Hierarchical Types Constrained Topic Entity Detection for Knowledge Base Question Answering

Yunqi Qiu, Manling Li, Yuanzhuo Wang, Yantao Jia, Xiaolong Jin
CAS Key Laboratory of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{yunqi_qiu, lmlylc2013, jamaths.h}@163.com, {wangyuanzhuo, jinxiaolong}@ict.ac.cn

ABSTRACT

Topic entity detection is to find out the main entity asked in a question, which is a significant task in question answering. Traditional methods ignore the information of entities, especially entity types and their hierarchical structures, restricting the performance. To take full advantage of KBs and detect topic entities correctly, we propose a deep neural model to leverage entity hierarchical types and entity relations in KB. The experimental results demonstrate the effectiveness of out method.

Keywords

Question Answering; Topic Entity Detection; Hierarchical Types

1. INTRODUCTION

Knowledge Base Question Answering (KBQA) aims to answer questions with simple facts in KB. Generally, KBQA systems have two key components: (1) topic entity detection, which detects topic entities in questions and links them to KB; (2) answer selection, which identifies KB relations linked to topic entities and selects correct answers. This paper focuses on topic entity detection, the foundation of getting correct answers.

Previous studies identify topic entities by existing entity linkers like Freebase API using the n-grams in questions. To better measure the similarity between the entity name and the question context, a bunch of statistical features are leveraged [1]. However, they only consider the string itself and ignore the question semantics and entity information in KB, such as entity types and relations, which are of great importance in topic entity detection, since the semantics of question context are always closely tied to the types and relations of topic entities. For example, the question "Who is the CEO of Apple?" implies the type "*organization.company*" and the relation "*CEO_of*" for topic entity "*Apple*", so the topic entity is the company Apple not the food. The effectiveness of relations has been demonstrated in [2], which uses KB relations to re-rank candidate topic entities and achieves the state-of-the-art accuracy. However, it suffers from zero-shot problem, i.e., test entities may have unseen relations in the training data. This is caused by the huge number of relations in KB, e.g., more than 6,000 relations in a small KB Freebase2M. To solve this problem, Yu et al. [2] breaks these relation names into 4,500 tokens, but it is still easy to find unseen tokens for test entities. As is known, types constitute a multi-layered hierarchical structure, i.e., type hierarchy. It can naturally ease the zero-shot problem, because a test entity always has a coarsest parent type (the type on the first layer of type hierarchy, e.g., person, location) with adequate training, and the infrequent child types can get support from their parent types. For example, thousands of types in Freebase can be mapped to 112 fine-grained hierarchical types [3], greatly reducing zero-shot problem.

Considering that hierarchical types can not only benefit the semantic understanding of questions, but also avoid zero-shot problem, we propose Hierarchical Types constrained Topic Entity Detection (HTTED) to increase detection accuracy. It is a neural model to match questions and entities by learning the representation of question context, entity hierarchical types and entity relations.

2. HTTED MODEL

The key idea of HTTED is to leverage the type hierarchy in KB by modeling the interplay of child and parent types. Given that the semantics of a parent type is close bound to the semantics of its child types, we represent the parent type as a semantic composition of all its child types, i.e.,

$$\mathbf{t}_p = \mathbf{t}_{c_1} \oplus \mathbf{t}_{c_2} \oplus \cdots \oplus \mathbf{t}_{c_k}, \tag{1}$$

where $\mathbf{t}_p \in \mathbb{R}^d$ is a parent type's embedding, $\mathbf{t}_{c_i} \in \mathbb{R}^d, i = 1, 2, \dots, k$ are its child types' embeddings, and d is the embedding dimension. $\oplus$ is a composition operator. In this paper, we consider addition operator, i.e., the sum of vectors. This formulation enables the model to share parameters between parent and child types, so that it helps learn embeddings of child types, which suffers from dearth of training data. Due to the hierarchical structure of types, the training data for parent types is much larger than child types. For example, 30.7% entities in FB2M only have the coarsest types and are not linked to any child types. With assistance of the training data for parent types, the embeddings of child types, especially infrequent ones, will train more sufficiently.

The overall architecture of HTTED model is illustrated in Figure 1, which is comprised of three modules: (1) question context encoder; (2) entity type encoder. (3) entity relation encoder; Given a question q and its candidate topic entities

WWW 2018 April 23–27, 2018, Lyon, France

DOI: 10.475/123_4

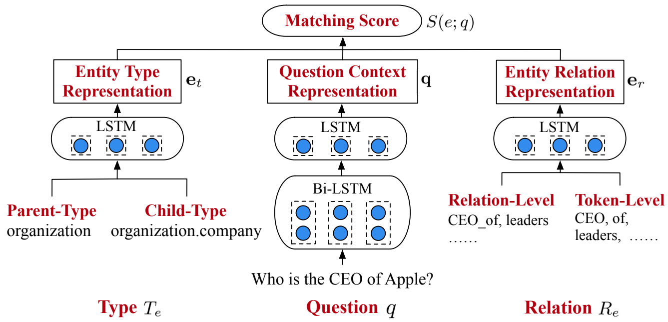

Figure 1: The architecture of HTTED. For a given question q and each of its candidate topic entity $e \in E_q$, we generate matching score $S(e;q)$ by encoding the question q, the entity's hierarchical types T_e and relations R_e.

E_q, HTTED learns the representations of question context, entity types and entity relations. Then, E_q is re-ranked in terms of the matching score between q and $e \in E_q$.

Question Context Encoder: Given a question q, we put word embeddings of question context $\{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$ into a deep LSTM network, where n is its length. The first layer is a bidirectional LSTM. The left-LSTM works from w_1 to w_n with the output $\mathbf{h}_n^l$, while the right-LSTM works from w_n to w_1 with the output $\mathbf{h}_n^r$. We concatenate these outputs $[\mathbf{h}_n^l, \mathbf{h}_n^r]$ and pass it through the second layer LSTM. The final output is the question context representation $\mathbf{q} \in \mathbb{R}^d$.

Entity Type Encoder: For each $e \in E_q$, we get its linked types T_e in KB. The embeddings of parent types are calculated by Eq.(1) according to type hierarchy. A single layer LSTM network is applied on these types' embeddings to generate entity type representation $\mathbf{e}_t \in \mathbb{R}^d$.

Entity Relation Encoder: For each $e \in E_q$, we get its linked relations R_e in KB, and the relation embeddings contribute to relation-level relation representations. Additionally, relation names are broken into tokens RT_e following [2], and the word embeddings of them are token-level relation representations. The two levels of relation representations are put into a LSTM network to get the last output as entity relation representation $\mathbf{e}_r \in \mathbb{R}^d$.

Matching Score: Intuitively, an entity could be represented by its relations and types. Thus, the matching score of e given q is

$$S(e;q) = \alpha \cdot \cos(\mathbf{e}_r, \mathbf{q}) + (1-\alpha) \cdot \cos(\mathbf{e}_t, \mathbf{q}) \qquad (2)$$

where $\alpha \in (0,1)$. In training step, we maximize the margin between the positive topic entity e^+ and other negative ones e^- with a ranking loss as following,

$$L = max\{0, \gamma - S(e^+, q) + S(e^-, q)\}. \qquad (3)$$

where γ is a constant parameter.

3. EXPERIMENTS AND ANALYSIS

We conducted experiments on a widely used single-relation KBQA dataset SimpleQuestions [4]. It contains 108,442 questions and each question can be answered with a single fact. The KB we used was FB2M, which contains 2,150,604 entities and 6,701 predicates, with 112 hierarchical types introduced by [3]. For comparison purpose, the candidate topic entity set was the one provided by [1].

All word embeddings were initialized with 300-d pre-trained word embeddings. The relation embeddings were randomly initialized, and type embeddings were initialized under the constraint of Eq.(1) with leaf node type embeddings randomly initialized. We tuned the following hyper-parameters: (1) the size of hidden states for LSTMs ({150, 200, 300}); (2) the learning rate ({0.01, 0.001}); (3) the weight α ({0.3, 0.5, 0.7}). We set $\gamma = 0.5$. Our evaluation metrics were Top-K accuracy (K = 1, 10, 20, 50) following [2]. The baselines adopted were Freebase API, Yin et al. [1] and Yu et al. [2]. To evaluate the effectiveness of hierarchical types, we also constructed a baseline without considering the hierarchical structures of types.

Table 1: Topic Entity Detection Accuracy

Methods	Top-1	Top-10	Top-20	Top-50
Freebase API	40.9	64.3	69.7	75.7
Yin et al., 2016	72.7	86.9	88.4	90.2
Yu et al., 2017	79.0	89.5	90.9	92.5
HTTED	**81.1**	**91.7**	**93.4**	**95.1**
−Hierarchy	80.0	90.3	92.2	94.2

We observe from Table 1 that HTTED outperforms all baselines on all evaluation metrics. (1) Compared to the state-of-the-art method, HTTED promotes Top-1 accuracy by 2.1%, and still performs well when the beam sizes are larger, i.e., with improvement 2.2% on top-10, 2.5% on top-20, 2.6% on top-50. It proves the ability of hierarchical types to help semantic understanding of questions and find correct topic entities. (2) Without type hierarchy, the model yields a performance drop, which demonstrates the effectiveness of hierarchical types in topic entity detection. With support from parent types, the child types can be trained more sufficiently, and thus the detection accuracy can be improved.

Table 2: Correct predictions with hierarchical types

Question	Topic Entity Ranking
What position does **carlos gomez** play?	**carlos gomez(person.athelete)** carlos gomez(person.actor) carlos gomez(person.artist)
What video game platform was **ghostbusters** released on?	**ghostbusters(product.game)** ghostbusters(art.film) ghostbusters(art.music)

It can be seen from Table 2 that with the help of hierarchical types, the entities with same names can be discerned, and the one more related to the semantics of questions can be selected, thus the correct topic entities are detected.

4. CONCLUSIONS

In this paper, we propose a deep LSTM network HTTED for topic entity detection. Apart from relation information, we leverage hierarchical fine-grained types, which could provide more constraints and avoid zero-shot problem.

5. REFERENCES

[1] W. Yin, M. Yu, B. Xiang, B. Zhou, and H. Schütze. Simple question answering by attentive convolutional neural network. *In Proceedings of COLING*, 2016.

[2] M. Yu, W. Yin, K. S. Hasan, C. dos Santos, et al. Improved neural relation detection for knowledge base question answering. *In Proceedings of ACL*, 2017.

[3] X. Ling and D. S. Weld. Fine-grained entity recognition. *In Proceedings of AAAI*, 2012.

[4] A. Bordes, N. Usunier, S. Chopra, and J. Weston. Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075, 2015.