Inferring Binary Relation Schemas for Open Information Extraction

Kangqi Luo[1] and **Xusheng Luo**[2] and **Kenny Q. Zhu**[3]
Department of Computer Science & Engineering
Shanghai Jiao Tong University, Shanghai, China
[1]luokangqi@sjtu.edu.cn [2]freefish_6174@sjtu.edu.cn [3]kzhu@cs.sjtu.edu.cn

Abstract

This paper presents a framework to model the semantic representation of binary relations produced by open information extraction systems. For each binary relation, we infer a set of preferred types on the two arguments simultaneously, and generate a ranked list of type pairs which we call schemas. All inferred types are drawn from the Freebase type taxonomy, which are human readable. Our system collects 171,168 binary relations from ReVerb, and is able to produce top-ranking relation schemas with a mean reciprocal rank of 0.337.

1 Introduction

Open information extraction (or Open IE) is a task of extracting all sorts of relations between named entities or concepts from open-domain text corpora, without restraining itself to specific relations or patterns. State-of-the-art Open IE systems (Carlson et al., 2010; Fader et al., 2011; Schmitz et al., 2012; Nakashole et al., 2012) extract millions of binary relations with high precision from the web corpus. Each extracted relation instance is a triple of the form $\langle arg_1,\ rel,\ arg_2\rangle$, where the relation rel is a lexical or syntactic pattern, and both arguments are multi-word expressions representing the argument entities or concepts.

Whereas Open IE provides concrete relation instances, we are interested in generalizing these instances into more abstract semantic representations. In this paper, we focus on inferring the schemas of binary relations.

For example, given the binary relation "play in", an Open IE system extracts many triples of the form $\langle X,\ play\ in,\ Y\rangle$. The following relation triples are extracted in ReVerb:

$$\langle \text{Goel Grey}, played\ in, \text{Cabaret}\rangle$$
$$\langle \text{Tom Brady}, play\ in, \text{National Football League}\rangle$$

Informally, the goal of our system is to automatically infer a set of schemas such as $\langle t_1,\ play\ in,\ t_2\rangle$, where t_1 and t_2 are two semantic types drawn from a standard knowledge base such as WordNet (Miller, 1995), Yago (Suchanek et al., 2007), Freebase (Bollacker et al., 2008), and Probase (Wu et al., 2012), and each such schema can be used to represent a set of "play in" relation instances. For the above example, two possible schemas for "play in" are:

$$\langle \textbf{film actor},\ play\ in,\ \textbf{film}\rangle$$
$$\langle \textbf{athlete},\ play\ in,\ \textbf{sports league}\rangle$$

The schema of a binary relation is useful information in NLP tasks, such as context-oriented entity recognition and open domain question answering. Suppose we are to recognize the entities in the sentence "*Granger* played in *the NBA*". "*Granger*" is a highly ambiguous term, while "*the NBA*" is probably a sports league. Then with the the above relation schemas for "play in", the entity recognizer knows that "*Granger*" is more likely to be an athlete, which results in the correct linking to "Danny Granger", who is an NBA player, even though the Open IE has never extracted such fact before.

One relevant technique to achieve our goal is *selectional preference* (SP) (Resnik, 1996; Erk, 2007; Ritter et al., 2010), which computes the most appropriate types for a specific argument of a predicate. SP is based on the idea of mutual information (Erk, 2007), which tends to select types which are *unique* to the relation. In other words, common types which can be used for many different relations are less preferred. However, in Open IE, many relations are related or even similar, e.g., *play in*, *take part in* and *be involved in*. There's no reason for these relations not to share schemas. Therefore in this paper, our problem is, given a re-

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 555–560,
Lisbon, Portugal, 17-21 September 2015.

lation and its instances, identify the smallest types that can cover as many instances as possible. Our approach first attempts to link the arguments in the relation instances to a set of possible entities in a knowledge base, hence generate a set of $\langle e_1, e_2 \rangle$ entity pairs. Then we select a pair of types $\langle t_1, t_2 \rangle$ that covers maximum number of entity pairs. We resolve ties by selecting the smaller (more specific) types according to a type taxonomy inferred from knowledge base.

This paper makes the following contributions: i) we defined the schema inference problem for binary relations from Open IE; ii) we developed a prototype system based on Freebase and entity linking (Lin et al., 2012; Ratinov et al., 2011; Hoffart et al., 2011; Rao et al., 2013; Cai et al., 2013), which simultaneously models the type distributions of two arguments for each binary relation; iii) our experiment on ReVerb triples showed that the top inferred schemas receive decent mean reciprocal rank (MRR) of 0.337, with respect to the human labeled ground truth.

2 Problem Definition

A knowledge base K is a 5-tuple $\langle E, Alist, T, P, IsA \rangle$, where:

- E is a finite set of entities $e \in E$,
- $Alist(e) = \{n_1, n_2, ...\}$ is a function which returns a set of names (or aliases) of an entity,
- T is a finite set of types $t \in T$,
- P is a finite set of relation instances $p(e_1, e_2)$, where p is a predicate in K.
- IsA is a finite set entity-type pairs (e, t), representing the isA relation between entities and types. An entity belongs to at least one type.

An Open IE triple set S contains all relation instances extracted by the IE system, of the form $\langle a_1,\ rel,\ a_2 \rangle$, where a_1 and a_2 are the arguments of extracted relation pattern rel. The set of argument pairs sharing the same relation pattern rel is denoted by S_{rel}.

The problem is, for each S_{rel}, return a set of type pairs (or schemas) from T, $\langle t_1, t_2 \rangle$, ordered by the number of argument pairs covered in S_{rel}. If two schemas cover the same number of argument pairs from S_{rel}, the schema covering smallest number of entities wins.

3 System

The workflow of our system is shown in Figure Figure 1. The system takes Open IE relation tuples as the input, then performs entity linking, relation grouping and schema ranking to translate them into final ranked list of schemas.

(1) Entity Linking: Relation arguments are linked to entities in the knowledge base by fuzzy string matching. Each entity in the knowledge base has a unique identifier.

(2) Relation Grouping: Linked tuples sharing similar relation patterns are grouped together. Besides, each group has a representative relation pattern, which is generated from all the patterns within the group.

(3) Schema Ranking: For each linked tuple in one relation group, argument entities are transformed into types drawn from the knowledge base. Then this procedure ranks type pairs (schemas) in terms of how much Open IE tuples a type pair can cover and how specific a type concept is.

3.1 Entity Linking

In the entity linking step, by matching arguments to entities in the knowledge base, each relation tuple is transformed into linked tuples, $ltup = \langle e_1,\ rel,\ e_2 \rangle$, with linking scores.

We aim to support fuzzy matching between arguments and entity aliases, so we take all the aliases into consideration, and build an inverted index from words to aliases. Different words in one alias cannot be treated equally. Intuitively, a word is more important if it occurs in fewer aliases (n), and vice versa. Based on the inverted index, we use inverted document frequency score to approximately model the weight of a word w:

$$idf(w) = 1 \,/\, \log(|\{n : w \in n\}|) \qquad (1)$$

Besides, stop words are removed from aliases, treating their idf scores as 0. In order to measure the probability of fuzzy matching from an argument (a) to an alias (n), we introduce the weighted overlap score:

$$overlap(a, n) = \frac{\sum_{w \in a \cap n} idf(w)}{\sum_{w \in a \cup n} idf(w)} \qquad (2)$$

We merge all the aliases of an entity together to producing a similarity score of fuzzy matching between an entity and an argument:

$$sim(e, a) = \max_{n \in Alist(e)} overlap(a, n) \qquad (3)$$

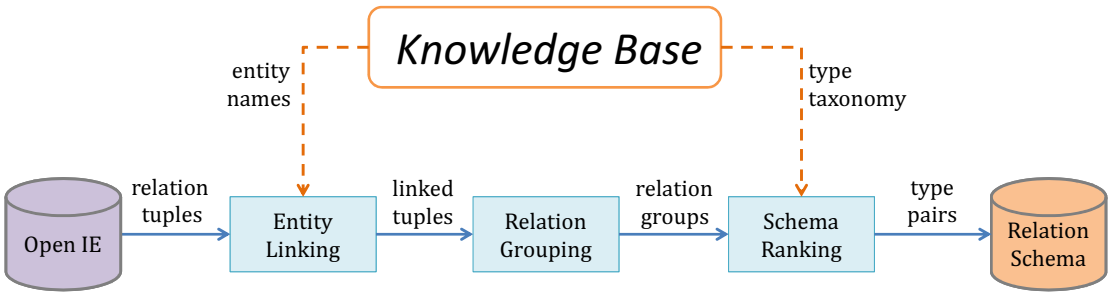

Figure 1: System Architecture

In order to control the quality of candidate entities, for an argument having m words (with stop words removed), we only keep entities that have at least one alias matching $m-1$ words in the argument, and have a similarity score larger than a threshold, τ. With similarity score computed, we generate 10 best entity candidates respectively for both the subject and the object of rel.

Next, we model the joint similarity score (F) of the relation tuple $\langle a_1, rel, a_2 \rangle$ with each entity pair combination $\langle e_1, e_2 \rangle$ in two ways. One is a **naive method** which only considers the similarity between arguments and corresponding entities:

$$F(a_1, e_1, a_2, e_2, rel) = sim(e_1, a_1) \times sim(e_2, a_2). \quad (4)$$

The other method takes predicate paths between e_1 and e_2 into consideration. Let $\vec{w}$ be the word vector of rel, and $\vec{p}$ be a path of predicates connecting e_1 and e_2 in at most 2 hops. Here we say two entities e_1 and e_2 are connected in 1 hop, if there exists a predicate p, such that $p(e_1, e_2)$ (or $p(e_2, e_1)$) is in the knowledge base.

Similarly, e_1 and e_2 are connected in 2 hops, if there exists two predicates p_1, p_2 and a transition entity e', such that $p_1(e_1, e')$ (or $p_1(e', e_1)$) and $p_2(e', e_2)$ (or $p_2(e_2, e')$) are in the knowledge base.

We hence define the relatedness between $\vec{p}$ and $\vec{w}$ in the form of a conditional probability according to the Naive Bayes model:

$$P(\vec{p}\,|\,\vec{w}) \approx \prod_p P(p\,|\,\vec{w}) \propto \prod_p P(p) \prod_w P(w\,|\,p), \quad (5)$$

and we follow the IBM alignment Model 1 (Yao and Van Durme, 2014) to calculate the conditional probability between predicates and relation words $P(\vec{p}\,|\,\vec{w})$. Based on the information above, we define a richer joint similarity score, considering all valid paths between e_1 and e_2:

$$F(a_1, e_1, a_2, e_2, rel) = sim(e_1, a_1) \times sim(e_2, a_2) \times \sum_{\vec{p}} P(\vec{p}|\vec{w}). \quad (6)$$

Due to the multiplications, the value of $P(\vec{p}|\vec{w})$ varies a lot among different entity pair candidates. The large deviation makes $P(\vec{p}|\vec{w})$ the most important term in Eq. (6), especially in the case when none of predicate paths are similar enough to the relation words. Therefore, we trust the factor of $P(\vec{p}|\vec{w})$ only when there exists a similar predicate path. In practice, we use a threshold ρ to control whether to use Eq. (6) or Eq. (4). We call this an **ensemble method**. For each case of entity linking, if there exists one candidate entity pair satisfying $P(\vec{p}\,|\,\vec{w}) > \rho$, we use the ensemble method, otherwise we fall back to the naive method for the current case.

3.2 Relation Grouping

In the step of relation grouping, linked tuples with similar relation patterns form a group. Each linked tuple belongs to one unique group.

The idea is to simplify relation patterns by syntactic transformations. If two patterns share the same simplified pattern, we treat them as being equivalent and put them into one group. First, since adjectives, adverbs and modal verbs can hardly change the type distribution of arguments in a relation, we remove these words from a pattern. Second, many relations from Open IE contain verbs, which come in different tenses. We transform all tenses into present tense. In addition, passive voice in a pattern, if any, is kept in the transformed pattern. A simple example below shows a group of relations:

⟨X, *resign from*, Y⟩
⟨X, *had resigned from*, Y⟩
⟨X, *finally resigned from*, Y⟩

All linked tuples with the same simplified pattern form a group. This pattern is selected as the representative pattern, like the pattern "*resign from*" in the above example.

3.3 Schema Ranking

Given a relation group, the step of schema ranking produces a ranked list of relation schemas with two constraints. Take "play in" as an example, the ideal schemas will contain the pair $\langle actor,\ film\rangle$ and $\langle athlete,\ sports\ league\rangle$

Each linked tuple $\langle e_1,\ rel,\ e_2\rangle$ supports the type pair $\langle t_1,\ t_2\rangle$ where $(e_1, t_1),\ (e_2, t_2) \in IsA$ in the knowledge base. We treat these pairs equally, since it's not trivial to tell which type is more related to the argument given the relation tuple as context. Combining all tuples in one group, we define the support of a type pair tp in a group (using the representative pattern r to stand for the group):

$$sup_r(\langle t_1, t_2\rangle) = \{(e_1, t_1) \in IsA,\ (e_2, t_2) \in IsA\} \quad (7)$$

A simple intuition is to rank schemas by the size of the support. Since one entity belongs to multiple types, relation schemas with general types will be ranked higher. However, two different schemas may share the same support. For instance, given the relation "*X die in Y*", suppose Open IE extractions and entity linking step returns correct results, the schema $\langle person,\ location\rangle$ and $\langle deceased\ person,\ location\rangle$ have identical supports. The latter one shows a more concrete representation of the relation, because *deceased person* covers small entities than *person* in the knowledge base.

Therefore, the schemas cannot be ranked by using the support alone. Next, we aim to extract the subsumption relations between types in the knowledge base, building the taxonomy of types.

We first define all entities in t as

$$cover(t) = \{e \mid (e, t) \in IsA\}. \quad (8)$$

Intuitively, type t_1 is subsumed in t_2, if all entities in t_1 also belong to t_2, that is, $cover(t_1) \subseteq cover(t_2)$. This uses the idea of **strict set inclusion**. For example, we can learn that the type *person* subsumes types such as *actor, politician* and *deceased person*.

However, strict set inclusion doesn't always hold in the knowledge base. For example, entities in type *award winner* are mostly *person*, but there still has some organizations in it. The strict method fails to find the subsumption relation between *award winner* and *person*, while this subsumption actually holds with a large confidence.

To resolve this problem, we use a **relaxed set inclusion**, where the set $cover(t_1)$ can be a subset of another set $cover(t_2)$ to a certain degree. We define the degree of the subsumption as the ratio between the number of entities in the two sets:

$$deg(t_1 \subseteq t_2) = \frac{|cover(t_1) \cap cover(t_2)|}{|cover(t_1)|}. \quad (9)$$

If $deg(t_1 \subseteq t_2) > \epsilon$, then t_1 is subsumed by t_2, and ϵ is a confidence parameter determined by weight tuning. By scanning all types in the knowledge base, all subsumption relations with enough confidence are extracted, forming our type taxonomy.

With a type hierarchy computed by above relaxed set inclusion, we can define a schema $\langle t_1,\ t_2\rangle$ subsumes another schema $\langle t_3,\ t_4\rangle$ if i) t_1 subsumes t_3 and t_2 subsumes t_4; ii) t_1 subsumes t_3 and $t_2 = t_4$; or iii) t_2 subsumes t_4 and $t_1 = t_3$.

If a schema (type pair) tp_1 subsumes another schema tp_2, and their supports ($|sup_r(tp)|$) are approximately equal, we give the more specific schema tp_2 a higher rank in the output list. Here two supports are roughly equal if:

$$\frac{|sup_r(tp_1)| - |sup_r(tp_2)|}{max(|sup_r(tp_1)|,\ |sup_r(tp_2)|)} < \lambda \quad (10)$$

Where λ is a threshold determined in the experiments.

4 Evaluation

Freebase (Bollacker et al., 2008) is a collaboratively generated knowledge base, which contains more than 40 million entities, and more than 1,700 real types [1]. In our experiment, We use the 16 Feb. 2014 dump of Freebase as the knowledge base.

ReVerb (Fader et al., 2011) is an Open IE system which aims to extract verb based relation instances from web corpora. The release ReVerb dataset contains more than 14 millions of relation tuples with high quality. We observed that in ReVerb, some argument is unlikely to be an entity in Freebase, for example:

$\langle Metro\ Manila,\ consists\ of,$ **12 cities**$\rangle$,

[1] Freebase types are identified by type id, for example, $sports.pro\_athlete$ stands for "professional athlete".

where the object argument is not an entity but a type. Since types are usually represented by lowercase common words, we remove the tuple if one argument is lowercase, or if it is made up completely of common words in WordNet. In addition, because date/time such as "Jan. 16th, 1981" often occurs in the object argument while Freebase does not have any such specific dates as entities, we use SUTime (Chang and Manning, 2012) to recognize dates as an virtual entity. After cleaning, the system collects 3,234,208 tuples and 171,168 relation groups.

The following parameters are tuned using a development set: $\tau = 0.667$, $\epsilon = 0.6$, $\lambda = 5\%$ and $\rho = e^{-50}$. For relation grouping, we use Stanford Parser (Klein and Manning, 2003) to perform POS tagging, lemmatizing and parsing on relations.

We first evaluate the results of entity linking. We randomly pick 200 relation instances from ReVerb, and manually labeled arguments with Freebase entities. For both naive and ensemble strategy, we evaluate the precision, recall, F1 and MRR score on the labeled set. An output entity pair is correct, if and only if both arguments are correctly linked. Experimental results are listed in Table 1.

Table 1: Entity Linking Result

Strategy	P	R	F1	MRR
Naive	0.371	0.327	0.348	0.377
Ensemble	0.386	0.340	0.361	0.381

For the evaluation of relation schema, we first randomly pick 50 binary relations with support larger than 500 from the system. For each relation, we selected top 100 type pairs with the largest support, as what we evaluated. We assigned 3 human annotators to label the fitness score of type pair for the relation. The labeled score ranges from 0 to 3. Then we merge these 3 label sets, forming 50 gold standard rankings. When evaluating a relation schema list from our system, we calculate the MRR score (Liu, 2009) by the top schemas in the gold rankings.

For comparison, we use Pointwise Mutual Information (Church and Hanks, 1990) as our baseline model, which is used in other selectional preference tasks (Resnik, 1996). We define the association score between relation and type pair as:

$$PMI(r, tp) = p(r, tp) \log \frac{p(r, tp)}{p(r, *)p(*, tp)} \quad (11)$$

Where $p(r, tp)$ is the joint probability of relation and type pair in the whole linked tuple set, and $*$ stands for any relations or type pairs.

Table 2 shows the MRR scores by using both baseline model (PMI) and our approach. As the result shows, our approach improves the MRR score by 10.1%.

Table 2: End-to-end Schema Inference Results

Approach	MRR Score
PMI Baseline	0.306
Our Approach	0.337

Finally, Table 3 shows some example binary relations, and their schemas inferred by our system. We can see that with a well-defined type hierarchy, our system is able to extract both coarse-grained and fine-grained type information from entities, resulting in a informative type lists.

Table 3: Sample Relation Schemas

Relation	Top Schemas
be find at	$\langle location,\ location\rangle$ $\langle employer,\ location\rangle$ $\langle organization,\ location\rangle$
appear on	$\langle person,\ tv\ program\rangle$ $\langle person,\ nominated\ work\rangle$ $\langle person,\ winning\ work\rangle$
be the writer of	$\langle person,\ nominated\ work\rangle$ $\langle person,\ film\rangle$ $\langle person,\ book\ subject\rangle$

5 Conclusion

In summary, our work describes a data driven approach of relation schema inference. By maximizing the support of both arguments simultaneously, our system is able to generate human-readable type pairs for a binary relation from Open IE systems. Our experiments shows that the top ranked relation schemas for each relation are accurate according to human judges. The proposed framework can be integrated with future Open IE systems.

Acknowledgement

Kenny Q. Zhu is the contact author and was supported by NSFC grant 61373031 and NSFC-NRF Joint Research Program No. 61411140247.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring

human knowledge. In *ACM SIGMOD*, pages 1247–1250. ACM.

Zhiyuan Cai, Kaiqi Zhao, Kenny Q. Zhu, and Haixun Wang. 2013. Wikification via link co-occurrence. In *ACM CIKM'13*, pages 1087–1096.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, page 3.

Angel X Chang and Christopher D Manning. 2012. Sutime: A library for recognizing and normalizing time expressions. In *LREC*, pages 3735–3740.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Katrin Erk. 2007. A simple, similarity-based model for selectional preferences. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, page 216.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *EMNLP*, pages 1535–1545.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of EMNLP*, pages 782–792.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *ACL*, pages 423–430.

Thomas Lin, Oren Etzioni, et al. 2012. Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 84–88.

Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. Patty: a taxonomy of relational patterns with semantic types. In *EMNLP*, pages 1135–1145.

Delip Rao, Paul McNamee, and Mark Dredze. 2013. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, multilingual information extraction and summarization*, pages 93–115. Springer.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of ACL:HLT*, pages 1375–1384.

Philip Resnik. 1996. Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61(1):127–159.

Alan Ritter, Oren Etzioni, et al. 2010. A latent dirichlet allocation method for selectional preferences. In *ACL*, pages 424–434.

Michael Schmitz, Robert Bart, Stephen Soderland, Oren Etzioni, et al. 2012. Open language learning for information extraction. In *EMNLP*, pages 523–534.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *16th international World Wide Web conference (WWW 2007)*, New York, NY, USA. ACM Press.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. 2012. Probase: a probabilistic taxonomy for text understanding. In *SIGMOD Conference*, pages 481–492.

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of ACL*.