Relation Linking for Wikidata Using Bag of Distribution Representation

Xi Yang, Shiya Ren, Yuan Li, Ke Shen, Zhixing Li(✉), and Guoyin Wang

Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, People's Republic of China
lizx@cqupt.edu.cn

Abstract. Knowledge graphs (KGs) are essential repositories of structured and semi-structured knowledge which benefit various NLP applications. To utilize the knowledge in KGs to help machines to better understand plain texts, one needs to bridge the gap between knowledge and texts. In this paper, a Relation Linking System for Wikidata (RLSW) is proposed to link the relations in KGs to plain texts. The proposed system uses the knowledge in Wikidata as seeds and clusters relation mentions in text with a novel phrase similarity algorithm. To enhance the system's ability of handling unseen expressions and make use of the location information of words to reduce false positive rate, a *bag of distribution* pattern modeling method is proposed. Experimental results show that the proposed approach improves traditional methods, including word based pattern and syntax feature enriched system such as OLLIE.

Keywords: Relation linking · Knowledge graph · NLP

1 Introduction

Knowledge graph (KG) is able to provide structured and connected information between entities. Before utilizing knowledge graph in natural language understanding applications, one of the main challenge is mapping the knowledge in KGs to plain texts. Entity linking is one attempt towards this challenge. It links the surface names in the texts to corresponding entity objects in KGs. While most research focus on entity linking [2,3], few attention has been paid on mapping relations in KGs to texts which we called *Relation Linking*.

Table 1 shows the number of possible patterns of 9 different relations collected from Wikipedia. The second column is the number of sentences that contains corresponding mentions, namely positive samples. The third column is the number of distinct accurate patterns in these sentences and the last column is the number of distinct patterns extracted by OLLIE [8]. One may use virous relation expressions extracted OLLIE directly to link relations to texts. However, it may suffer from two drawbacks. Firstly, these patterns cannot cover all the possible expressions of the relations so it will inevitably encounter low recall problem when dealing with new texts. Secondly, it is difficult to find if an accurate pattern

X. Huang et al. (Eds.): NLPCC 2017, LNAI 10619, pp. 652–661, 2018.
https://doi.org/10.1007/978-3-319-73618-1_55

Table 1. Statistic of patterns and sentences of 9 relations in Wikidata

Relations	# of sentences	# of Acc. patterns	# of OLLIE patterns
Father	6,630	92	205
Mother	3,324	31	81
Child	4,396	39	185
Spouse	7,586	106	146
Language	3,958	13	85
Country	10,116	76	407
Religion	2,239	30	58
Edu	5,033	81	130
Politic	5,530	85	133

is really a mention of the given relation since there exists a lot of noise in the automatically labelled data.

To address these problems, this paper proposes a Relation Linking System for Wikidata (RLSW) which learns flexible relation patterns from Wikidata and Wikipedia. For each relation defined in Wikidata, one or more patterns that can cover various natural language phrases in Wikipedia articles are learnt. As a result, each pattern can cover a cluster of words sequence. It has to be noticed that although RLSW is designed for Wikidata, the framework proposed in this paper can be immigrated to other KGs easily.

Bridging the gap between KGs and plain texts is one vital step in NLP. The RLSW proposed in this paper focuses on mapping relations to plain texts and its contributions are as follows:

- A framework for relation linking is proposed while much research focus on entity linking. In this framework, word sequences are clustered and then are connected to relations in KGs.
- A new phrase similarity scoring method is proposed for pattern clustering. It combines semantic distance and spacial distance of words and produces better results than *bag of words* settings.
- A new phrase pattern algorithm is proposed. In the proposed algorithm, a pattern is a set of distributions of words. We also give the algorithm to obtain these patterns from phrases and match new phrases with these patterns.

The rest of this paper is organized as follows. Section 2 reviews related work about the topic. The general framework and key part of RLSW are described in Sects. 3 and 4. Section 5 gives results to the examine the proposed RLSW. Finally, in Sect. 6, conclusions are summarized.

2 Related Work

RLSW contains several necessary procedures and each procedure relates to a specific research topic. In this section, we will briefly survey existing related work such as relation extraction, sentence similarity and entity linking.

2.1 Relation Extraction

As a typical subtask of information extraction, relation extraction identifies the relation between entities which is closely related to relation linking. NELL [8] learned many types of knowledge from self-supervised experience with self-reflection. The system classified pairs of phrases by learning boolean-valued relation. The relation is limited by the type that mostly are *is-a* or *class-of*. Wu and Weld [11] presented WOE^{parse} which used features generated from dependency parsing trees to extract *subj-rel-obj* tuples. However, this method relies on the output of dependency parsing and any errors of dependency parsing may cause error propagation. OLLIE overcame WOE's limitations and extracted relations by including contextual information from the sentences. However, OLLIE fails to consider the connection between unstructured text and structured information.

2.2 Sentence Similarity

RLSW adopts sentence similarity to cluster relation mentions. Much research in recent years has been reported in this task. Li et al. [7] utilized information from a structured lexical database and corpus statistics to calculate the semantic similarity of two sentences. He et al. [4] proposed a multiplicity of perspectives by using a convolutional neural network to extract features and then used multiple similarity metrics to compare sentences at different granulates. When faced large amounts of training data, it cannot perform well. The WMD, presented by Kusner et al. [6], learned semantically meaningful representations for words. However, these approaches only consider the semantic similarity between words, while in natural language, the position of words also matters. In this paper, a location-sensitive WMD algorithm is proposed to capture the words location information for the calculation of sentence similarities.

2.3 Entity Linking

Entity linking is another task of mapping knowledge in KGs to texts which determines the identity of entities mentioned in text in a structured KG. There have been a great number of studies in entity linking. Shen et al. [10] used two candidate entity ranking methods such as supervised ranking and unsupervised ranking to get top entity for mapping to KB. Another application is the ZenCrowd [2] which implemented large-scale entity linking using probabilistic reasoning and crowd-sourcing techniques. Han et al. [3] proposed a graph-based collective EL method, which can model and exploit the global interdependence between different EL decisions. These studies on entity linking have not taken the relationship between entities into account to provide the whole picture of information in knowledge graph mapping. As a supplement, RLSW is proposed to build connections between relations in KG and their mentions in texts.

3 Problem Definition

Formally, the relation linking is a mapping from natural language mentions to relations in KGs which can be defined as follows:

$$f : \mathscr{M} \rightarrow \mathscr{R}$$

where $\mathscr{M}$ is the set of possible mentions of relations and $\mathscr{R}$ is the set of relations in KG. In practice, mention $m \in \mathscr{M}$ is a sequence of words.

The task of relation linking is to find the mentions of a relation. E.g., mentions such as *is born in*, *is the hometown of*, *come from* can be linked to Wikidata relation *birthplace*. Since it is impossible to enumerate all possible mentions, one alternative is approximating f with relation classification algorithm $\hat{f}$ which takes word sequence as input and decides if it contains a relation mention.

4 Relation Linking Using Bag of Distribution Representation

The goal of relation linking is to link the mention of relations to KGs. Figure 1 shows the architecture of the proposed framework. First, for a given relation (in Wikidata, *property* is used instead of *relation*), such as *father* (P22), SPARQL API is used to query $\langle subj., obj. \rangle$ pairs from Wikidata and then sentences contains both subject and object from Wikipedia articles are selected as the source sentences of relation mentions. Second, the obtained mentions are clustered using a new similarity scoring method proposed in this paper. Third, each mention cluster is represented as a Bag of Distribution (BoD) and last, a classifier is trained based on the mention vectors which consists of the fitness between the mention and the BoDs. The main contributions of this work focus on the clustering part and the BoD pattern learning part.

4.1 Preprocessor

The main goal of preprocessor is to obtain sufficient data for the learning and training. It consists of two steps. First, obtaining $\langle subj., obj. \rangle$ pairs from Wikidata. Second, finding relation mentions according to $\langle subj., obj. \rangle$ pairs.

As for the first step, Wikidata SPARQL API [5] is used. We first get a list of entities of a selected category (e.g. *HUMAN*), then query the property value (e.g. *father*) of these entities to build $\langle subj., obj. \rangle$ pairs list. At the second step, $\langle subj., obj. \rangle$ pairs are used to find relation mentions from corresponding Wikipedia Articles. Each sentence that matches a $\langle subj., obj. \rangle$ pair be selected and the word sequence between the label of subject and the label of object is annotated as relation mentions.

To get rid of noises, only mentions that contains 4–10 words are reserved for further processing.

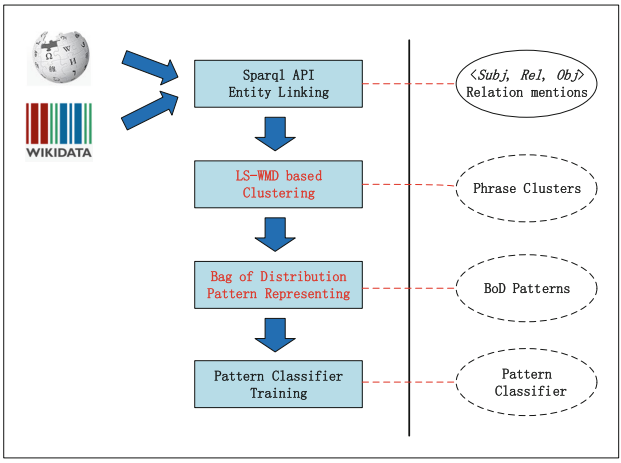

Fig. 1. Architecture of RLSW

4.2 Mention Clustering

Mention Clustering aims at grouping mentions to mining patterns. There are various natural language representations for a given relation and it's better to mine patterns from groups of similar mentions instead from all mentions.

Since mentions are word sequence but not numeric vectors, one needs to give an effective measure for word sequence pairs before clustering. Word Mover's Distance (WMD) [6] leverages word embedding to measure the semantic distances between words and then uses Earth Mover Distance (EMD) to measure the distance between word sequences. However, due to the natural of EMD, WMD does not take into consideration the location and order of words. In fact, apart from semantic similarity between words, syntactic information plays a vital role in sentence understanding. For instance, the meaning of two sentences *John's father is Dan* and *John is the father of Dan* is very close according to WMD and therefor the system would not tell if this is a mention of relation *son* or *father*. To address this problem, the position information of words is imported in this work when calculating the similarity between word sequences.

Location Sensitive Word Mover's Distance: In WMD, words are represented as embedding vectors and the calculation of distance between words is converted to the calculation of distance between vectors. Since location of words also matters, one needs to add location information to the calculation of distances between words. In this work, the location of a word in a sequence is encoded a value between 0 and 1. Given a mention $\{w_1, w_2, ...w_n\}$, the location value of w_i is calculated as:

$$loc(w_i) = \frac{1}{n} * (i - 0.5)$$

E.g., given mention *is the son of*, the location value of *son* $loc(w_3) = \frac{1}{4}(3-0.5) = 0.625$. The bias 0.5 is used to ensure that the location value lies in the center of

word's range in the sentence. As a result, the location differences between words can be calculated as the absolute difference between their location values. These location differences are imported to WMD as follows:

$$\begin{aligned} &min_{\mathbf{T}\geq\mathbf{0}} \sum_{i,j=1}^{n} \mathbf{T}_{ij} D_{sem+loc}(s_i, s'_j) \\ &suject\ to: \sum_{j=1}^{n} \mathbf{T}_{ij} = d_i\ \forall i \in \{1, ..., n\} \\ &\sum_{i=1}^{n} \mathbf{T}_{ij} = d'_j\ \forall j \in \{1, ..., n\} \end{aligned} \tag{1}$$

where s_i is the i_{th} word of s, d_i is the weight of i_{th} word of s, T is the flow matrix which defines the mass from words in s to words in s' and $D_{sem+loc}(s_i, s'_j)$ is the similarity distance between s_i and s'_j. It is composed of the semantic distance and syntactic distance, namely,

$$D_{sem+loc}(s_i, s'_j) = \alpha D_{sem}(s_i, s'_j) + (1 - \alpha) D_{loc}(s_i, s'_j), \alpha \in [0, 1] \tag{2}$$

α is hyperparameter and its value is detailed in Sect. 5.2. Semantic distance is calculated as in WMD, and location distance is calculated as stated in this paper. Since this measurement takes into consideration the location of words, it is named as Location Sensitive Word Mover's Distance (LSWMD).

Density Peaks Based Clustering (DPC): In this paper, Density Peak Based Clustering [9] is chosen as the clustering algorithm for relation mentions because of its efficiency and ability of handling distance matrix. Currently, clustering center are manually selected and the number of clusters is manually set which is controlled within ten.

4.3 Bag of Distribution Pattern Representing

After detailed data observation, we find that for a given relation, the locations of words in mentions are relatively stable. E.g., although relation *father* has many different mentions such as *was son of*, *was daughter of*, *is first son of*, the word *was/is* always locates at the front of the mention and *son/daughter* always locates at the second half. If one uses a distribution to describe a word in the mentions, it may provide much more information than frequency based methods.

Since the location of words are encoded as values between 0 to 1, beta distribution is selected to model the location distribution of words. As a result, a mention cluster that contains a list of mentions can be converted to a list of weighted beta distributions of words. The probability density of the beta distribution indicates how likely the word appears in some position and the weight indicates that how frequently the word appears in the mention cluster. A Bag of Distribution Representing of a mention cluster is defined as follows:

$$BoD(c) = \{(p_i, \alpha_i, \beta_i) | w_i \in W_c\} \tag{3}$$

where c is a cluster of mentions, W_c is the vocabulary of c, $p_i = \frac{count(w_i)}{\sum_i count(w_i)}$ is the probability of w_i in W_c. α_i and β_i is the parameter of modeled beta distribution of word w_i.

Mention Scoring: Given a mention $m = \{w_1, w_2, ...w_n\}$, its fitness towards a given BoD pattern c is calculated as follows:

$$fit_{BoD(c)}(m) = \sum_{w_i}^{w_i \in m} p_{w_i} * \int_{range(w_i)} Beta(\alpha_{w_i}, \beta_{w_i}) \tag{4}$$

where $range(w_i)$ is the normalized location range of word w_i in m. E.g., $r(son)$ in $is\ son\ of$ is $[0.333, 0.667]$. An alterative method is using a vector to describe the fitness between m and $BoD(c)$ where the components of this vector is the fitness of single word in m w.r.t $BoD(c)$.

4.4 Relation Classification

For each mention, a vector which consists of its finesses towards all BoD patters is build. We use a window which is generated by sentence frequency to san the testing sentences and find the most matched word strings as the mention of a relation. The rest sentences are selected as negative samples. The last component of RLSW is a relation classifier trained on such vectors. GBDT is used as the classifier in our experiment. For testing sentences to be linked, we first find the candidate word sequences by window from them and then convert each word sequence to a vector. The trained classifier takes the vector as input to decide if it should be linked to the given relation.

5 Experiments and Evaluations

5.1 Data Sets

The relations used in experiments are manually collected from Wikidata. For convenience, all relations are associated with category *HUMAN*. In total, 20,000 entities are queried from Wikidata and due to the data sparse problem, not all entities have all property values of these selected relations. The Wikipedia page of these 20,000 is crawled as the source plain text and the matching method described in Sect. 4.1 is used to find mentions. At last, about 3,000–10,000 sentences are matched for these relations. The details can be found in Table 1.

5.2 Experiments Setting

Since RLSW is a framework that can collaborate with different kinds of instance based matching method, four series of experiments are conducted, i.e., $Accurate\ Pattern(Acc.)$, $OLLIEPattern(OLLIE)$, $BoD + AccuratePattern(BoD(Acc))$ and $BoD + OLLIEPattern(BoD(OLLIE))$. *Acc.* uses the word sequences in training samples directly to match mentions from testing sentences. $OLLIE$ parses all training samples and selects the predicates in tuples contained correct $\langle subj., obj. \rangle$ pairs as templates to match mentions. $BoD(Acc.)$ learns BoD patterns from word sequences used in *Acc.* and $BoD(OLLIE)$ learns BoD patterns

from predicates extracted by *OLLIE*. For all relations, 10-fold cross-validation is used to evaluate all tested methods. *Acc.* and *OLLIE* do not need to be trained and false samples are used to calculate the confidence of patterns.

We find that the selection of cut-off distance d_c in DPC has little effect on the results. So we set the around 1% quantile as the DPC parameter.

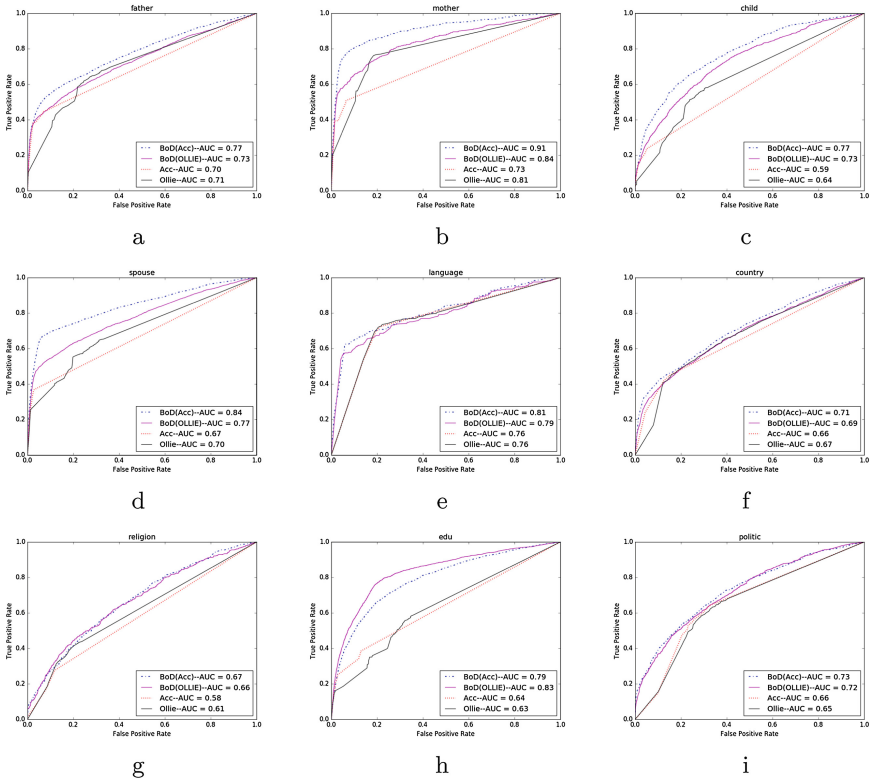

Fig. 2. Experimental results of 4 tested methods

5.3 Performance Analysis

This subsection reports and discusses the experimental results of all four tested methods. The performance is compared using AUC value.

Overall Performance Comparison

From Fig. 2 we can see that:

– BoD Pattern brings improvements to both *Acc.* and *OLLIE* on all relations, especially for the relation *mother*, *spouse* and *edu*. It reserves low FPR at the early stage and produces higher TPR than the baseline as the growth of FPR. It means that BoD Pattern recalls more true positive samples with the

same amount of false positive samples. E.g., for relation *mother*, the TPR of $BoD(Acc)$ is 0.8 at a FPR 0.1 while the TRP of Acc and $OLLIE$ are 0.6.
– $OLLIE$ outperforms $Acc.$ in 6 of 9 relations. One possible reason is that accurate pattern contains too much noise that leads a high FPR.
– BoD pattern brings more significance improvements to $Acc.$ Although $OLLIE$ outperforms $Acc.$, after the denoising of BoD, it archives a higher AUC than $OLLIE$ in most relations. The dependency or syntactical information prevent OLLIE find word sequence that are not syntactically close to each other.

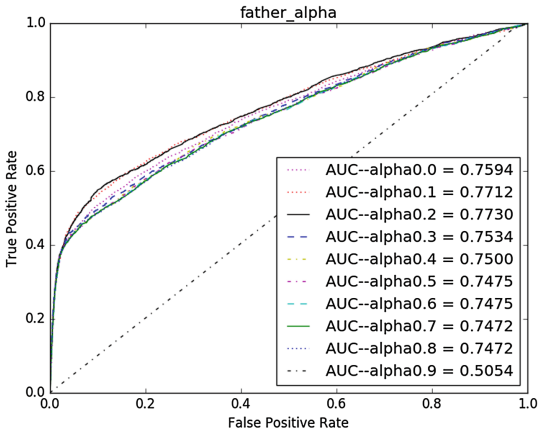

Fig. 3. The AUC of BoD(OLLIE) with different α on relation *father*

Discussion of α: α controls the ratio of semantic distance and location distance when calculating the distance between mentions. Here we tested how RLSW's performance varies with the α value of the relation *father*; the results are shown in Fig. 3. It can be observed that the AUC value is best when the α is 0.2. It has been empirically proved [1] that a sentence similarity measure performs best when semantic measure is weighted more than syntactic measure. Therefore, a small α value is preferred in practice. In experiments of this work, α is set to 0.2 for all relations.

6 Conclusions

In this paper, a new framework of relation linking for knowledge graphs called RLSW is proposed. The relation mentions are clustered with location sensitive phase similarity measurements algorithm LSWMD and then each cluster is represented as a Bag of Distributions of words. For each testing word sequence, a classifier is used to decide if it should be linked to the given relation.

However, there are still a lot work to do in future. First, shallow syntax feature such as PoS tags can be used for similarity calculation. Second, the entities of subject and object can be taken into consideration when building relation patterns. Last, in future, we plan to develop a plain text oriented deduction system based on relation linking and entity linking techniques with the data provided by KGs such as Wikidata.

Acknowledgments. This paper is supported by the national key R&D of China program (No. 2016YFB1000900), NFSC program young scholar project (No. 61502066), scientific and technological research program of Chongqing municipal education commission (No. KJ1500438), basic and frontier research project of Chongqing, China (No. cstc2015jcyjA40018).

References

1. Achananuparp, P., Hu, X., Zhou, X., Zhang, X.: Utilizing Sentence Similarity and Question Type Similarity to Response to Similar Questions in Knowledge-Sharing Community (2008)
2. Demartini, G., Difallah, D.E., Cudré-Mauroux, P.: ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: International Conference on World Wide Web, pp. 469–478 (2012)
3. Han, X., Sun, L., Zhao, J.: Collective entity linking in web text: a graph-based method. In: Proceeding of the International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, pp. 765–774, July 2011
4. He, H., Gimpel, K., Lin, J.: Multi-perspective sentence similarity modeling with convolutional neural networks. In: Conference on Empirical Methods in Natural Language Processing, pp. 1576–1586 (2015)
5. Huang, J., Abadi, D.J., Ren, K.: Scalable SPARQL querying of large RDF graphs. Proc. VLDB Endow. **4**(11), 1123–1134 (2011)
6. Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q.: From Word Embeddings to Document Distances, pp. 957–966 (2015)
7. Li, Y., McLean, D., Bandar, Z.A., O'Shea, J.D., Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. Knowl. Data Eng. **18**(8), 1138–1150 (2006)
8. Schmitz, M., Bart, R., Soderland, S., Etzioni, O.: Open language learning for information extraction. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (2012)
9. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science **344**(6191), 1492–1496 (2014)
10. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: issues, techniques, and solutions. IEEE Trans. Knowl. Data Eng. **27**(2), 443–460 (2015)
11. Wu, F., Weld, D.S.: Open information extraction using Wikipedia. In: Proceedings of the Meeting of the Association for Computational Linguistics, ACL 2010, Uppsala, Sweden, 11–16 July 2010, pp. 118–127 (2010)