Open Relation Extraction and Grounding

Dian Yu, Lifu Huang, Heng Ji
Computer Science Department
Rensselaer Polytechnic Institute
{yud2, huangl7, jih}@rpi.edu

Abstract

Previous open Relation Extraction (open RE) approaches mainly rely on linguistic patterns and constraints to extract important relational triples from large-scale corpora. However, they lack of abilities to cover diverse relation expressions or measure the relative importance of candidate triples within a sentence. It is also challenging to name the relation type of a relational triple merely based on context words, which could limit the usefulness of open RE in downstream applications. We propose a novel importance-based open RE approach by exploiting the global structure of a dependency tree to extract salient triples. We design an unsupervised method to name relation types by grounding relational triples to a large-scale Knowledge Base (KB) schema, leveraging KB triples and weighted context words associated with relational triples. Experiments on the English Slot Filling 2013 dataset demonstrate that our approach achieves 8.1% higher F-score over state-of-the-art open RE methods.

1 Introduction

Open Relation Extraction (open RE) (Banko and Etzioni, 2008) aims at extracting relational triples from an open-domain corpus. Each triple contains two arguments and a phrase which denotes the relation between them. In this paper, we focus on discovering relations between *entities*.

Most successful open RE approaches (Fader et al., 2011; Xu et al., 2013; Bovi et al., 2015; Bhutani et al., 2016) extract salient relational triples based on lexical or syntactic patterns. However, such handcrafted or automatically learned patterns are incapable of covering diverse relation expressions (Soderland et al., 2013). Subsequently, the shortest path between arguments derived from a dependency tree has been widely applied to generate patterns to capture long-distance and complex relations. However, additional heuristic rules are usually needed to filter out the resulting large number of meaningless patterns (Wu and Weld, 2010; Mausam et al., 2012; Bovi et al., 2015). Besides, such flat syntactic structures lack the ability to measure the relative importance of candidate triples in a sentence. For example, the sentence in $E1$ places particular emphasis on the relation between *"Lucille Clifton"* and *"1936"* which therefore should be retained.

E1 *"Lucille Clifton, whom he married in 1958, was born in 1936."*

We notice that a candidate relational triple is likely to be salient if its two arguments are strongly connected in a dependency tree. Instead of relying on patterns to capture important triples, we use an importance-based strategy by exploring the entire dependency tree structure to automatically measure the connection strength of candidate argument pairs. Specifically, we assume that a relational triple is important if there is a relatively short random walk-based distance between two relatively important arguments, measured against the entire dependency tree of a given sentence. For each argument pair, we apply an effective random-walk based method to assign weights to context words in the sentence (Section 2).

How to assign a meaningful relation type name to a relational triple is also a primary challenge for open RE. Previous methods use relevant context words in the associated sentence as relation phrases (type names) (Del Corro and Gemulla, 2013; Bhutani et al., 2016). However, there is still no generally accepted guideline for relation phrase extraction. Multiple relation phrases can corre-

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 854–864,
Taipei, Taiwan, November 27 – December 1, 2017 ©2017 AFNLP

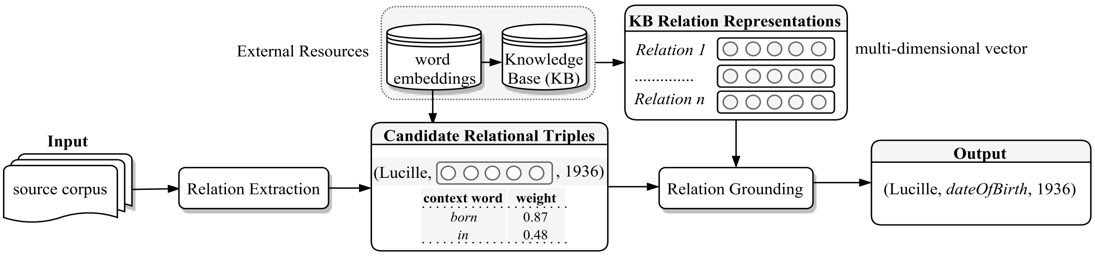

Figure 1: Framework overview.

spond to the same relation type. Besides, overly-specific or implicit relation phrases are incapable of providing adequate information for downstream applications. For example, the relation between *"Patricia"* and *"Gary Cooper"* cannot be clearly expressed by a set of words in the following sentence $E2$. Therefore, previous studies heavily rely on resources such as patterns (Soderland et al., 2013), training data (Weston et al., 2013), or distantly-labeled corpora (Angeli et al., 2015b) to map open RE triples to a known relation schema.

E2 *"Patricia later described her relation with Gary Cooper as one of the most beautiful things that ever happed to her in her life."*

Compared with a small number of predefined relation types such as those defined in Automatic Content Extraction (ACE) [1], the relation schema in a large-scale Knowledge Base (KB) such as DBpedia (Auer et al., 2007) covers a much wider range of informative relations along with their type signatures. Considering the open-domain nature shared by open RE and a large-scale KB, we propose an unsupervised grounding method to name the relation type between two arguments as either a KB relation or NONE, by leveraging KB triples and weighted context information associated with each argument pair based on pre-trained word embeddings (Section 3). Compared with previous methods (*e.g.*, (Riedel et al., 2013; Weston et al., 2013)), we regard intra-sentence context words as intermediate results for the subsequent grounding process, and we do not require any aligned training corpora or relation phrases for KB triples. The proposed framework is illustrated in Figure 1.

To the best of our knowledge, this is the first open RE method which exploits the global structure of a dependency tree to extract salient relational triples. This is also the first unsupervised relation grounding method to name relation types for open RE based on KB triples and intra-sentence context information. Experiments on the English Slot Filling (SF) (Ji et al., 2010, 2011) 2013 dataset demonstrate that our approach outperforms state-of-the-art open RE approaches.

2 Relation Extraction

In this section, we introduce a graph-based method to extract argument pairs of salient relational triples. We first present the extended dependency tree construction for each sentence (Section 2.1). Then we show the computation of the relation strength between two arguments (Section 2.4) considering both their random-walk based distance (Section 2.2) and the relative importance of each argument in the tree (Section 2.3).

2.1 Extended Dependency Tree Construction

Given a sentence containing N words, we construct a weighted directed graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \dots, v_N\}$ represents words, and $\mathcal{E}$ is a directed edge set, associated with each directed edge $v_i \rightarrow v_j$ representing a dependency relation originating from v_i to v_j. We assign a weight $w_{ij} = 1$ to $v_i \rightarrow v_j$ and add its reverse edge $v_j \rightarrow v_i$ with $w_{ji} = 0.5$. By adding lower-weighted reverse edges, we can analyze the relation between two nodes which are not connected by directed dependency links while maintaining our preferences toward the original directions.

We first apply a dependency parser to generate basic uncollapsed dependencies.[2] We annotate an entity or time mention node with its type. For example in $E1$, "*Lucille Clifton*" is annotated as a person, and "*1936*" is annotated as a date. Finally we perform coreference resolution which introduces coreference links between nodes that refer to the same entity within a document. We replace any nominal or pronominal entity mention with its coreferential name mention. For example, "*he*" is

[1] www.ldc.upenn.edu/collaborations/past-projects/ace.

[2] All the tools we used are introduced in Section 4.

replaced by "*Fred James Clifton*". Formally, an extended dependency tree is an annotated tree of entity mentions and their links. By adding the reverse edges, we generate the final extended dependency tree in Figure 2. We regard any two entities as a candidate argument pair. $E1$ contains 4 entities and therefore we can extract $\binom{4}{2} = 6$ argument pairs (*e.g.*, (*"Lucille Clifton"*, *"1936"*)).

E1: Lucille Clifton, whom he married in 1958, was born in 1936.

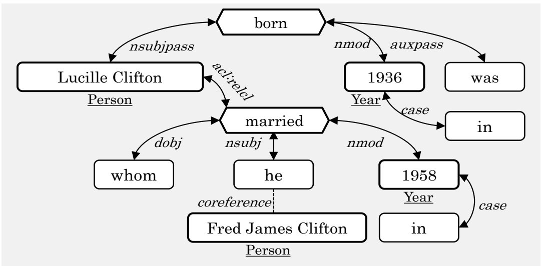

Figure 2: Extended dependency tree of E1.

2.2 Distance Computation

As mentioned previously, a shorter distance between two strongly connected nodes is more likely to indicate the existence of an important relation. We compute the distance between two nodes based on a Markov-chain model of random walk. We define a random walk through G by assigning a transition probability to each directed edge. Thus, a random walker can jump from node v_i to v_j and represent a state of the Markov chain. For a node v_i, we denote $\mathcal{N}(i)$ as the set of its neighbors. The probability of transitioning from node v_j to node v_i is defined as $p_{ji} = w_{ji}/\sum_{k \in \mathcal{N}(j)} w_{jk}$ for nodes v_i that have an edge from v_j to v_i, and 0 otherwise. We define the transition probability matrix of the Markov chain associated with random walks on G as $\boldsymbol{P}$.

The ***mean first-passage time*** m_{ji} (Aldous and Fill, 2002) is the average number of steps needed by a random walker for reaching state i for the first time, when starting from state j. We call $c_{ij} = m_{ij} + m_{ji}$ as the ***average commute time*** (Lovász, 1993). The fact that c_{ij} can be regarded as a distance in G between nodes v_i and v_j is proven by Klein and Randić (1993). Compared with the shortest path between v_i and v_j, the value of c_{ij} will decrease when the number of paths connecting v_i and v_j increases and when the length of any path decreases (Fouss et al., 2007).

The fundamental matrix $\boldsymbol{Z}$ plays an essential role in computing various quantities related to random walks. For a weighted and directed graph, Li and Zhang (2010) demonstrate that $\boldsymbol{Z}$ can be computed directly using the following equation:

$$\boldsymbol{Z} = (\boldsymbol{I} - \boldsymbol{P} + \boldsymbol{ED})^{-1} - \boldsymbol{ED} \qquad (1)$$

where $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{E}$ is a matrix containing all 1s, and $\boldsymbol{D}$ is the diagonal matrix with elements $d_{kk} = \pi(k)$ where $\pi(k)$ is the stationary distribution of node v_k in the Markov chain.

We can directly compute a mean first-passage $|\mathcal{V}| \times |\mathcal{V}|$ matrix and a symmetric average commute time matrix $\boldsymbol{C}$ based on $\boldsymbol{Z}$ as follows:

$$m_{ij} = \frac{z_{jj} - z_{ij}}{\pi_j} \qquad (2)$$

$$c_{ij} = \frac{z_{jj} - z_{ij}}{\pi_j} + \frac{z_{ii} - z_{ji}}{\pi_i} \qquad (3)$$

Using the example in Figure 2, we can obtain a 10×10 matrix $\boldsymbol{M}$ based on the above steps (10 nodes in total). We list the result involving only entity nodes in Table 1.

$m(row, col)$	Lucille	1958	he	1936
Lucille	0.0	64.2	24.7	**74.7**
1958	24.3	0.0	17.3	90.2
he	22.6	32.1	0.0	87.7
1936	**20.2**	101.3	37.1	0.0

Table 1: Mean first-passage time matrix M for E1.

Argument Role Identification: We notice that argument roles can be identified based on the mean first-passage time. In a weighted directed graph, m_{ij} and m_{ji} are not necessarily similar. Actually in many cases nodes that lie on the boundaries have shorter mean first-passage time to the central nodes in the graph while there exists longer mean first-passage time from a central node to a node close to the boundary. A central node is more likely to be the central argument. We define the first argument as the more important argument. Therefore, we can regard v_i as the first argument of the argument pair (v_i, v_j) if m_{ij} is larger than m_{ji}. If m_{ij} and m_{ji} are equal, v_i and v_j have similar argument roles. For example, the boundary node *"1936"* in Figure 2 has shorter first-passage time to the central node *"Lucille Clifton"* (*i.e.*, the first argument) compared with the reverse direction.

2.3 Node Importance Computation

As we mentioned earlier, a candidate relational triple is more likely to be salient if it involves important entities of the sentence. In this section, we illustrate the node importance computation based on the extended dependency tree of a sentence.

TextRank (Mihalcea and Tarau, 2004) can be used to compute the importance of each node within G. Similarly, suppose a random walker keeps visiting adjacent nodes in G at random. The expected percentage of walkers visiting each node converges to the TextRank score.

We define a set of preferred nodes $\mathcal{R}$ which correspond to entities in a sentence. We assign higher preferences toward these nodes when computing the importance scores since entities are more informative for relation extraction (Björkelund and Farkas, 2012). We extend TextRank by introducing a new measure called "back probability" $d \in [0, 1]$ to determine how often walkers jump back to the nodes in $\mathcal{R}$ so that the converged score can be used to estimate the relative probability of visiting these preferred nodes. We define a preference vector $\boldsymbol{p}_{\mathcal{R}} = \{p_1, ..., p_{|\mathcal{V}|}\}$ such that the probabilities sum to 1, and p_k denotes the relative importance attached to v_k. p_k is set to $1/|\mathcal{R}|$ for $v_k \in \mathcal{R}$, otherwise 0. Let I be the $1 \times |\mathcal{V}|$ importance vector to be computed over all nodes as follows.

$$I(i) = (1-d) \sum_{j \in \mathcal{N}(i)} \frac{w_{ji}}{\sum_{k \in \mathcal{N}(j)} w_{jk}} I(j) + d \cdot p_i \quad (4)$$

ENTITY i	Lucille	he	1958	1936
$I(i)$	0.28	0.12	0.06	0.01

Table 2: Importance score of each entity in $E1$.

2.4 Combination and Filtering

Given the average commute time c_{ij} between nodes v_i and v_j (Section 2.2) and their relative importance scores $I(i)$ and $I(j)$ in G (Section 2.3), we will discuss how to combine them and generate the final score which can be used to measure the relation strength between two nodes. Intuitively, there exists a strong relation when there is a shorter distance between two relatively important nodes.

Previous approaches (Spagnola and Lagoze, 2011; Guo et al., 2011) consider the distance between two nodes and the influence of each node modeled by its weighted frequency to measure the strength of links in networks. Similarly, in our setting we can regard c_{ij} as the distance between v_i and v_j and use the relative importance score to measure the influence of each node in G. Therefore, we obtain Equation 5 to compute the relation strength $F(i, j)$ between nodes v_i and v_j. We are more confident in predicting the existence of a salient relation with stronger relation strength.

$$F(i, j) = \frac{I(i) \times I(j)}{c_{ij}^2} \quad (5)$$

Relation Filtering: We get a complete entity graph since we analyze the connection between any two entities in a sentence. In this work, we focus on identifying the most significant structures among entities based on the connection strength we have obtained. Since the entity graph is undirected, we can simply apply the maximum spanning tree algorithm to keep those relatively important pairs. For $E1$, we obtain three argument pairs resulting after filtering: (*"Lucille"*, *"1936"*), (*"Fred"*, *"1958"*), and (*"Lucille"*, *"Fred"*). In comparison, the relations between argument pairs such as (*"1958"*, *"1936"*) and (*"he"*, *"1936"*) are less important.

3 Relation Grounding

We have presented how to extract candidate argument pairs in Section 2. In this section, we first introduce how to rank the context words given a pair of arguments (Section 3.1). Then we describe methods of learning KB relation representations from existing KB triples based on pretrained word embeddings. Finally we ground each relational triple to a KB relation or assign NONE (Section 3.2).

3.1 Context Word Selection and Weighting

In this section, we introduce how to extract informative context words and their associated weights given an argument pair (v_i, v_j) in a sentence based on the average commute time matrix $\mathbf{C}$ introduced in Section 2.2. Previous work (Yu and Ji, 2016) regards this problem as finding important nodes in G relative to given arguments. However, they need to run the algorithm repeatedly to analyze the same graph for each argument pair. Here we discuss an efficient method to extract weighted context words.

We only keep nouns, verbs, adjectives, prepositions, and particles as indicative context words $\mathcal{X}$. We assume that a context word $v_k \in \mathcal{X}$ is more important relative to (v_i, v_j) if $c_{ik} + c_{kj}$ is close to c_{ij}. Actually if the relation between v_i and v_j does not rely on any indicative words, c_{ij} will be much smaller than $c_{ik} + c_{kj}$ considering other nodes in the same sentence. We denote Λ as the weight set for all the context words of a given argument pair (v_i, v_j) as follows. The higher λ_k is, the more important the context word v_k is relative to (v_i, v_j).

$$\lambda_k = \frac{c_{ij}}{c_{ik} + c_{kj}} \tag{6}$$

In $E1$, given the argument pair (*Lucille Clifton*, *Fred James Clifton*), we generate the following weighted context words: $\{$*married* : 0.60, *in*^{1} : 0.36, *born* : 0.29, *in*^{2} : 0.24$\}$.

3.2 Grounding

The associated weighted context words of each candidate argument pair are not sufficiently informative and flexible to clearly express the relation between two arguments. Thus, we aim to name the relation between a pair of arguments as one of the KB relations or NONE by comparing the semantic representations of context words and KB relations based on word embeddings. We also learn argument type signatures from KB triples.

For each word we obtain its pretrained word embedding $\boldsymbol{e} \in \mathbb{R}^k$ where k is the embedding dimensionality. For a phrase which contains multiple words, we simply average the vectors of all the single words in the phrase as its embedding.

Given a KB triple (h, l, t) composed of two entities h, t and a KB relation $l \in \mathcal{L}$ (the set of KB relations), we leverage a large-scale KB to learn the representation for each KB relation motivated by the basic idea behind previous studies (Bordes et al., 2013; Mikolov et al., 2013) that relation patterns can be represented as linear translations. We use $\mathcal{S}_l = \{(h_i, l, t_i), i = 1, \ldots, |\mathcal{S}_l|\}$ to represent all the KB triples with the KB relation l.

KB relation type names can also provide important semantic information for relation representation and disambiguation especially when multiple relations co-occur in the same sentence, such as family relations (e.g., *spouse*, *parents*, and *other family*). We segment a compound name of a KB relation type into a set of words. For example, we separate a DBpedia relation type name *political-Groups* into $\{$*political*, *groups*$\}$. Similarly, we average the vectors of all the words in a relation type name as its embedding $\widetilde{\boldsymbol{e}}_l \in \mathbb{R}^k$. Incorporating both implicit semantics from KB tuples and explicit semantics from KB relation names, we represent the relation embedding of each KB relation l as follows.

$$\boldsymbol{e}_l = \frac{1}{|\mathcal{S}_l|} \sum_{i=1}^{|\mathcal{S}_l|} (\boldsymbol{e}_{h_i} - \boldsymbol{e}_{t_i} + \widetilde{\boldsymbol{e}}_l) \tag{7}$$

Both of the involved embeddings are obtained from the linear combination of pretrained word embeddings, which guarantees that they are in the same space.

Given a single KB relation type l, an argument pair (v_i, v_j) and a single context word x, we can compute the cosine similarity between any candidate open RE triple and any KB relation. We calculate the absolute value since we have already captured the direction of arguments in Section 2.2. Therefore, we can regard similarity scores -1 and 1 equally and 0 as the lowest score.

$$S(l, (i, j, x)) = \frac{|\boldsymbol{e}_x \cdot \boldsymbol{e}_l|}{\|\boldsymbol{e}_x\| \|\boldsymbol{e}_l\|} \tag{8}$$

When there are multiple context words $x \in \mathcal{X}$, we can compute the weighted cosine similarity between them as follows based on the squared weights of context words described in Section 3.1.

$$S(l, (i, j, \mathcal{X})) = \max_{x \in \mathcal{X}} S(l, (i, j, x)) \times \lambda_x^2 \tag{9}$$

Since we have multiple KB relations $l \in \mathcal{L}$, we can ground a candidate relational triple $(i, j, \mathcal{X})$ and obtain its relation $\widehat{l}_{i,j,\mathcal{X}}$ considering all the possible relations. The predicted relation can either be assigned a valid KB relation or NONE. We use a marker to denote the relation between v_i and v_j which cannot be grounded to any KB relation.

$$\widehat{l}_{i,j,\mathcal{X}} = \arg\max\nolimits_{l \in \mathcal{L}} S(l, (i, j, \mathcal{X})) \tag{10}$$

Relation Argument Type Constraints

For each KB relation, we can obtain its type constraints for its two arguments. Take the relation *birthPlace* as an example: the entity types of two arguments should be person and location.

Given all the KB triples, we can estimate the probability of one of the arguments belonging to a certain entity type $z \in \mathcal{Z}$, where $\mathcal{Z}$ represents the set of all the KB concept types. For a given KB relation l, we define $c(k \rightsquigarrow z \,|\, l)$ to be the number of times the k_{th} argument is seen paired with the entity type z where $k \in \{1, 2\}$ since there are two arguments. Given the above definitions, the maximum likelihood estimate is as follows.

$$p(k, z \,|\, l) = \frac{c(k \rightsquigarrow z \,|\, l)}{\sum_{z \in \mathcal{Z}} c(k \rightsquigarrow z \,|\, l)} \qquad (11)$$

Therefore, given a candidate argument pair (v_i, v_j) and their entity types z_i and z_j, we can compute the probability of its being labeled as the relation l by considering both $p(1, z_i \,|\, l)$ and $p(2, z_j \,|\, l)$. We set $S(l, (i, j, \mathcal{X}))$ to 0 if the harmonic mean of $p(1, z_i \,|\, l)$ and $p(2, z_j \,|\, l)$ is smaller than a given threshold which will be introduced later in Section 4.4. We will not consider a candidate KB relation for comparison if the argument type of i or j fails to satisfy its type constraints. In this way, we can filter out some candidate triples and reduce the number of similarity computations. For example, given a KB relation *placeOfBurial*, the concept type *Species* is less likely to be the correct second argument type compared with other entity types such as *City* and *Location*. Remind that the order of arguments in the candidate triple has been introduced in Section 2.2.

4 Experiments

4.1 Knowledge Base and Word Embeddings

We use the April 2016 dump of DBpedia as our KB which contains $2,060$ relation types and $30,024,093$ relation triples in total. We use the 300-dimensional GloVe vectors (Pennington et al., 2014) pretrained on 6 billion tokens from the English Gigaword Fifth Edition and a 2014 Wikipedia dump.

4.2 Evaluation based on Slot Filling

There are several benchmarks developed for open RE (*e.g.*, (Fader et al., 2011; Stanovsky and Dagan, 2016)). However, we mainly focus on relations between entities and therefore we cannot directly compare with state-of-the-art open RE methods on those datasets. To evaluate the effectiveness of our approach, we choose the TAC-KBP SF (McNamee and Dang, 2009; Ji et al., 2010, 2011; Surdeanu and Ji, 2014) task as our evaluation platform which has been widely used by open RE methods (Soderland et al., 2013; Angeli et al., 2015b) since 2009. The goal of SF is to extract the values (***slot fillers***) of specific attributes (***slot types***) for a given entity (***query***) from a large-scale corpus which includes news documents, web blogs, and discussion forum posts. Justification sentences should be provided to support slot fillers. SF defines 25 slot types for person queries and 16 slots for organization queries.

We use the SF 2013 dataset for which we can compare with the ground truth and state-of-the-art open RE results reported in SF. We obtain $1,701$ relevant documents from the official evaluation assessment for 50 person queries and 50 organization queries. We manually map KB relations to slot types based on TAC-KBP slot descriptions.[3] Note that a single KB relation can be mapped to multiple slot types. For example, *birthPlace* can be mapped to *per:city_of_birth*, *per:stateofprovince_of_birth*, and *per:country_of_birth*. We assign a subtype (e.g., country, province, or city) to a location entity based on gazetteer matching.

DBpedia Relations	Slot Types
founder	org:founded_by
keyPeople	org:top_members_employees
education	per:schools_attended
workInstitution	per:employee_or_member_of
birthDate	per:date_of_birth

Table 3: Example Mappings from DBpedia relations to slot types.

We ignore all the slot types which require nominal phrases as fillers (*e.g.*, *per:cause_of_death*) and slot types *per/org:alternate_names* which depend on cross-document coreference resolution. We apply Stanford CoreNLP (Manning et al., 2014) for English part-of-speech tagging, name tagging, time expression extraction, dependency parsing, and coreference resolution. We use the official Slot Filling evaluation scoring metrics: Precision (P), Recall (R), and F-measure (F_1).

As shown in Table 4, our method outperforms the KBP2013 SF submission from the University of Washington (Soderland et al., 2013) which applies Open IE V4.0, which is an extension of SRL-based IE (Christensen et al., 2011) and noun

[3] The resource is publicly available for research purposes at: http://nlp.cs.rpi.edu/data/dbpedia2slot.zip.

Method		P	R	F_1
UW Official (Soderland et al., 2013)		**69.9**	12.2	20.8
UMass Official (Singh et al., 2013)		10.6	19.5	13.7
Our Approach	[1] KB Tuples	17.3	21.1	19.0
	[2] Relation Names	24.3	30.9	27.2
	[1]+[2] Joint	26.2	**32.4**	**28.9**

Table 4: Performance (%) on KBP2013 English SF based on different relation representations.

phrase processing (Pal and Mausam, 2016), to generate relation triples. This is their latest published approach which uses Open IE for Regular Slot Filling. Their approach achieves very high precision but comparatively low recall (12.2%). In our experiments, we keep all the candidate triples which could be mapped to a slot type without tuning thresholds. On the same dataset, we also compare with an approach (Singh et al., 2013) which extracts relations with matrix factorization and *universal schemas* (Riedel et al., 2013) consisted of textual patterns and all the slot types. We do not directly compare with the work of Angeli et al. (2015b) because of the lack of access to their SF output.[4]

The importance-based strategy is effective at extracting more salient information. For example, previous methods only extract one argument pair (*“the top Egyptian cleric”*, *“Wednesday”*) from the sentence *“**Sheikh Tantawi**, the top **Egyptian** cleric who died on **Wednesday** on a visit to . . . ”* while omitting the person name. Our method extracts both (*“Sheikh Tantawi”*, *“Egyptian”*) and (*“Sheikh Tantawi”*, *“Wednesday”*) with their associated top-weighted context words *“cleric”* and *“died”* respectively, since the connection between *“Egyptian”* and *“Wednesday”* is much weaker.

Compared with relation phrases, the word embeddings of weighted context words are more flexible for comparison when we map relational triples to a known schema. For example, it is impossible for previous methods (*e.g.*, (Soderland et al., 2013)) to summarize all the related mentions (*e.g.*, *“appointed”* and *“CEO”*) and manually map them to the relation *employment*. Therefore previous approaches missed the slot filler *“Al-Azhar University”* of the query *“Mohammed Sayed Tantawi”* from the following sentence *“Tayeb, the president of **Al-Azhar University** since 2003, succeeds **Mohammed Sayed Tantawi**”* as *“succeeds”* was not included into the related terms. Our approach extracts it based on their semantic representations.

In addition, we obtain more generalized relation type names based from the KB schema. For example, we ground the relation in $E2$ between *“Patricia”* and *“Gary”* to *influencedBy*. Similarly, in the sentence *“**Ginzburg** shared the Nobel Physics Prize with US physicists **Alexei Abrikosov** and Anthony Leggett for their contributions to the theory of superconductors ...”*, the relation phrase *“shared the Nobel Physics Prize with”* between *“Ginzburg”* and *“Alexei”* is too specific compared with the grounded KB relation *alongside* by our approach for subsequent applications.

4.3 Impact of Relation Representations

In Section 3.2, we use KB tuples and their relation type names to learn KB relation representations. As shown in Table 4, our approach can already achieve promising performance based on the relation representations learned from KB relation names. However, sometimes relations are implicitly expressed. It is likely that the context words of a relation triple and its corresponding KB relation name are not semantically similar. In this case, we need more general relation representations with the help of millions of KB tuples. For example, we can ground the relation *school* between *“McGregor”* and *“Colorado State University”* successfully by comparing the representation of context words *“tight”* and *“end”* with the joint relation representations from the following sentence: *“**McGregor** was a two-time All-America tight end at **Colorado State University**”* even though this relation is not explicitly described.

4.4 Impact of Argument Type Constraints

As mentioned in Section 3.2, we aim to filter out some candidate relation triples if the entity types of the arguments are not popular for a given KB relation. By tuning thresholds, there are no significant differences in performance when the threshold falls in the range 0.05–0.2. On the other hand, if the threshold is set too high (*e.g.*, greater than 0.35%), we will mistakenly discard correct candidates which satisfy type constraints.

We implement Jenks optimization (Wikipedia, 2017) to automatically split the frequency values of all entity types into two tiers given a certain argument position and a KB relation. This is done by minimizing each tier’s average deviation from

[4] The highest recall they achieve is around 13% on all the slot types including nominal relations on the same dataset.

the tier mean, while maximizing each tier's deviation from the means of other groups (McMaster and McMaster, 2002). We set the threshold automatically using the obtained natural breaks for two arguments respectively to compute the harmonic mean of them. This approach achieves 28.9% F_1 which is comparable to the highest F_1 (29.2%) obtained by threshold tuning.

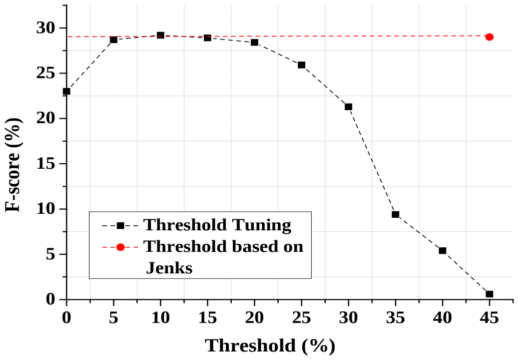

Figure 3: Performance (%) based on different thresholds for argument type constraints.

5 Related Work

5.1 Open Information Extraction

Lexical or syntactic features and patterns have been widely used to extract relational triples (Suchanek et al., 2009; Poon and Domingos, 2009; Wu and Weld, 2010; Nakashole et al., 2011; Fader et al., 2011; Nakashole et al., 2012; Mausam et al., 2012; Bovi et al., 2015; Angeli et al., 2015b; Grycner and Weikum, 2016). Our work explores the global structure of a dependency tree to identify salient triples within a sentence. Some open IE approaches have the capability to extract relations between concepts or phrases (Kok and Domingos, 2008; Min et al., 2012; Del Corro and Gemulla, 2013). Currently we focus on relations between two entities.

Given the SF schema, Soderland et al. (2013) manually design rules to map relational triples to slot types within hours. Researchers also use distantly labeled corpora to compute the PMI^2 value between open IE and SF relation pairs (Angeli et al., 2015b). Instead, we propose a novel grounding approach which facilitates building a mapping table between KB relations and slot types. We do not compare with RE methods specifically designed for SF (Sun et al., 2011; Li et al., 2012; Angeli et al., 2015a) since these methods actively search for candidate fillers of the given queries based on slot-specific training resources while ignoring the salient relations which are irrelevant to the queries or the predefined slot types.

5.2 Relation Grounding

Besides textual features, large-scale knowledge bases are widely used for distant supervised relation extraction (Mintz et al., 2009; Riedel et al., 2010) to deal with the challenges caused by insufficient training data. Weston et al. (2013) combine two relation representations trained from KB triples and context words independently for relation extraction. Recent studies such as (Toutanova et al., 2015) train relation representations of KB and textual relations jointly. Another kind of representations combining matrix factorization (Riedel et al., 2013) with first-order logic information is learned by Rocktäschel et al. (2015). Compared with these previous efforts, our unsupervised grounding method does not need the aligned training corpus or relation mentions for KB tuples. Wijaya and Mitchell (2016) introduce an approach to map words to KB relations based on web text, but they only focus on verb phrases.

5.3 Node Importance Computation

Graph-based algorithms such as PageRank (Page et al., 1999) and TextRank (Mihalcea and Tarau, 2004) are useful in keyword extraction. The way we rank nodes is most similar to the work of White and Smyth (2003) and Yu and Ji (2016) which generate the relative importance score of each node toward a set of preferred nodes. However, they only deal with unweighted undirected graphs.

6 Conclusions and Future Work

We propose an unsupervised open relation extraction method by exploring the global structure of dependency tree and show its effectiveness in extracting salient candidate relation triples. We also leverage the knowledge from the large-scale KB relation triples and weighted context words based on general embeddings to enhance the quality of our relation grounding technique. Experiments on English Slot Filling demonstrate that our approach outperforms state-of-the-art open RE approaches. In the future, we aim to extend our framework for multilingual open RE based on the KB schema.

Acknowledgments

This work was supported by the DARPA DEFT No. FA8750-13-2-0041, U.S. ARL NS-CTA No. W911NF-09-2-0053, and NSF IIS 1523198. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

David Aldous and Jim Fill. 2002. Reversible markov chains and random walks on graphs.

Gabor Angeli, Sonal Gupta, Melvin Johnson Premkumar, Christopher D Manning, Christopher Ré, Julie Tibshirani, Jean Y Wu, Sen Wu, and Ce Zhang. 2015a. Stanford's distantly supervised slot filling systems for kbp 2014. In *Proceedings of the TAC*, Gaithersburg, MD.

Gabor Angeli, Melvin Johnson Premkumar, and Christopher D Manning. 2015b. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the ACL*, pages 344–354, Beijing, China.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735, Busan, Korea.

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of the ACL*, pages 28–36, Columbus, OH.

Nikita Bhutani, HV Jagadish, and Dragomir R Radev. 2016. Nested propositions in open information extraction. In *Proceedings of the EMNLP*, pages 55–64, Austin, TA.

Anders Björkelund and Richárd Farkas. 2012. Data-driven multilingual coreference resolution using resolver stacking. In *Proceedings of the EMNLP-CoNLL*, pages 49–55, Jeju, South Korea.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the NIPS*, pages 2787–2795, Lake Tahoe, NV.

Claudio Delli Bovi, Luca Telesca, and Roberto Navigli. 2015. Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *TACL*, 3:529–543.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2011. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the 6th International Conference on Knowledge Capture*, pages 113–120, Banff, Canada.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: clause-based open information extraction. In *Proceedings of the WWW*, pages 355–365, Rio de Janeiro, Brazil.

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

Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. 2007. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE TKDE*, 19:355–369.

Adam Grycner and Gerhard Weikum. 2016. Poly: Mining relational paraphrases from multilingual sentences. In *Proceedings of the EMNLP*, pages 2183–2192, Austin, TX.

Jun Guo, Hanliang Guo, and Zhanyi Wang. 2011. An activation force-based affinity measure for analyzing complex networks. *Scientific reports*, 1:113–122.

Heng Ji, Ralph Grishman, and Hoa T. Dang. 2011. An overview of the tac2011 knowledge base population track. In *Proceedings of the TAC*, Gaithersburg, MD.

Heng Ji, Ralph Grishman, Hoa T. Dang, Kira Griffitt, and Joe Ellis. 2010. An overview of the tac2010 knowledge base population track. In *Proceedings of the TAC*, Gaithersburg, MD.

Douglas J Klein and Milan Randić. 1993. Resistance distance. *J. Math. Chemistry*, 12:81–95.

Stanley Kok and Pedro Domingos. 2008. Extracting semantic networks from text via relational clustering. In *Proceeding of the Joint Conference on Machine Learning and Knowledge Discovery in Databases*, pages 624–639, Antwerp, Belgium.

Yan Li, Sijia Chen, Zhihua Zhou, Jie Yin, Hao Luo, Liyin Hong, Weiran Xu, Guang Chen, and Jun Guo. 2012. Pris at tac2012 kbp track. In *Proceedings of the TAC*, Gaithersburg, MD.

Yanhua Li and Zhi-Li Zhang. 2010. Random walks on digraphs, the generalized digraph laplacian and the degree of asymmetry. In *Proceedings of the WAW*, pages 74–85, Stanford, CA.

László Lovász. 1993. Random walks on graphs. *Combinatorics, Paul erdos is eighty*, 2:1–46.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the*

ACL: System Demonstrations, pages 55–60, Baltimore, MD.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the Joint Conference on EMNLP and CONLLL*, pages 523–534, Jeju, South Korea.

Robert McMaster and Susanna McMaster. 2002. A history of twentieth-century american academic cartography. *Cartography and Geographic Inform. Sci.*, 29:305–321.

Paul McNamee and Hoa Trang Dang. 2009. Overview of the tac 2009 knowledge base population track. In *Proceeding of the TAC*, Gaithersburg, MD.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of the EMNLP*, pages 404–411, Barcelona, Spain.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, Lake Tahoe, NV.

Bonan Min, Shuming Shi, Ralph Grishman, and Chin-Yew Lin. 2012. Ensemble semantics for large-scale unsupervised relation extraction. In *Proceedings of the 2012 Joint Conference on EMNLP and CONLL*, pages 1027–1037, Jeju, South Korea.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the ACL-IJCNLP*, pages 1003–1011, Suntec, Singapore.

Ndapandula Nakashole, Martin Theobald, and Gerhard Weikum. 2011. Scalable knowledge harvesting with high precision and high recall. In *Proceedings of the WSDM*, pages 227–236, Hong Kong, China.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. Patty: a taxonomy of relational patterns with semantic types. In *Proceedings of the Joint Conference on EMNLP and CoNLL*, pages 1135–1145, Jeju, South Korea.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, Stanford, CA.

Harinder Pal and Mausam. 2016. Demonyms and compound relational nouns in nominal open IE. In *Proceedings of the 5th Workshop on AKBC*, pages 35–39, San Diego, CA.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the EMNLP*, pages 1532–1543, Doha, Qatar.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of the EMNLP*, pages 1–10, Suntec, Singapore.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the Joint ECML and PAKDD*, pages 148–163, Barcelona, Spain.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the NAACL-HLT*, pages 74–84, Atlanta, GA.

Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. 2015. Injecting logical background knowledge into embeddings for relation extraction. In *Proceeding of the NAACL-HLT*, pages 1119–1129, Denver, CO.

Sameer Singh, Limin Yao, David Belanger, Ari Kobren, Sam Anzaroot, Mike Wick, Alexandre Passos, Harshal Pandya, Jinho D Choi, Brian Martin, et al. 2013. Universal schema for slot filling and cold start: Umass iesl at tackbp 2013. In *Proceedings of the TAC*, Gaithersburg, MD.

Stephen Soderland, John Gilmer, Robert Bart, Oren Etzioni, and Daniel S Weld. 2013. Open information extraction to kbp relations in 3 hours. In *Proceedings of the TAC*, Gaithersburg, MD.

Steve Spagnola and Carl Lagoze. 2011. Edge dependent pathway scoring for calculating semantic similarity in conceptnet. In *Proceedings of IWCS*, pages 385–389, Oxford, UK.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *Proceedings of the EMNLP*, pages 2300–2305, Austin, TX.

Fabian M Suchanek, Mauro Sozio, and Gerhard Weikum. 2009. Sofie: a self-organizing framework for information extraction. In *Proceedings of the International Conference on WWW*, pages 631–640, Madrid, Spain.

Ang Sun, Ralph Grishman, Wei Xu, and Bonan Min. 2011. Nyu 2011 system for kbp slot filling. In *Proceedings of the TAC*, Gaithersburg, MD.

Mihai Surdeanu and Heng Ji. 2014. Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In *Proceedings of the TAC*, Gaithersburg, MD.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the EMNLP*, pages 1499–1509, Lisbon, Portugal.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the EMNLP*, pages 1366–1371, Seattle, WA.

Scott White and Padhraic Smyth. 2003. Algorithms for estimating relative importance in networks. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 266–275, Washington, D.C.

Derry Tanti Wijaya and Tom M Mitchell. 2016. Mapping verbs in different languages to knowledge base relations using web text as interlingua. In *Proceedings of the NAACL-HLT*, pages 818–827, San Diego, CA.

Wikipedia. 2017. Jenks natural breaks optimization — wikipedia, the free encyclopedia.

Fei Wu and Daniel S Weld. 2010. Open information extraction using wikipedia. In *Proceedings of the ACL*, pages 118–127, Uppsala, Sweden.

Ying Xu, Mi-Young Kim, Kevin Quinn, Randy Goebel, and Denilson Barbosa. 2013. Open information extraction with tree kernels. In *Proceedings of the NAACL-HLT*, pages 868–877, Atlanta, GA.

Dian Yu and Heng Ji. 2016. Unsupervised person slot filling based on graph mining. In *Proceedings of the ACL*, pages 44–53, Berlin, Germany.