Open Relation Extraction by Matrix Factorization and Universal Schemas

Jiho Kim, Sangha Nam, and Key-Sun Choi

Korea Advanced Institute of Science and Technology, Republic of Korea
{hogajiho, sangha.nam, kschoi}@kaist.ac.kr

Abstract. Universal schemas are a remarkable approach used to solve relation extraction, constructed by a union of all usable schemas. However, in free word order languages where surface form predicates are difficult to extract, original universal schemas cannot be applied. We introduce a novel extension: using dependency paths and entity types instead of surface form predicates. Based on the extension, we could cluster similar ontological relations, which broaden the coverage of question answering. We verified the performance of our extended model by constructing and evaluating our model in Korean based on Korean DBpedia and Korean Wikipedia distant supervision data.

Keywords: Open Relation Extraction · Universal Schemas · Matrix Factorization · Shortest Dependency Path · Natural Language Question

1 Introduction

Relation extraction is a crucial intermediate step in many NLP applications, especially in information extraction. Relation extraction can be defined as the problem of extracting structured information from unstructured raw text. Specifically, the task involves finding relations between two target entities in a sentence. Traditional relation extraction can be divided into two paradigms: 1) unsupervised clustering of textual relations such as OpenIE [5] and 2) relation extraction for a fixed schema (e.g., a predefined ontological relation set). Various studies ranging from rule-based to machine learning methods have been proposed to perform both paradigms. However, most previous studies focus on a single paradigm, which makes it difficult to apply those methods to realistic problems.

Riedel [19] first introduced the concept of universal schemas as a solution to join the two paradigms. Universal schemas can be constructed by using a combination of all possible knowledge bases and raw text. Universal schemas can theoretically have an infinite set of relation types because the set contains both surface form predicates (as in OpenIE) and predefined ontological relations. Then, the low-dimensional latent feature vectors for entity tuples and relations can be learned by matrix factorization models. The novelty of universal schemas comes from avoiding pre-labeled datasets by using surface form predicates as a source for relation schemas, and from mutually supporting both unstructured and structured data.

In English, surface form predicates between two entities can usually serve as relations. This approach was taken by OpenIE and OLLIE [22]. For example, in the sentence "*Steve Jobs is a founder of Apple.*", the surface form predicate *is-a-founder-of* can serve as a relation between *Steve Jobs* and *Apple*. However, in free word order languages, it is difficult to extract surface pattern relations between entities due to flexibility in how words are ordered. Without the presence of language specific high-performance surface form predicate extractors, universal schemas cannot be built in those languages.

In this study, we propose a novel extended methodology of universal schemas: universal schemas using shortest dependency paths (USSDPs). Shortest dependency paths (SDPs) illustrate the syntactic structure between entities using a list of directed binary grammatical relations. We can build USSDPs by using the shortest dependency path between entities instead of surface form predicates. Unlike the surface form predicates, the dependency trees are a common feature in all languages. In addition, cross-lingual grammatical annotations such as universal dependencies (UD) are being actively studied. When the study of multilingual dependency parsing reaches a certain level, the USSDP can be made quickly and easily constructed in any language without further studies. Considering that most relation extraction solutions usually target a single language, or at most several languages, USSDPs can be a very innovative approach to multilingual relation extraction.

Furthermore, we can construct clusters of extended dependency structures that can be used for finding ontological relations as well as open relations through USSDPs. The construction of such clusters will broaden the coverage of knowledge graphs that can be used for answering open questions. This is due to the ambiguity between ontological relations; many of them are not clearly defined and can be confused even in the eyes of a human.

We performed two experiments on Korean to illustrate our assertions. Korean is a language that allows much flexibility in word order. Korean is mainly based on *subject-object-verb* word order but not limited to it, where English is mainly based on *subject-verb-object* word order. For the first experiment, we trained and tested a Korean USSDP from the perspective of relation extraction by using Korean DBpedia and distant supervision data constructed from Korean Wikipedia. For the second experiment, we constructed clusters of the ontological relations based on the extended USSDP from the first experiment, and we measured the prediction performance of our model on simple Korean questions based on the constructed relation clusters.

The remainder of the paper is organized as follows: Section 2 presents the methods and the architecture of our model. Section 3 shows the related work and background. Section 4 discusses our experiments and the results. Section 5 presents our conclusions and suggests avenues for future research.

2 Methods

2.1 Building the Matrix

We built a matrix that has entity tuples as rows and a combination of ontological relations and SDPs as columns. For all observed facts we filled in 1 for the corresponding slots, and we filled unobserved slots with 0. We extracted SDPs between the target

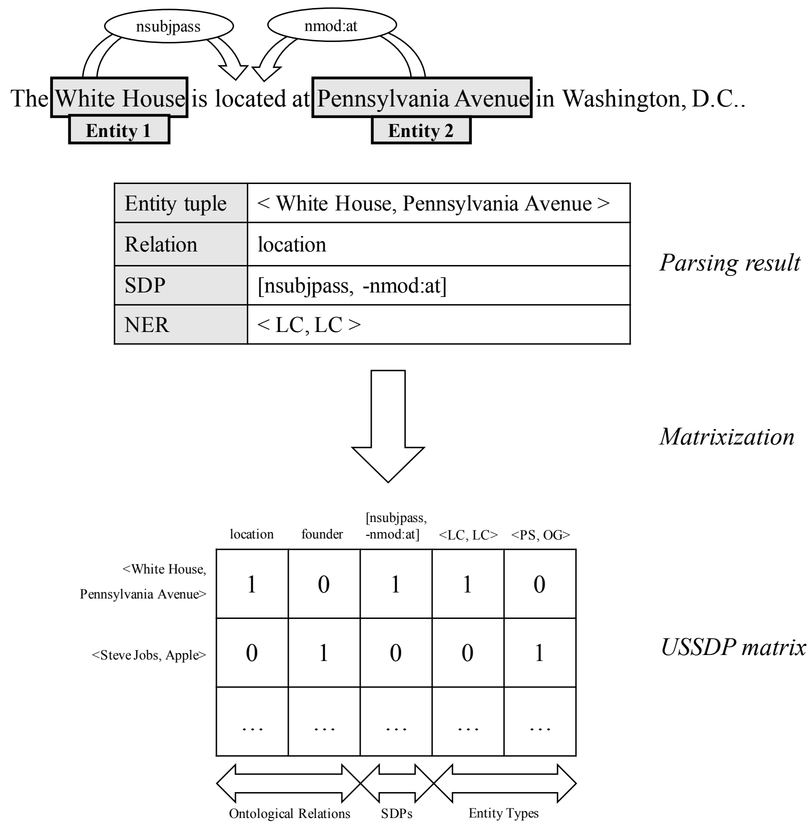

Fig. 1. An example of building USSDP matrix

entities, including directions for each dependency. Figure 1 shows an example of filling the USSDP matrix from raw text. The sentence ”*The White House is located at Pennsylvania Avenue in Washington, D.C.*” has two target entities: ”*White House*” and ”*Pennsylvania Avenue*”. The SDP from ”*White House*” to ”*Pennsylvania Avenue*” is [*nsubjpass*, *-nmod:at*], where the ”-” sign indicates that the direction of the dependency is backwards.

We define the length of an SDP as the number of grammatical arcs between entities. For example, [*nsubjpass*, *-nmod:at*] is an SDP with length 2. The two target entities also have the ontological relation *location*. Thus we fill the slots with column *location* and [*nsubjpass*, *-nmod:at*] with 1.

In this study, we only used SDPs with length less or equal to 3 as columns. This is because long SDPs commonly have a low frequency in the dataset, so considering those can cause huge data sparsity and degrade performance of the entire model. In addition, we could reduce noise data, which are the chronic side effects of distant supervision as previously mentioned by Roth [21], i.e., entities which are connected by a long dependency path are less likely to be related in that sentence. Those sentences should be considered as a noise of distant supervision.

2.2 Matrix Factorization

Our model mainly employs the *nfe* matrix factorization model previously proposed by Riedel [19]. The difference comes from the objective function, where we use the same objective function from BPR (Bayesian Personalized Ranking) introduced by Rendle et al. [18]. For an entity tuple $t =< e_1, e_2 >$, where e_1 and e_2 are target entities, assume that r_i is observed between the two entities but r_j is not. We define the probability that our model prefers r_i than r_j for t as follows:

$$p(r_i >_t r_j) := \sigma(\widehat{x}_{tij}(\Theta)) \tag{1}$$

σ is the logistic sigmoid function, Θ is the set of all parameters used in the model including the latent feature values for each entity tuple and each relation, and $\widehat{x}_{tij}(\Theta) := \widehat{x}_{ti}(\Theta) - \widehat{x}_{tj}(\Theta)$. $\widehat{x}_{ti}(\Theta)$ is the value learned by the model for the triple (t, r_i) during training. For more details regarding $\widehat{x}_{ti}$, refer to the *nfe* model in previous work by Riedel [19]. Our objective is to maximize $p(r_i >_t r_j|\Theta)$ for the relations r_i and r_j where the triple (t, r_i) is observed during training but the triple (t, r_j) is not. The objective function is then as follows:

$$Obj := \sum_{(t,i,j)\in D} ln\, \sigma(\widehat{x}_{tij}(\Theta)) - \lambda_\Theta \parallel \Theta \parallel^2 \tag{2}$$

λ_Θ is the regularization parameter, and D is defined as the follows:

$$D := \{(t, i, j) \,|\, (t, r_i) \in KB, (t, r_j) \notin KB\} \tag{3}$$

Due to the computation size, we cannot optimize Θ for all (t, i, j)'s in D for each batch. Rather, we randomly sample (t, i, j)'s from D for each batch of gradient descent. We applied the stochastic gradient descent method to jointly learn the latent feature vectors of each row and column in the matrix. The details of this algorithm are described in Algorithm 1, where α is the learning rate.

Algorithm 1 Matrix Factorization algorithm

```
1: procedure OptimizeUssdp(D, Θ, nfe-model)
2:     initialize Θ with random values
3:     for batch size do
4:         randomly draw (t, i, j) from D
5:         Θ ← Θ + α(∂ObjectiveFunction/∂Θ)
       return Θ
```

We then optimized the embedding vector dimension by comparing the largest area under the ROC curve (AUC) value. The AUC value is defined as follows:

$$AUC := \frac{1}{|D|} \sum_{(t,i,j) \in D} \delta(\hat{x}_{ti}(\Theta) - \hat{x}_{tj}(\Theta)) \tag{4}$$

A higher AUC value indicates a better quality model. A random guess method has 0.5 as its AUC value, where the highest achievable value is 1. Specifically, we conducted one held-out experiment for each dimensions ranging from 2 to 100, and we measured the AUC value for each dimension. After learning, we normalized the matrix for each row on a scale of 0 to 1.

2.3 The System

Figure 2 shows the entire system architecture of our proposed model. Natural language corpus and structured KBs are used as data. The data are preprocessed, which includes three subprocesses: entity linking, NER, and dependency parsing. Note that because entity linking is based on the target KB, sentences with less than two KB entities are filtered during preprocessing. Then the preprocessed data is converted into a matrix as explained in Section 2.1. This matrix is completed through matrix factorization as explained in Section 2.2. Based on the completed matrix, new knowledge and clusters of ontological relations are acquired through simple normalization and analysis. For new knowledge extraction, we select slots with values that were initially 0 but increased above the threshold after completion. For clustering ontological relations, we analyze the correlation matrix for the ontological relations. Note that we used hierarchical clustering during the experiment.

3 Related Work

3.1 Universal Schemas

Universal schemas were introduced as a successful approach to relation extraction, which joins both structured KBs and unstructured raw text. The novelties of Universal Schemas come from the following: 1) solving the data limitation problem, and 2) outperforming state-of-art distant supervision methods by mutually supporting both unstructured and structured data. To build a universal schema, we combine all involved KBs and raw text, and we then turn it into a matrix with entity tuples as rows and relations (ontological relations and surface form predicates) as columns. The model then learns low dimensional latent feature vectors of entity tuples and relations through matrix factorization. The entire matrix is a concatenation of sparse one-shot vectors (either by rows or columns). We can reduce the dimension of these one-shot vectors into latent feature vectors through matrix factorization, which reduces data sparsity. Over the past few years, several algorithmic extensions used to outperform the original universal schemas have been performed [7, 14, 20, 24, 29]. Recently, new approaches that improved universal schemas were introduced to generalize all textual patterns over arbitrary text samples [25, 27]. One notable approach for multilingual relation extraction using compositional universal schemas which combined English and Spanish [26].

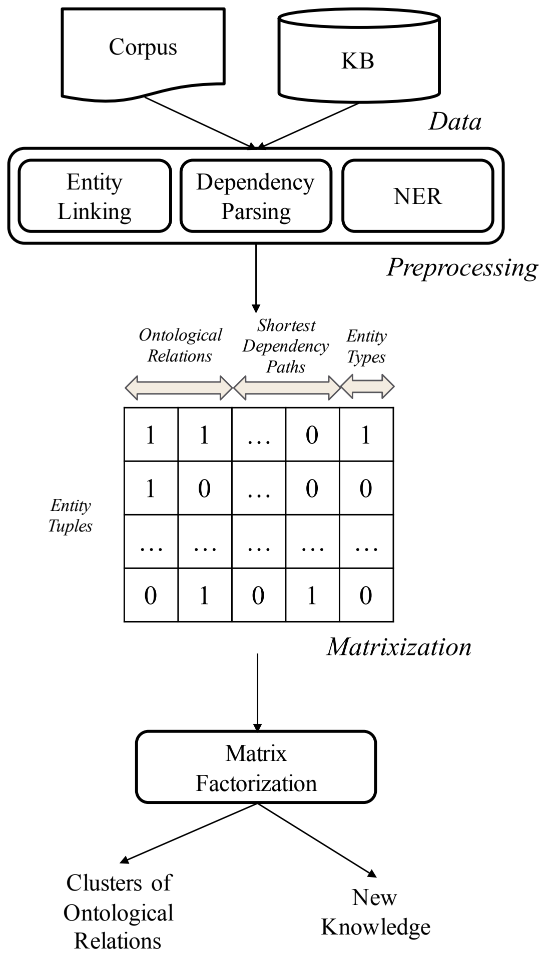

Fig. 2. System architecture

However, they built the compositional universal schema by using surface form predicates in both languages, which still leaves the task of applying universal schemas to an arbitrary language.

3.2 Universal Dependencies

UD is a framework for cross-linguistically consistent grammatical annotation with the goals of developing a multilingual parser, cross-lingual learning, and parsing research from a language typology perspective [15]. UD targets to represent grammatical language structures in a useful and revealing way for the purpose of natural language processing. UD builds on and includes Stanford dependencies [3, 4, 2], Google universal POS tags [17], and the Interset interlingua for morphosyntactic tagsets [31]. UD provides globalized categories and guidelines for consistent annotation over all languages, also allowing necessary language-specific extensions. The initial version of UD was proposed with homogeneous treebanks in 6 languages [13]. Recently, UD version 1 was introduced with the release of available treebanks in 33 languages [16]. At the beginning of 2018, the UD community has produced more than 100 treebanks in over 60 languages. The rise of UD not only develops consistent treebanks, it also opens up possibilities for multilingual NLP applications such as multilingual dependency parsing and cross-lingual information extraction. This global trend empowers the usability of our model, because USSDPs can only be built with SDPs and entity types which are globally available features.

3.3 Shortest Dependency Paths

In most cases, syntactic parsing does not end in itself. The main purpose of syntactic parsing in NLP is to provide rich information regarding syntactic structures of raw text for the improvement of NLP applications such as question answering, relation extraction, and machine translation. The dependency graph is a common way to represent syntactic parsing. It discloses the syntactic structure of a sentence and the grammatical dependency between words. The SDP represents the shortest path between two entities in the dependency graph of a sentence, which makes SDP a useful feature in relation extraction. Previous work used SDP kernels in relation extraction [1]. RelEx used dependency parse trees with a small number of simple rules for relation extraction [6]. A dependency-based neural network was applied to the relation classification [11]. Jointly embedded words and SDPs were used in aspect term extraction [30].

3.4 Natural Language Questions

The need for natural language question (NLQ) answering is emerging as people's demands for information on-line are rapidly increasing [23, 8]. There are two core steps of NLQ answering: 1) understanding NLQ and 2) converting NLQ into a database query language. Matching surface form relations to ontological relations plays an important role in both steps. A previously proposed methodology for translating NLQs into structured SPARQL queries over linked data maps natural language phrases to ontological relations based on a prepared dictionary [28]. *Querix*, a natural language interface for query translation, asks the user to map natural phrases to ontological relations through clarifying dialogs [9]. Unlike *Querix*, which constructs mappings with crowdsourcing, our proposed model can automatically build a dictionary between strings and ontological relations represented by dependency paths. Furthermore, we could broadened the coverage of acceptable NLQs by creating clusters of ambiguous ontological relations.

4 Experiments

We performed two experiments to illustrate the effectiveness of our model. For the first experiment, we compared the performance of two models in relation extraction of ontological relations. The first model is the baseline: Korean USSDP based on the combination of Korean DBpedia and Korean Wikipedia distant supervision data. For the extended model, we added NER results for each entity pair to the baseline. Then we compared mean average precision (MAP) and weighted mean average precision (WMAP) of the two models for evaluation. In the second experiment, we constructed clusters of ontological relations based on the trained matrix from the first experiment. Furthermore, we tested whether our model matches simple natural language questions to appropriate clusters.

4.1 Relation Extraction Evaluation

Dataset We built a Korean USSDP for the baseline model by using Korean DBpedia as the KB and Korean Wikipedia distant supervision data as raw text. Korean DBpedia is a branch of DBpedia based on the Korean Wikipedia [10]. The Korean Wikipedia distant supervision data comes from Korean Wikipedia documents chosen by Korean DBpedia triples. Specifically, we extracted sentences where two entities in the sentence have a relation according to Korean DBpedia. We rejected sentences with SDP of length longer than 3 because these SDPs are rarely observed in our dataset. For our target relation set, we selected 35 predefined ontological relations, each with more than 200 entity tuples in the distant supervision data. We then chose entity tuples with two or more SDPs in the distant supervision data. We finally obtained 12k entity tuples, 35 ontological relations and 348 frequent SDPs. The size of the baseline USSDP matrix was $[12360 \times 383]$.

We added NER results of each entity pair to the baseline USSDP matrix for the extended model. We used the following entity type categories: *AF* (artifacts: e.g., TV programs, books, movies, etc.), *DT* (date/time), *LC* (locations: e.g., countries, cities, towns, etc.), *OG* (organizations: e.g., governments, public corporations, companies, etc.), *PS* (persons), and *OT* (others). For each entity tuple, we added a new slot that indicates the entity type for each entity in the tuple (e.g., we placed 1 in the slot in the $\langle LC, LC \rangle$ column for the $\langle$*New York, United States*$\rangle$ row). Total 15 new columns were added to the baseline matrix, making the size of the extended matrix $[12360 \times 398]$.

Training Both models share the training and test sets. We built the training matrix for each ontological relation by filling the corresponding slots in the test set with zeros. The *nfe* model and stochastic gradient descent were used in the matrix factorization algorithm for each matrix. The embedding dimension was set to 5, which showed the highest AUC value in our parameter optimization step. The AUC measurement results are shown in Figure 3. Our batch size and the learning rate were set to 50 and 0.02, respectively.

Evaluation We performed a 10-fold cross-validation for evaluation. For evaluation, we measured the average precision (AP) for each ontological relation over all entity tuples in the test set. Then we measured MAP and WMAP for the performance of the entire model. MAP is simply the average of APs, and WMAP is the weighted version of MAP, where each AP is weighted by the number of entity tuples in each relation. According to previous results, MAP and WMAP are found to be robust and stable metrics for evaluating classification models by previous work [12].

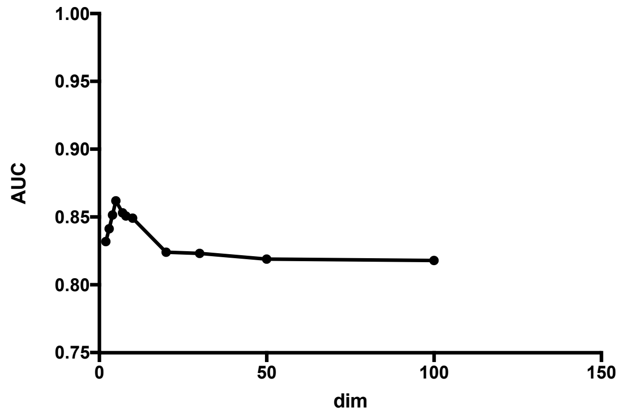

Fig. 3. AUC measurement

Results The experimental results for all ontological relations are shown in Table 1. The "#" column indicates the number of triples in the test set and their corresponding relation. The *"+NER"* column indicates the results from the extended USSDP with injected entity types. Winners for each relation are marked bold. The baseline USSDP shows performance of 0.62 (MAP) and 0.64 (WMAP), where the extended version of USSDP shows performance of 0.72 (MAP) and 0.75 (WMAP). The performance considerably increases after injecting NER results for the entity tuples. This is because the model can consider a significantly smaller set of relations for each entity tuple when the entity types are decided. For example, there are only 5 of 35 ontological relations that can have $\langle PS, PS \rangle$ as its entity types: *associatedMusicalArtist*, *predecessor*, *successor*, *parent*, and *spouse*. Even though we used a broad set of entity types due to the incomplete predefined types in Korean DBpedia, we expect better performance if we use specific entity type categories (e.g., subdividing *LC* into *LC-country*, *LC-city*, *LC-street*, *LC-mountain*, *LC-river*, etc.) because the model can rely on a smaller set of choices.

4.2 Ontological Relation Clusters

Building Clusters Based on the Korean USSDP from the first experiment, we divided the 35 target ontological relations into clusters. Our matrix factorization model contains the *neighborhood model* [19], which considers the correlation matrix among columns from the matrix. This correlation matrix is trained during the construction of our Korean USSDP. We form flat clusters from hierarchical clustering defined by the correlation matrix between the 35 ontological relations. Specifically, observations in each flat cluster have no greater co-phenetic distance than the threshold (0.7 in this experiment). Figure 4 shows the result. Each block in the figure corresponds to a cluster.

Some clusters such as [*club*, *team*] and [*isPartOf*, *region*] contain relations that can be confused even by humans. Other clusters such as [*birthPlace*, *deathPlace*, *nationality*, *country*] and [*activeYearsStartYear*, *activeYearsEndYear*] contain relations that

Relation	#	Baseline	+NER	Relation	#	Baseline	+NER
country	166	0.49	**0.91**	channel	25	0.57	**1.00**
team	73	0.81	**0.91**	currentMember	25	0.91	**0.94**
isPartOf	67	0.57	**0.83**	parent	24	**0.65**	0.51
birthPlace	63	**0.80**	0.73	city	22	0.37	**1.00**
deathPlace	51	**0.66**	0.51	owner	22	0.69	**0.76**
location	50	0.46	**0.76**	operator	20	0.75	**0.76**
associatedBand	47	**1.00**	0.93	activeYearsEndYear	20	0.79	**0.76**
associatedMusicalArtist	47	**1.00**	0.95	bandMember	19	**0.78**	0.70
club	46	**0.83**	0.68	spouse	18	0.33	**0.64**
nationality	39	**0.62**	0.33	artist	18	0.58	**0.79**
predecessor	33	**0.64**	0.53	capital	17	0.53	**0.84**
writer	33	**0.71**	0.56	managerClub	14	**0.32**	0.23
director	30	0.81	**0.87**	routeStart	14	0.60	**0.66**
producer	29	**0.60**	0.52	routeEnd	14	0.53	**0.66**
region	27	**0.26**	0.09	activeYearsStartYear	14	0.67	**0.88**
position	27	0.38	**0.94**	deathYear	12	0.58	**0.74**
successor	27	0.50	**0.66**	author	12	0.38	**0.81**
league	27	0.59	**0.95**				
MAP		0.62	**0.72**	**WMAP**		0.64	**0.75**

Table 1. AP and WMAP of of two USSDP models for 35 ontological relations

cannot be easily distinguished by relation extraction models. The fact that most clusters contain a reasonable set of relations indicates that our model shows decent relation clustering performance.

Predicting Simple Natural Language Questions We have created simple Korean NLQs that ask for triples containing the 35 ontological relations. Then we preprocessed each NLQ in a similar way we preprocessed Korean DS data. In each question, we selected the relational clause and the main topic word as target entities. For example, in the NLQ *"Who was born in Korea?"*, we selected *Who* and *Korea* as the target entities. Then we extracted the dependency path between the entities. Each relative pronoun was treated as the corresponding predefined entity type: e.g. *Who* as *PS*, *Where* as *LC*, and *When* as *DT*. We concatenated the preprocessed NLQs to the Korean USSDP matrix and then completed the whole matrix through matrix factorization.

Results Then we measured the prediction accuracy in two ways: 1) whether the prediction is exactly correct and 2) whether the prediction is in the same cluster with the right relation. Table 2 is the results. Since our model does not consider the lexical terms in the dependency path, it seems to be easily confused between similar relations and actually showed low accuracy for exact prediction. The accuracy increased to 0.69 when it comes to cluster prediction.

5 Conclusion

In this study we introduced the concept of USSDPs, which are a novel extension of universal schemas. USSDPs are built by using SDPs and entity tuple types with alternating surface form predicates. SDPs contain information regarding the syntactic structure of

birthPlace	associatedBand	club	activeYears StartYear	artist
deathPlace	associated MusicalArtist	team	activeYears EndYear	producer
nationality	parent	bandMember	director	league
country	predecessor	currentMember	writer	position
location	isPartOf	operator	capital	successor
routeEnd	region	owner	channel	author
routeStart	managerClub	deathYear	city	spouse

Fig. 4. Clusters of 35 ontological relations

a sentence, which makes SDPs a suitable alternative to surface form predicates. Entity tuple types considerably improve performance by considerably reducing the number of relation candidates in the model.

We constructed experiments in Korean and compared two different models to measure the performance of USSDPs: 1) the baseline USSDP in Korean built with Korean DBpedia and Korean Wikipedia distant supervision data 2) NER results injected into the baseline USSDP. The injection of entity types increased the performance of WMAP from 0.64 to 0.75. We also constructed clusters of target ontological relations based on the completed USSDP matrix in the first experiment, which showed a reasonable grouping of similar and confusing relations.

Our model has shown considerable performance in predicting the relation cluster of NLQs, but we still need to improve exact relation predictions. To distinguish the exact relation from clusters of similar relations can be improved by using fine-grained features. Refining predefined entity types or considering important lexical strings on the dependency path are the examples of fine-grained features.

Despite the verification of our model in Korean, additional multilingual experiments are required to fully verify the applicability of USSDPs in all languages. Further work in this area will provide converging evidence which will advance our idea that SDPs are an ample alternative for surface form predicates in building universal schemas, which makes them globally applicable. We believe our study and conclusions offer deep intuitive understanding of multilingual relation extraction. Rather than entangling models to language-specific features transferring to globally consistent features, our transfor-

	Answer	Exact prediction	Cluster prediction
[Who] was born in [Korea]?	birthPlace	birthPlace	[birthPlace, country, deathPlace, nationality]
[Where] does the [Korean Train Express] ends?	routeEnd	routeStart	[location, routeEnd, routeStart]
[When] did [PSY] started singing?	activeYearsStartYear	activeYearsEndYear	[activeYearsEndYear, activeYearsStartYear]
[Which] team does [Ji-Sung Park] play?	team	club	[club, team]
[Where] is [Incheon National Airport]?	location	location	[location, routeEnd, routeStart]

(a)

	Accuracy
Exact prediction	0.29
Cluster prediction	0.69

(b)

Table 2. (a) Examples of successful cluster prediction results (b) Accuracy measurement results of exact prediction and cluster prediction

mation of surface form predicates into UD may offer much greater potential in future development and applicability.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (2013-0-00109, WiseKB: Big data based self-evolving knowledge base and reasoning platform)

References

1. Bunescu, R., Mooney, R., Ramani, A., Marcotte, E.: Integrating co-occurrence statistics with information extraction for robust retrieval of protein interactions from medline. In: Proceedings of the workshop on linking natural language processing and biology: towards deeper biological literature analysis. pp. 49–56. Association for Computational Linguistics (2006)
2. De Marneffe, M.C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., Manning, C.D.: Universal stanford dependencies: A cross-linguistic typology. In: LREC. vol. 14, pp. 4585–4592 (2014)
3. De Marneffe, M.C., MacCartney, B., Manning, C.D., et al.: Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC. vol. 6, pp. 449–454. Genoa Italy (2006)
4. De Marneffe, M.C., Manning, C.D.: The stanford typed dependencies representation. In: Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation. pp. 1–8. Association for Computational Linguistics (2008)
5. Etzioni, O., Banko, M., Soderland, S., Weld, D.S.: Open information extraction from the web. Communications of the ACM **51**(12), 68–74 (2008)
6. Fundel, K., Küffner, R., Zimmer, R.: Relexrelation extraction using dependency parse trees. Bioinformatics **23**(3), 365–371 (2006)
7. Gardner, M., Talukdar, P.P., Krishnamurthy, J., Mitchell, T.: Incorporating vector space similarity in random walk inference over knowledge bases. Proceedings of the 2014 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (2014)

8. Hirschman, L., Gaizauskas, R.: Natural language question answering: the view from here. natural language engineering **7**(4), 275–300 (2001)
9. Kaufmann, E., Bernstein, A., Zumstein, R.: Querix: A natural language interface to query ontologies based on clarification dialogs. In: 5th International Semantic Web Conference (ISWC 2006). pp. 980–981. Springer (2006)
10. Kim, E.k., Weidl, M., Choi, K.S., Auer, S.: Towards a korean dbpedia and an approach for complementing the korean wikipedia based on dbpedia. OKCon **575**, 12–21 (2010)
11. Liu, Y., Wei, F., Li, S., Ji, H., Zhou, M., Wang, H.: A dependency-based neural network for relation classification. arXiv preprint arXiv:1507.04646 (2015)
12. Manning, C.D., Raghavan, P., Schütze, H., et al.: Introduction to information retrieval, vol. 1. Cambridge university press Cambridge (2008)
13. McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., et al.: Universal dependency annotation for multilingual parsing. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). vol. 2, pp. 92–97 (2013)
14. Neelakantan, A., Roth, B., Mc-Callum, A.: Compositional vector space models for knowledge base inference. In: 2015 aaai spring symposium series (2015)
15. Nivre, J.: Towards a universal grammar for natural language processing. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 3–16. Springer (2015)
16. Nivre, J., de Marneffe, M.C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C.D., McDonald, R.T., Petrov, S., Pyysalo, S., Silveira, N., et al.: Universal dependencies v1: A multilingual treebank collection. In: LREC (2016)
17. Petrov, S., Das, D., McDonald, R.: A universal part-of-speech tagset. arXiv preprint arXiv:1104.2086 (2011)
18. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. pp. 452–461. AUAI Press (2009)
19. Riedel, S., Yao, L., McCallum, A., Marlin, B.M.: Relation extraction with matrix factorization and universal schemas. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 74–84 (2013)
20. Rocktäschel, T., Singh, S., Riedel, S.: Injecting logical background knowledge into embeddings for relation extraction. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1119–1129 (2015)
21. Roth, B., Barth, T., Wiegand, M., Klakow, D.: A survey of noise reduction methods for distant supervision. In: Proceedings of the 2013 workshop on Automated knowledge base construction. pp. 73–78. ACM (2013)
22. Schmitz, M., Bart, R., Soderland, S., Etzioni, O., et al.: Open language learning for information extraction. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 523–534. Association for Computational Linguistics (2012)
23. Simmons, R.F.: Natural language question-answering systems: 1969. Communications of the ACM **13**(1), 15–30 (1970)
24. Singh, S., Rocktäschel, T., Riedel, S.: Towards combined matrix and tensor factorization for universal schema relation extraction. In: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing. pp. 135–142 (2015)
25. Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., Gamon, M.: Representing text for joint embedding of text and knowledge bases. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1499–1509 (2015)

26. Verga, P., Belanger, D., Strubell, E., Roth, B., McCallum, A.: Multilingual relation extraction using compositional universal schema. arXiv preprint arXiv:1511.06396 (2015)
27. Verga, P., Neelakantan, A., McCallum, A.: Generalizing to unseen entities and entity pairs with row-less universal schema. arXiv preprint arXiv:1606.05804 (2016)
28. Yahya, M., Berberich, K., Elbassuoni, S., Ramanath, M., Tresp, V., Weikum, G.: Natural language questions for the web of data. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 379–390. Association for Computational Linguistics (2012)
29. Yao, L., Riedel, S., McCallum, A.: Universal schema for entity type prediction. In: Proceedings of the 2013 workshop on Automated knowledge base construction. pp. 79–84. ACM (2013)
30. Yin, Y., Wei, F., Dong, L., Xu, K., Zhang, M., Zhou, M.: Unsupervised word and dependency path embeddings for aspect term extraction. arXiv preprint arXiv:1605.07843 (2016)
31. Zeman, D.: Reusable tagset conversion using tagset drivers. In: LREC. vol. 2008, pp. 28–30 (2008)