectively. The facts in both KBs are represented by triplets (h, r, t), where $h \in E$ (the set of entities) is the head entity, $t \in E$ is the tail entity, and $r \in R$ (the set of relationships) is the relationship. For example, *(Obama, president_of, USA)* is a fact. Different from previous KB embedding learning methods, our model learns the joint embeddings of the entities and the relations of two KBs. In detail, we firstly generate several entity alignments using simple strategies which leverage some extra information or other measures. As shown in Fig. 2, the entities in the same color are the entity alignments, i.e., the selected seed entities. In this way, the seed entity alignments could serve as bridges between $kb1$ and $kb2$, thus we can learn the joint embeddings of both KBs in a uniform framework.

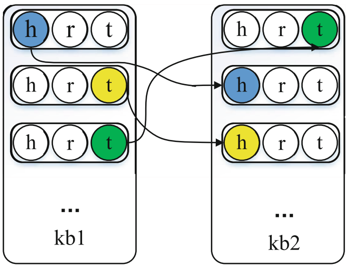

Fig. 2. Selecting seed entities in two KBs.

A KB is embedded into a low-dimensional continuous vector space while certain properties of it are preserved. Generally, each entity is represented as a point in that space while each relation is interpreted as an operation over entity embeddings. For instance, TransE [5] interprets a relation as a translation from the head entity to the tail entity. Following the energy-based framework in TransE, the energy of a triplet is equal to $d(h+r, t)$ for some dissimilarity measure d, which we take to be either the L_1 or L_2-norm. To learn such embeddings, we minimize the margin-based objective function over the training set:

$$L = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'_{(h,r,t)}} \{[\gamma + d(h+r,t) - d(h'+r,t')]_+ + \lambda_1 \sum_{y\in\{h,h',r,t,t'\}} ||\,||y||_2 - 1|\} + \lambda_2 \sum_{(e_i,e'_i)\in A} ||e_i - e'_i||_2 \tag{1}$$

where $[x]_+$ denotes the positive part of x, $\gamma > 0$ is a margin hyper-parameter, λ_1, λ_2 are ratio hyper-parameters, A is the selected seed alignments whose entities are represented by e_i in $kb1$ and e'_i in $kb2$, and

$$S'_{(h,r,t)} = \{(h',r,t)|h' \in E\} \cup \{(h,r,t')|t' \in E\} \tag{2}$$

The set of corrupted triplets, constructed according to Eq. (2), is composed of training triplets with either the head or tail replaced by a random entity (but not both at the same time). The objective function is optimized by stochastic gradient descent (SGD) with mini-batch strategy. The soft constraints of the entities and relations (the λ_1 part in Eq. (1)) is important because they are meaningful in preventing the training process to trivially minimize the loss function by increasing the embedding norms and shaping the embeddings [5]. The alignment part (the λ_2 part in Eq. (1)) helps learn the alignment information between KBs.

Following the projection transformation idea, we can fix Eq. (1) by adding a projection transformation matrix M_d:

$$L = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'_{(h,r,t)}} \{[\gamma + d(h+r,t) - d(h'+r,t')]_+ + \lambda_1 \sum_{y\in\{h,h',r,t,t'\}} ||\,||y||_2 - 1|\} + \lambda_2 \sum_{(e_i,e'_i)\in A} ||M_d e_i - e'_i||_2 \tag{3}$$

The projection matrix M_d serves as the transformation of different KB vector spaces. It is more reasonable to transfer one KB vector space to another when we want to connect two KBs.

In the learning process, the embeddings of the entities in $kb1$ could become more and more similar with the same factual world entities in $kb2$ through seed entities. So the jointly learned embeddings can help improve entity alignment between the two KBs. The key of our model is to align two KBs using embeddings in a uniform space that jointly learned via the overlapping parts between the two KBs.

5 Experimental Evaluations

5.1 Baseline

Given the two KBs generated from FB15K, we suffix all the intersecting entities in $kb2$ to make $kb2$ totally different from $kb1$. Then we learn the embeddings of the entities and relations in the two KBs in two vector space individually following TransE [5]. Since the intersecting entities are split into two parts, we use the remaining part to learn the projection transformation matrix M, representing transformation of the same entities from one vector space to the other using the following equations:

$$Y^T = MX^T \tag{4}$$

$$M = Y^T X(X^T X)^{-1} \tag{5}$$

Where X denotes the embedding matrix of the remaining part of $kb1$, Y denotes the embedding matrix of the remaining part of $kb2$, and M denotes the projection transformation matrix. Let len denote the number of entities in the remaining part, and dim denotes the dimension of the embeddings. So the matrixes X and Y are $\mathbb{R}^{len \times dim}$, while the matrix M is $\mathbb{R}^{dim \times dim}$.

As for the changing part, we could obtain the projection embeddings of the entities of $kb1$ Y in the vector space of $kb2$, using Eq. (4). In other words, the function of matrix M is to transform the embeddings in $kb1$'s vector space to $kb2$'s vector space in order to find the degree of similarity between the projected embeddings and the true embeddings.

In DB-FB dataset, we can directly use the Eqs. (4) and (5) without changing the forms of the entities.

5.2 Implementation

For our model, we regard the remaining part as the seed alignment part. Some hyper-parameters in two models were just set empirically. For experiments settings, when we learn the embeddings, we choose the margin γ as 1, the dimension k as 100, the λ_1 in loss function as 0.1, the λ_2 in loss function as 1, the epoch for training as 2000. The dissimilarity measure d is L_2 distance. The embeddings of entities and relations are initialized in the range of $[-0.01, 0.01]$ with uniform distribution. Table 3 shows the comparison of overall results where there are 7,365 entities in the target entity part for evaluation and 14,825 entities in $kb2$ totally under the parameters setting $splitRatio = 0.5$. Every entity in the target entity part could have rank value from 1 to 14,285. In this table, Mean_Rank represents the mean rank value of the target entities part, and Hits@n means the ratio number of entities that rank at top n.

Table 3. Overall results of FB15K. JE denotes our joint Embedding model in Eq. (1), and $JEwP$ denotes as our joint Embedding model with projection matrix in Eq. (3).

Models	Mean_Rank	Hits@1	Hits@10	Hits@100
Baseline	95.97	23.96%	54.96%	83.22%
JE	94.76	29.73%	56.36%	81.91%
JEwP	**88.51**	**29.88%**	**59.21%**	**84.97%**

Our model improves the performance significantly compared with the baseline approach. We believe that the good performance of our model is due to jointly embedding two KBs into a uniform vector space via seed entities "bridge" connecting two KBs. The seed alignments help learn potential alignments of the two KBs in the uniform expressive vector space via the triplets' network, while in the baseline model, we can only utilize the projection transformation matrix learned from the seed alignment part with no extended alignment information on the whole.

Table 4. Effect of *splitRatio* on FB15K.

Models	splitRatio	Mean_Rank	Hits@1	Hits@10	Hits@100
Baseline	0.1	91.79	25.10%	56.52%	83.84%
	0.3	92.71	23.34%	54.25%	82.95%
	0.5	95.97	23.96%	54.96%	83.22%
	0.7	94.44	25.12%	55.66%	83.10%
JE	0.1	352.00	10.25%	20.19%	47.18%
	0.3	239.56	15.47%	31.63%	63.30%
	0.5	94.76	29.11%	56.62%	81.91%
	0.7	97.85	29.73%	56.36%	81.48%
JEwP	0.1	205.74	17.59%	42.34%	66.67%
	0.3	123.28	25.63%	55.35%	78.60%
	0.5	88.51	29.88%	59.21%	84.97%
	0.7	**86.83**	**30.38%**	**60.70%**	**85.14%**

We also explore the effect of *splitRatio*, i.e., the number of seed entities, on our models. As shown in Table 4, along with the ascending order of *splitRatio*, the Mean_Rank value of our model decreases and the Hits@n increases, indicating the performance of our model getting better because of more seed entities. While the baseline model shows much more placid when the *splitRatio* increases, as shown in Fig. 3. The impression of the baseline model is that the performance should be increasing along with the ascending order of *splitRatio* because there are more and more data to learn the projection transformation matrix M well. But the result is almost placid. The reason in further analysis shows that when $splitRatio = 0.1$ the categories of the entities in the remaining part to learn are already covered enough and the projection transformation based method cannot depict the influence of different relations to the entity alignment. While our joint embedding method learns the different representations of different relations which help improve the performance of alignment. For example, the relation "son_of" is more important than the relation "nationality" in judging whether two entities are the same or not.

We conduct experiments on the DB-FB dataset, and the results are as Table 5. The baseline model has better $Mean\_Rank$, and our joint embedding

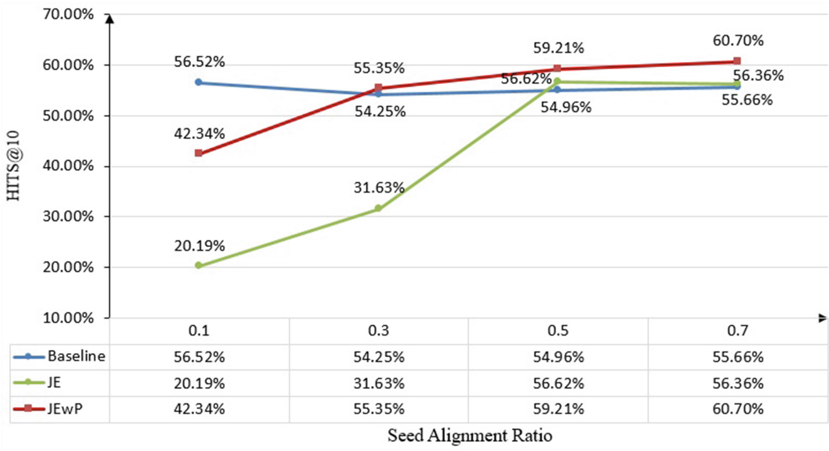

Fig. 3. The performance of our models on FB15K along with the ascending *splitRatio*.

Table 5. Results on the DB-FB dataset.

Models	SeedAlignments_Ratio	Mean_Rank	Hits@1	Hits@10	Hits@100
Baseline	0.1	554.43	2.20%	14.56%	45.81%
	0.3	485.46	2.00%	14.85%	46.46%
	0.5	490.23	2.18%	14.76%	48.11%
	0.7	**476.25**	2.20%	14.81%	49.13%
JE	0.1	1019.98	0.66%	4.46%	27.06%
	0.3	785.63	1.25%	8.55%	35.67%
	0.5	723.18	1.57%	11.35%	38.68%
	0.7	700.00	1.87%	13.15%	41.40%
JEwP	0.1	639.89	1.91%	9.86%	41.72%
	0.3	605.39	2.38%	15.18%	45.65%
	0.5	524.74	3.91%	19.39%	53.34%
	0.7	510.18	**4.64%**	**19.90%**	**54.89%**

projection model has better performance at *Hits@n* when we have a certain number of seed Alignments. The reason may be that the baseline model learns the projection transformation matrix from a global perspective, while our models learn the embeddings of KBs and projection matrix M_d (especially the JEwP model) in the iterative optimization process. The DB-FB dataset is relatively large and the selected DBpedia set which has 515,937 triples and 57,076 entities is more sparse than the selected Freebase set which has 724,894 triples and 19,166 entities. So on the DB-FB dataset, it may be more difficult to capture the global accurate alignment information for our models in the learning process. Note that our models only utilize the structure information of KBs to align entities, not the accurate content information. When we are faced with actual KB alignment task, our model may be an efficient supplement to the alignment pipeline systems.

6 Conclusions

We propose a model which jointly learns the embeddings of KBs in a uniform vector space via seed entity alignments to align KBs. Generally, our model with projection matrix has better performance than our model without projection matrix, which is reasonable for that projection matrix indicates transformation of KBs, and projection matrix should be added when we associate one vector space with another. To utilize structure information of KBs, we construct two datasets including FB15K and DB-FB based on real-world large scale KB. The experimental results show that the proposed approach which only utilize the structure information of KBs also works well, and may be an efficient supplement for KB alignment pipeline systems.

Acknowledgement. This work was supported by the Natural Science Foundation of China (No. 61533018), the National Basic Research Program of China (No. 2014CB340503) and the National Natural Science Foundation of China (No. 61272332). And this work was also supported by Google through focused research awards program.

References

1. Akbari, I., Fathian, M., Badie, K.: An improved mlma+ and its application in ontology matching. In: Innovative Technologies in Intelligent Systems, Industrial Applications, CITISIA 2009, pp. 56–60. IEEE (2009)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). doi:10.1007/978-3-540-76298-0_52
3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 1247–1250. ACM (2008)
4. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: Conference on Artificial Intelligence, number EPFL-CONF-192344 (2011)
5. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, pp. 2787–2795 (2013)
6. Chang, K.-W., Yih, W.-T., Meek, C.: Multi-relational latent semantic analysis. In: EMNLP, pp. 1602–1612 (2013)
7. Damak, S., Souid, H., Kachroudi, M., Zghal, S.: EXONA Results for OAEI (2015)
8. Gokhale, C., Das, S., Doan, A., Naughton, J.F., Rampalli, N., Shavlik, J., Zhu, X.: Hands-off crowdsourcing for entity matching. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 601–612. ACM (2014)

9. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of ACL, pp. 687–696 (2015)
10. Joslyn, C.A., Paulson, P., White, A., Al Saffar, S.: Measuring the structural preservation of semantic hierarchy alignments. In: Proceedings of the 4th International Workshop on Ontology Matching, CEUR Workshop Proceedings, vol. 551, pp. 61–72. Citeseer (2009)
11. Khiat, A., Benaissa, M.: InsMT+ Results for OAEI 2015 Instance Matching (2015)
12. Lacoste-Julien, S., Palla, K., Davies, A., Kasneci, G., Graepel, T., Ghahramani, Z.: Simple greedy matching for aligning large knowledge bases. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 572–580. ACM (2013)
13. Prytkova, N., Weikum, G., Spaniol, M.: Aligning multi-cultural knowledge taxonomies by combinatorial optimization. In: Proceedings of the 24th International Conference on World Wide Web Companion, pp. 93–94. International World Wide Web Conferences Steering Committee (2015)
14. Pushpakumar, R., Sai Baba, M., Madurai Meenachi, N., Balasubramanian, P.: Instance Based Matching System for Nuclear Ontologies (2016)
15. Scharffe, F., Zamazal, O., Fensel, D.: Ontology alignment design patterns. Knowl. Inf. Syst. **40**(1), 1–28 (2014)
16. Seddiqui, M., Nath, R.P.D., Aono, M.: An efficient metric of automatic weight generation for properties in instance matching technique. arXiv preprint arXiv:1502.03556 (2015)
17. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems, pp. 926–934 (2013)
18. Suchanek, F.M., Abiteboul, S., Senellart, P.: Paris: probabilistic alignment of relations, instances, and schema. Proc. VLDB Endow. **5**(3), 157–168 (2011)
19. Suna, Y., Maa, L., Wangb, S.: A Comparative Evaluation of String Similarity Metrics for Ontology Alignment (2015)
20. Wang, W., Wang, P.: Lily Results for OAEI 2015 (2015)
21. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI, pp. 1112–1119. Citeseer (2014)
22. Zhang, Y., Li, J.: RiMOM Results for OAEI 2015 (2015)
23. Shvaiko, P., Euzenat, J.: Ten challenges for ontology matching. In: Meersman, R., Tari, Z. (eds.) OTM 2008. LNCS, vol. 5332, pp. 1164–1182. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88873-4_18
24. Berstein, P.A., Madhavan, J., Rahm, E.: Generic schema matching, ten years later. Proc. VLDB Endow. **4**(11), 695–701 (2011)
25. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Trans. Know. Data Eng. **25**(1), 158–176 (2013)
26. Han, J.W., Kambe, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco (2006)
27. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Berlin (2000)
28. Kantardzic, M.: Data Mining. Wiley, Hoboken (2011)
29. Cohen, W.W., Richman, J.: Learning to match, cluster large high-dimensional data sets for data integration. In: Proceddings of Advances in Neural Information Processing Systems, pp. 905–912. MIT Press, Cambridge (2005)