Learning to Map Natural Language Statements into Knowledge Base Representations for Knowledge Base Construction

Chin-Ho Lin, Hen-Hsen Huang, and Hsin-Hsi Chen
Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan
{chlin, hhhuang}@nlg.csie.ntu.edu.tw, hhchen@ntu.edu.tw

Abstract

Directly adding the knowledge triples obtained from open information extraction systems into a knowledge base is often impractical due to a vocabulary gap between natural language (NL) expressions and knowledge base (KB) representation. This paper aims at learning to map relational phrases in triples from natural-language-like statement to knowledge base predicate format. We train a word representation model on a vector space and link each NL relational pattern to the semantically equivalent KB predicate. Our mapping result shows not only high quality, but also promising coverage on relational phrases compared to previous research.

Keywords: knowledge base, knolwedge base construction, relation mapping

1. Introduction

Knowledge bases (KBs) such as Freebase (Bollacker et al., 2008) and DBpedia (Auer et al., 2007) are fundamental resources for many intelligent applications. Currently, the construction and updating of KBs, directly or indirectly, rely on human labor. Keeping KBs up-to-date by humans is cost intensive and impractical. To reduce the cost and to minimize the updating latency, automatically updating a KB with knowledge extracted from natural language (NL) content is a feasible strategy.

In KBs, a fact is represented by a triple (*subject*, *predicate*, *object*), where *subject* and *object* are two entities in the KB, and *predicate* describes their relation. With an information extraction system (Carlson et al., 2010; Fader et al., 2011), we can also extract facts from NL text in the format (*np*$_1$, *pattern*, *np*$_2$), where *np*$_1$ and *np*$_2$ are two entities and *pattern* is the relational phrase between them. However, the extracted triples from NL text do not always follow the paradigm of KB, and that becomes a challenging issue for KB construction. For example, the NL triple (Garnett, was born in, Mauldin) extracted by ReVerb (Fader et al., 2011) shows a fact identical to the KB triple (Kevin Garnett, birthPlace, Mauldin (South Carolina)). Although these two triples state the same fact, there is a vocabulary gap between them. In the former triple, “was born in” is an NL-like expression and “birthPlace” in the latter triple is a formatted predicate used in KB. These two relational phrases are different in surface forms. They cannot be mapped by string matching directly. In addition, a KB predicate may be described in multiple NL statements. A number of ReVerb patterns such as “is the hometown of”, “was raised in”, and “grew up in” are related to the predicate “hometown” in DBpedia. That makes the mapping between KB and NL even more challenging.

Recently, more and more works show their interests in the issue of KB construction. Knowledge graph embedding models (Bordes et al., 2013; Yang et al., 2015; Xie et al., 2016a; Xie et al., 2016b) focus on learning the vector representation on the KB side only. Previous works (Nakashole et al., 2012; Riedel et al., 2013; Dutta et al., 2015) aim to solve similar problems as ours. Nakashole et al. (2012) in their PATTY approach try to learn paraphrases to define the predicates of DBpedia, and Dutta et al. (2015) propose clustering-based approaches to transform the knowledge extracted by an open information extraction system into DBpedia paradigm. Riedel et al. (2013) propose universal schemas, which are the mapping between NL surface forms to the KB predicates, by using matrix factorization. However, all of them suffer from low coverage on relational phrases. In this work, we aim to propose a more general framework that maps relational phrases extracted from an NL resource to DBpedia predicates. Our method is capable of covering most NL patterns and KB predicates. The relational mappings can be used for a range of applications. For KB construction, the mappings can be consulted for mining the new facts from textual data written in NL. For question answering over the KB, the mappings can be used for looking up the facts in KB that are candidates for the answer.
The rest of this paper is organized as follows. Section 2 describes the corpora used as NL data for learning the relational mapping. Section 3 presents our learning to map approach. In Section 4, we conduct experiments for evaluating the results. The challenging issues of this work are discussed in Section 5. Finally, Section 6 concludes this work.

2. Linguistic Resource

English Wikipedia is regarded as the NL text resource in this study. We obtain the NL relational triples from ReVerb (Fader et al., 2011), a dataset of relational triples extracted from Wikipedia. Let an NL dataset D_{NL} be a 3-tuple (P, N, I_{NL}), where P is a set of NL patterns, N is a set of entities, and I_{NL} is a set of NL triples. For example, (Garnett, was born in, Mauldin) $\in$ $I_{NL} = \{(n_i, p_k, n_j) | n_i, n_j \in N, p_k \in P\}$.

On the other hand, DBpedia (Auer et al., 2007) serves as our target KB. We define a KB dataset D_{KB} as a 5-tuple (R, E, T, E_T, I_{KB}), where R is a set of KB predicates, E is a set of entities in KB triples, T is a set of KB entity types, E_T is a set of entity-type pairs in KB, and I_{KB} is a set of KB triples. For example, (Kevin Garnett, birthPlace, Mauldin (South Carolina)) $\in$ $I_{KB} = \{(e_m, r_o, e_n) | e_m, e_n \in E, t_s, t_u \in T, (e_m, t_s), (e_n, t_u) \in E_T, r_o \in R\}$.

To resolve the entity disambiguation problem, a lexicalization dataset[1] released on the DBpedia Spotlight (Mendes et al., 2011) official website is consulted. We extract an entity-alias list from the dataset and let the alias list be

$$Alias = \{(a_{m1}, a_{m2}, \ldots, a_{m|A(e_m)|}) | a_{m1}, a_{m2}, \ldots, a_{m|A(e_m)|} \in A(e_m)\}$$

where $e_m \in E$ and $A(e_m)$ is a set of entity aliases corresponding to the KB entity e_m.

We further randomly sampled 10 million sentences Sen_{Clue} from ClueWeb09 dataset[2] as an NL resource, which is considered as an auxiliary dataset for training word embedding models.

3. Relational Mapping

We propose an approach inspired by word2vec model (Mikolov et al., 2013), i.e., Skip-gram and CBOW, to build a relational mapping. Our method projects all relational phrases, i.e., NL patterns and KB predicates, to a vector space and measures cosine similarity between relational phrases on this space.

As illustrated in Figure 1, our method consists of three components, EB (Entity Bridging with Alias Resolution), DR (Decompose Relational Phrases and Introduce Additional NL Text), and TF (Filter Relational Mapping by Argument Types constraint). The first workflow builds a relational mapping by conducting entity bridging on training triples with consultation of entity alias dictionary. Our second workflow is based on EB and further performs DR, which considers the information of words decomposed from relational phrases and adds auxiliary natural language sentences into training data. Finally, TF is used to filter the mappings built by EB or EB+DR. The descriptions of EB, DR, and TF are presented in Sections 3.1, 3.2 and 3.3, respectively.

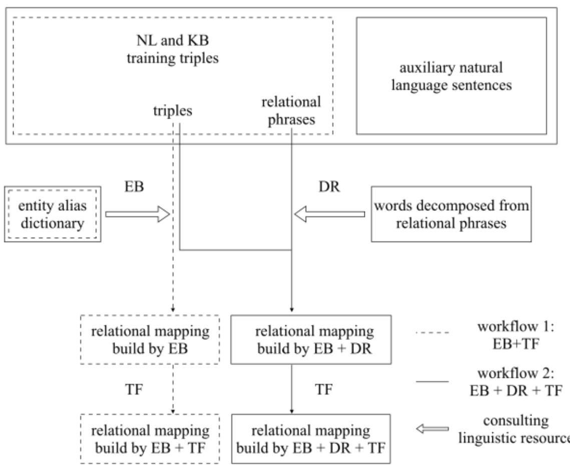

NL and KB training triples
triples
relational phrases
auxiliary natural language sentences
entity alias dictionary
EB
DR
words decomposed from relational phrases
relational mapping build by EB
relational mapping build by EB + DR
workflow 1: EB+TF
TF
TF
workflow 2: EB + DR + TF
relational mapping build by EB + TF
relational mapping build by EB + DR + TF
consulting linguistic resource

Figure 1: Overview of our approach to relational mapping.

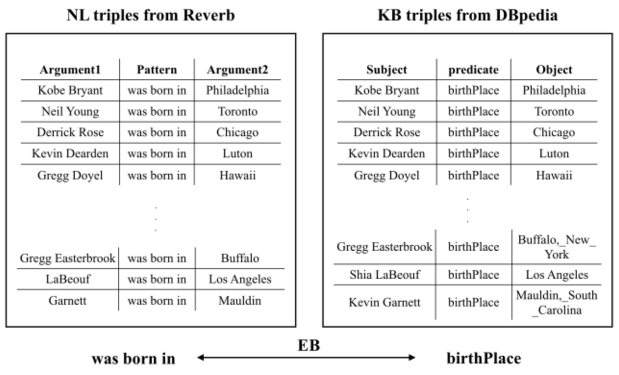
3.1 EB : Entity Bridging with Alias Resolution

Unlike KB predicates that are formatted, patterns in ReVerb are NL-like expressions. As the examples shown in Table 1, KB predicates and semantically related NL patterns might be different in surface form such as the predicate "headquarter" and the pattern "is based in". Another example in Table 1 is the KB predicate "spouse" and NL patterns "is the wife of", "is the husband of", and "is the first wife of". The NL patterns might contain more specific details but are still mapped to the KB predicate. To build a relational mapping of NL patterns and KB predicates, our model should learn the connection of information from KB and NL data resources.

KB Predicate	NL Patterns
headquarter	is headquartered in / is based in
spouse	is the wife of / is the husband of / is the first wife of
hometown	grew up in / was raised in
writer	was written by / is a novel by

Table 1: Examples of KB predicates and their semantically related NL patterns.

EB aims to capture the structural information between entities and relations, and then links relational phrases through entity bridging. Figure 2 shows how EB works. For instance, (Kobe Bryant, was born in, Philadelphia) and (Kobe Bryant, birthPlace, Philadelphia) are triples from NL and KB datasets, respectively. Through the co-occurrence of entity pairs and relational phrases, the model gradually learns the connection between "was born in" and "birthPlace".


NL triples from Reverb

Argument1	Pattern	Argument2
Kobe Bryant	was born in	Philadelphia
Neil Young	was born in	Toronto
Derrick Rose	was born in	Chicago
Kevin Dearden	was born in	Luton
Gregg Doyel	was born in	Hawaii
⋮	⋮	⋮
Gregg Easterbrook	was born in	Buffalo
LaBeouf	was born in	Los Angeles
Garnett	was born in	Mauldin

KB triples from DBpedia

Subject	predicate	Object
Kobe Bryant	birthPlace	Philadelphia
Neil Young	birthPlace	Toronto
Derrick Rose	birthPlace	Chicago
Kevin Dearden	birthPlace	Luton
Gregg Doyel	birthPlace	Hawaii
⋮	⋮	⋮
Gregg Easterbrook	birthPlace	Buffalo,_New_York
Shia LaBeouf	birthPlace	Los Angeles
Kevin Garnett	birthPlace	Mauldin,_South_Carolina

Figure 2: Examples of Entity Bridging (EB).

We train a word embedding model with the KB and NL triples. Through the update of entity pairs, the connection of a pattern p_k and a predicate r_o are captured in the model training process. More precisely, given a KB training triple (e_m, r_o, e_n), a training sequence W for the model will be w_1, w_2, w_3, where w_1=e_m, w_2=r_o, and w_3=e_n. The model maximizes the average log probability θ_{triple} as shown in Equation (1).

$$\theta_{triple} = \frac{1}{|W|}\sum_{x=1}^{|W|} \sum_{-c \le y \le c, y \ne 0} \log p(w_{x+y}|w_x) \qquad (1)$$

where c is the size of context windows, w_x is the central word, and w_{x+y} denotes one of the context words. Probability $p(w_{x+y}|w_x)$ is calculated using the softmax function as shown in Equation (2).

[1]http://spotlight.sztaki.hu/downloads/latest_data/en.tar.gz

[2] http://lemurproject.org/clueweb09/

$$p(w_{x+y}|w_x) = \frac{\exp({v'_{w_{x+y}}}^T v_{w_x})}{\sum_{z=1}^{|V|} \exp({v'_z}^T v_{w_x})} \quad (2)$$

where v_w and v'_w are the "input" and "output" embedding of word w, and $|V|$ is the vocabulary size of the model. In addition, the $Alias$ is consulted for alias resolution. For an entity e_m in KB training triple, its alias $a_{mx} \in A(e_m)$ is updated by maximizing average log likelihood θ_{alias} in Equation (3).

$$\theta_{alias} = \frac{1}{|A(e_m)|} \sum_{a_{mx} \in A(e_m)} \log p(a_{mx}|e_m) \quad (3)$$

where $p(a_{mx}|e_m)$ is computed by using the softmax function (2).

The trained model results in an embedding space, where NL patterns and KB predicates are represented as vectors in this space. Thus, the similarity between an NL pattern and a KB predicate can be measured by their cosine similarity. On the one hand, most similar KB predicates of an NL pattern can be considered as its mapping targets. On the other hand, most similar NL patterns of a KB predicate can be regarded as its mapping patterns. In Section 4, triple linking task and human verification task will evaluate the results from these two aspects, respectively.

3.2 DR: Decomposing Relational Phrases and Introducing Additional NL Text

EB may suffer from data sparseness because each relational phrase is treated as a distinct symbol, and information from the words that compose a relational phrase is completely ignored. For example, the meaning of predicate "birthPlace" can be captured from words "birth" and "place". Thus, DR is proposed and integrated with EB for leveraging the words decomposed from relational phrases. More clearly, given a KB predicate r_o, the word semantics of r_o will be jointly learned by maximizing the average log probability $\theta_{compose}$ as Equation (4).

$$\theta_{compose} = \log p(r_o|c_{o1}, c_{o2}, \dots, c_{on}) \quad (4)$$

where $c_{o1}, c_{o2}, \dots, c_{on}$ are words decomposed from r_o and n is the number of the words. The computation of the probability is done by the softmax function (2) and a composition function that simply averages the vectors of words $c_{o1}, c_{o2}, \dots, c_{on}$.

Figure 3 illustrates how DR encodes words decomposed from relational phrases and jointly learns the meaning of relational phrases. The advantage of DR is that the semantics of a predicate is expanded by its compositional words, which are written in NL. For example, the word meaning of KB predicate "birthPlace" will also be considered from words "birth" and "place" and the word meaning of NL pattern "was born in" will similarly be viewed from "was", "born" and "in". Relational phrases that are semantically related might consist of words with similar meaning such as "birth" and "born" in this example. In other words, we reduce the data sparseness of relational phrases by connecting patterns and predicates through the decomposed words.

In addition to the information from the KB, we add additional NL statements Sen_{Clue} to the training set. These statements serve as an auxiliary resource that aids to model the meaning of the decomposed words. In other words, we co-train a distributed word model in the same vector space. In the training process, the decomposed words play a role of bridges that connect similar relational phrases. The natural language statements provide semantic information for those words.

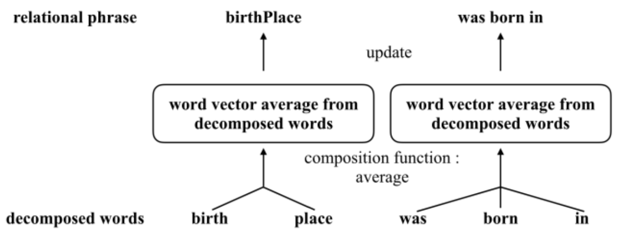

relational phrase
birthPlace
was born in
update
word vector average from decomposed words
word vector average from decomposed words
composition function : average
decomposed words
birth
place
was
born
in

Figure 3: Example of Decomposing Relational Phrases (DR).

3.3 TF: Filter Relational Mapping by Argument Types of Relational Phrases

We further filter the mapping results with argument type constraints. That is, the argument type of an NL pattern should be consistent with those of its corresponding KB predicates. The mappings with inconsistent argument types are removed from the relational mapping list.

We obtain the argument type constraint for each relational phrase by voting with training triples. More formally, for each KB triple $(e_m, r_o, e_n) \in I_{KB}$, $e_m \in t_s$, $e_n \in t_u$ the argument type for r_o is voted as (t_s, t_u). We count the majority argument type for each predicate and define the argument type constraint of r_o as $Arg_{type(r_o)} = (t_s, t_u)$.

In NL side, unlike the entity types defined in KB datasets, we have to find the types of entities extracted from natural language sentences. We obtain the type of each NL entity by matching it with KB entity aliases. Then, we vote argument types of NL patterns with NL triples in the similar way as above. If an entity n_i has c possible types, each type of the entity n_i will be weighted equally by $\frac{1}{c}$. If $c = 0$, the triple will not contribute to argument type determination. In this way, we generate argument type constraints for NL patterns. Let $Arg_{type(p_k)} = (t_d, t_f)$ be the type constraint of the corresponding pattern p_k for some $t_d, t_f \in T$.

4. Experiments

Dataset and Experiment Setting: The statistics of NL and KB datasets show that $|I_{NL}| = 407{,}239$, $|P| = 100{,}264$, $|I_{KB}| = 14{,}408{,}940$, and $|R| = 662$. Comparatively, even though the size of KB triple set is much larger than NL triple set, KB predicates are formatted and there are only 662 predicates in KB dataset. We randomly split our dataset into five folds and conduct five-fold cross validation for our experiment. We train 300-dimensional vector models with the proposed methods and analyze the mapping results. The training parameters of our model, i.e., (window size, negative sample, min count), are set to (5, 10, 0), respectively. The relational phrases with less than 5 occurrences in ReVerb and DBpedia are excluded from the mapping. After filtering, our approach builds a relational mapping that covers 7,361 of 9,171 frequent ReVerb patterns with 629 of 634 frequent DBpedia predicates.

Compared to previous works, Dutta et al. (2015) cover 212 ReVerb patterns and 41 DBpedia predicates, Riedel et al. (2013) cover 100 Freebase facts in training and test data, and Nakashole et al. (2012) cover 225 predicates, we generate a higher coverage of relational mapping.

Evaluation: Because of lack of ground truth, most previous work evaluates their relational mapping result by human annotation only. In this work, we try to evaluate our approach from two perspectives. Firstly, we conduct a triple linking task that aims to link NL triples to the KB triples sharing the same knowledge. It reflects the ability of our mapping approach to translate relational phrases from NL side to KB paradigm. Secondly, we further evaluate the performance of our mapping by human verification. It demonstrates the result and the accuracy of NL patterns that link to each KB predicate.

Triple linking task: This task simulates knowledge base construction. The KB test set is considered as new facts. The word embedding model learned from the KB training set builds a relational mapping between NL patterns and KB predicates. Through the mapping, we can add new knowledge into the KB by translating NL triples to KB triples. We judge the correctness of translation by checking if the translated triple is actually in the test data. Formally, given an NL triple (n_i, p_k, n_j), $n_i \in A(e_m), n_j \in A(e_n)$, we have to predict a KB predicate r_o such that (e_m, r_o, e_n) is the corresponding fact in KB test set. Thus, we generate NL test triples Λ_{NL} and ground truth Λ_{KB} as follows. Λ_{NL} is a set of triples $(n_i, p_k, n_j) \in I_{NL}$ and Λ_{KB} is a set of triples $(e_m, r_o, e_n) \in I_{KB}$, where $n_i \in A(e_m), n_j \in A(e_n)$. In this way, we derive $|\Lambda_{NL}| = 54{,}752$ and $|\Lambda_{KB}| = 58{,}504$.

Due to the low coverage on relational phrases, the results of Riedel et al. (2013) and Dutta et al. (2015) cannot be directly compared with ours under this task. Thus, we built a baseline model through a counting-based approach that counts the co-occurrence between each pattern and each predicate found in the same or alias entity pairs. The baseline model always selects the majority. TransE (Bordes et al., 2013) focuses on learning the vector representation in KB side only. Although it solves a problem different from ours, we also adapt it to this task. We train a model with NL and KB triples. Given an NL triple (n_i, p_k, n_j), the TransE model is trained to optimize the equation $n_i + p_k \approx n_j$, so we can predict a KB predicate r_o through the entity operation $n_j - n_i$. We denote this operation as TransE(entity). We can also rank mapping candidates of p_k by cosine similarity of NL patterns and KB predicates. We denote it as TransE(rel). We calculate hit@k and MRR to measure the performance. Hit@k indicates the percentage of NL triples where correct relations can be found in the top k positions. In the special case where k=1, hit@1 is equivalent to P@1 (Precision at 1). MRR is the mean reciprocal rank of correct mapping and is calculated to the 100 position.

As shown in Table 2, counting-based baseline suffers from the low translation rate on relational phrases, i.e., no suitable mapping can be applied, and its hit@k shows no difference when k is larger than 5. By contrast, though TransE(entity) and TransE(rel) do not have such a problem, all our methods outperform them. Besides, DR expands the meaning of relational phrases by decomposed words. Although hit@k drops at k=1, the performance shows large improvement when k is larger than 5. That indicates the effectiveness of semantic information provided by decomposed words and the additional natural language statements. TF shows the strength of type filter. EB+DR+TF even achieves 0.800 and 0.327 of hit@20 and MRR, respectively, in this task.

	hit@1	**hit@5**	**hit@10**	**hit@20**	**MRR**
baseline	0.113	0.151	0.151	0.151	0.129
TransE(entity)	0.042	0.116	0.174	0.250	0.086
TransE(rel)	0.035	0.122	0.202	0.308	0.087
EB	0.189	0.294	0.345	0.392	0.242
EB+DR	0.117	0.273	0.361	0.457	0.205
EB+TF	0.222	0.358	0.408	0.530	0.299
EB+DR+TF	0.158	0.538	0.709	0.800	0.327

Table 2: Evaluation results of triple linking task.

Human verification task: We select top 100 frequent predicates from KB triples and manually annotate their top 5 mapped NL patterns. Because human annotation is cost intensive, we only verify the mapping results by EB+DR+TF, the best performing method in the triple linking task.

The results of human verification are shown in Table 3. The hit@1, hit@3, and hit@5 of the most frequent 50 and 100 predicates, respectively, are reported. This results further confirm the quality of our mapping from another aspect. Some NL patterns are mapped to KB predicates that have exactly the same meaning, e.g., "birthplace" and "was born in" is counted a correct mapping. Some NL patterns do not have exactly the same meaning with the KB predicates, but they can be inferred. For example, the pattern "is a fantasy novel by" infers the predicate "author". Thus, we regard them as correct mapping. Besides, there are some incorrect mapping examples, such as pattern "was born in" and predicate "residence". The predicate "residence" indicates a place a person live in.

	hit@1	**hit@3**	**hit@5**
Top 50 Predicates	0.352	0.480	0.528
Top 100 Predicates	0.326	0.456	0.510

Table 3: Evaluation results of human verification task.

5. Discussion

We find four major types of errors in our mapping:

Complex concept: Some KB predicates containing complex concept are difficult to map accurately. For instance, the KB predicate "leftTributary" contains not only the relational expression "tributary", but also the concept "left". In this case, NL patterns such as "is a tributary of", "is a tributary to", and "is a river in" can only capture partial phrase meaning, and they are regarded as inaccurate mapping. The complex concept would be better modeled by decomposing it to multiple simple concepts.

Uncommon in NL sentences: Some KB predicates such as "youthWing" and "varietals" are uncommon in NL sentences, so that there are insufficient instances available for training. As a result, DR may not perform well since it aims to capture semantic information from NL sentences. Fortunately, most uncommon predicates are less important in general domain. For a specific-purpose application, the in-domain corpus can be used to train a dedicated relational mapping.

Similar context: Word embedding models learn word meaning through NL context. However, some antonyms share similar context in NL, and they may have small cosine distance in the vector space. For instance, the NL patterns “is a large town in” and “is a small town in” are hard to distinguish. This issue has been addressed in other applications that use word embedding models.

Multiple meanings: Some relational phrases have multiple meanings and only parts of the meanings are used in KB. This leads to wrong relational mapping. For example, the word “billed” has several meanings, but the predicate “billed” is only applied to the state that a wrestler comes from, such as (A-1(wrestler), billed, Niagara Falls (Ontari)). One of possible solutions to this issue is performing word sense disambiguation (WSD) on the corpus, and learning the relational mapping at the sense level, instead of at the word level.

6. Conclusion

Relational mapping is a challenging problem. This paper proposes an approach that provides a quality mapping in terms of coverage and correctness. This method is also unsupervised and is not restricted to a specific KB. It is easy to apply to different data resources for various applications such as KB construction and question-answering. Because the ground truth is not available, we also propose a triple linking task. The trask provides an automatic and scalable evaluation for relational mapping.

In the end, we suggest some research directions for improving the proposed approach in the future. DR encodes words decomposed from relational phrases by a compositional function that simply averages the embeddings of these words. The joint learning model may handle the compositionality better in phrase embedding learning. TF considers entity type information by filtering the relational mapping with argument type constraints. Embedding model learns type information while training knowledge embedding may be explored further.

7. Acknowledgements

This research was partially supported by Ministry of Science and Technology, Taiwan, under grants MOST-105-2221-E-002 -154 -MY3, MOST-106-2923-E-002 -012 -MY3 and MOST-107-2634-F-002-011-, and Academia Sinica under grant AS-107-TP-A05.

8. Bibliographical References

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD 2008), pages 1247–1250, Vancouver, Canada. ACM.

Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J. and Yakhnenko, O. (2013). Translating Embeddings for Modeling Multi-relational Data. In Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS 2013), pages 2787–2795, Lake Tahoe, Nevada, USA. Curran Associates Inc.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr., E. R., and Mitchell, T. M. (2010). Toward an Architecture for Never-ending Language Learning. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010), pages 1306–1313, Atlanta, Georgia, USA. AAAI Press.

Dutta, A., Meilicke, C., and Stuck-enschmidt, H. (2015). Enriching Structured Knowledge with Open Information. In Proceedings of the 24th International Conference on World Wide Web (WWW 2015), pages 267–277, Florence, Italy. International World Wide Web Conferences Steering Committee (IW3C2).

Mendes, P. N., Jakob, M., Garc ́ıa-Silva, A., and Bizer, C. (2011). Dbpedia Spotlight: Shedding Light on the Web of Documents. In Proceedings of the 7th International Conference on Semantic Systems (I-Semantics 2011), pages 1–8, Graz, Austria. ACM.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In Proceedings of International Conference on Learning Representations 2013 (ICLR 2013).

Nakashole, N., Weikum, G., and Suchanek, F. (2012). Patty: A Taxonomy of Relational Patterns with Semantic Types. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP 2012), pages 1135–1145, Jeju Island, Korea. Association for Computational Linguistics (ACL).

Riedel, S., Yao, L., McCallum, A., and Marlin, B. M. (2013). In Proceedings of NAACL-HLT 2013, pages 74–84, Atlanta, Georgia, USA. Association for Computational Linguistics (ACL).

Xie, R., Liu, Z., Jia, J., Luan, H., and Sun, M. (2016a). Representation Learning of Knowledge Graphs with Entity Descriptions. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016), pages 2659–2665, Phoenix, Arizona, USA. AAAI Press.

Xie, R., Liu, Z., and Sun, M. (2016b). Representation Learning of Knowledge Graphs with Hierarchical Types. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016), pages 2965–2971, New York, New York, USA. AAAI Press.

Yang, B., Yih, W.-T., He, X., Gao, J., Li Deng. (2015). Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of International Conference on Learning Representations 2015 (ICLR 2015).

9. Language Resource References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A Nucleus for a Web of Open Data. In Proceedings of the 6th International the Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference (ISWC 2007/ASWC 2007), pages 722–735, Busan, Korea. Springer-Verlag.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying Relations for Open Information Extraction. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011), pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics (ACL).