The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Path-Based Attention Neural Model for Fine-Grained Entity Typing

Denghui Zhang,[1] **Manling Li,**[1] **Pengshan Cai,**[2] **Yantao Jia,**[1] **Yuanzhuo Wang,**[1]
[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China
[2]School of Computer Science, University of Massachusetts Amherst, MA 01003
zhangdenghui@ict.ac.cn, limanlingcs@gmail.com, pengshancai@cs.umass.edu,
jamaths.h@gmail.com, wangyuanzhuo@ict.ac.cn

Abstract

Fine-grained entity typing aims to assign entity mentions in the free text with types arranged in a hierarchical structure. It suffers from the label noise in training data generated by distant supervision. Although recent studies use many features to prune wrong label ahead of training, they suffer from error propagation and bring much complexity. In this paper, we propose an end-to-end typing model, called the path-based attention neural model (PAN), to learn a noise-robust performance by leveraging the hierarchical structure of types. Experiments on two data sets demonstrate its effectiveness.

Introduction

Fine-grained entity typing aims to assign types to entity mentions in the local context (a single sentence), and the type set constitutes a tree-structured hierarchy (i.e., type hierarchy). Recent years witness the boost of neural models in this task, e.g., (Shimaoka et al. 2016) employs an attention based LSTM to attain sentence representations and achieves state-of-the-art performance. However, it still suffers from noise in training data, which is a main challenge in this task. The training data is generated by distant supervision, which assumes that if an entity has a type in knowledge bases (KBs), then all sentences containing this entity will express this type. This method inevitably introduces irrelevant types to the context. For example, the entity “Donald Trump” has types “person”, “businessman” and “politician” in KBs, thus all three types are annotated for its mentions in the training corpora. But in sentence “Donald Trump announced his candidacy for President of US.”, only “person” and “politician” are correct types, while “businessman” can not be deduced from the sentence, thus serves as noise. To alleviate this issue, a few systems try to denoise training data by filtering irrelevant types ahead of training. For instance, (Ren et al. 2016) proposes PLE to identify correct types by jointly embedding mentions, context and type hierarchy, and then use clean data to train classifiers. However, the denoising and training process are not unified, which may cause error propagation and bring much additional complexity.

Motivated by this, we propose an end-to-end typing model, called the Path-based Attention Neural model (PAN), to select relevant sentences to each type, which can dynamically reduce the weights of wrong labeled sentences for each type during training. This idea is inspired by some successful attempts to reduce noise in relation extraction, e.g.,(Lin et al. 2016). However, these methods fail to consider type hierarchy, which is distinct in fine-grained entity typing. Specifically, if a sentence indicates a type, its parent type can be also deduced from the sentence. Like the example above, “politician” is the subtype of “person”. Since the sentence indicates that “Donald Trump” is “politician”, “person” should also be assigned. Thus, we build path-based attention for each type by utilizing its path to its coarsest parent type (e.g., person) in the type hierarchy. Compared to the simple attention in relation extraction, it enables parameter sharing for types in the same path. With the support of hierarchical information of types, it can reduce noise effectively.

Path-Based Attention Neural Model

The architecture of PAN is illustrated in Figure1. Supposing that there are n sentences containing entity e, i.e., $\mathcal{S}_e = \{s_1, s_2, ..., s_n\}$, and $\mathcal{T}_e$ is the automatically labeled types based on KBs. Firstly PAN employs LSTM to generate representations of sentences $\mathbf{s}_i$ following (Shimaoka et al. 2016), where $\mathbf{s}_i \in \mathbb{R}^d$ is the semantic representation of s_i, $i \in \{1, 2, ..., n\}$. Afterwards, we build path-based attention $\alpha_{i,t}$ over sentences s_i for each type $t \in \mathcal{T}_e$, which is expected to focus on relevant sentences to type t. Then, the representation of sentence set $\mathcal{S}_e$ for type t, denoted by $\mathbf{s}_{e,t} \in \mathbb{R}^d$, is calculated through weighted sum of vectors of sentences. Finally, we obtain predicted types through a classification layer.

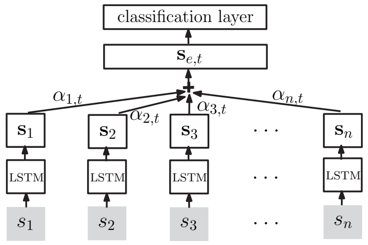

Figure 1: The architecture of PAN for given entity e, type t

More precisely, given e, an attention $\alpha_{i,t}$ is learned to score how well sentence s_i matches type t, i.e.,

$$\alpha_{i,t} = \frac{\exp(\mathbf{s}_i \mathbf{A} \mathbf{p}_t)}{\sum_{j=1}^{n} \exp(\mathbf{s}_j \mathbf{A} \mathbf{p}_t)},$$

where $\mathbf{A} \in \mathbb{R}^{d \times d}$ is a weighted diagonal matrix. $\mathbf{p}_t \in \mathbb{R}^d$ is the representation of path p_t for type t. Specifically, for each type, we define one path as a sequence of types starting from its coarsest parent type, and ending with it. More formally, for type t_l, $p_{t_l} = t_1 \rightarrow t_2 \rightarrow ... \rightarrow t_l$, where t_1 is its coarsest parent type, and t_{i+1} is the subtype of t_i. For example, for type $t_l = politician$, its path is $p_{t_l} = person \rightarrow politician$. We represent the path p_{t_l} as a semantic composition of all the types on the path, i.e., $\mathbf{p}_{t_l} = \mathbf{t}_1 \circ \mathbf{t}_2 \circ ... \circ \mathbf{t}_l$, where $\mathbf{t}_i \in \mathbb{R}^d$ is the representation of type t_i, which is a parameter to learn. $\circ$ is a composition operator. In this paper, we consider two types of operators: (1) Addition (PAN-A), where $\mathbf{p}_{t_l}$ equals the sum of type vectors. (2) Multiplication (PAN-M), where $\mathbf{p}_{t_l}$ equals the cumulative product of type vectors. In this way, path-based attention enables the model to share parameters between types in the same path. For example, the attention learned for "person" could assist the learning of the attention for "politician". It makes learning easier especially for infrequent subtypes, which suffer from dearth of training data, since the attentions for these subtypes can get support from the attention for parent type.

Then, the representation of sentence set $\mathcal{S}_e$ for type t, i.e., $\mathbf{s}_{e,t}$, is calculated through weighted sum of sentence vectors,

$$\mathbf{s}_{e,t} = \sum_{i=1}^{n} \alpha_{i,t} \mathbf{s}_i.$$

Since one mention can have multiple types, we employ a classification layer consisting of N logistic classifiers, where N is the total number of types. Each classifier outputs the probability of respective type, i.e.,

$$P(t|\mathbf{s}_{e,t}) = \frac{\exp(\mathbf{w}_t^{\mathrm{T}} \mathbf{s}_{e,t} + \mathbf{b}_t)}{1 + \exp(\mathbf{w}_t^{\mathrm{T}} \mathbf{s}_{e,t} + \mathbf{b}_t)},$$

where $\mathbf{w}_t, \mathbf{b}_t \in \mathbb{R}^d$ are the logistic regression parameters. To optimize the model, a multi-type loss is defined according to the cross entropy as follows,

$$J = -\sum_{e} \sum_{t} [\mathbb{I}_t \ln P(t|\mathbf{s}_{e,t}) + (1 - \mathbb{I}_t) \ln(1 - P(t|\mathbf{s}_{e,t}))],$$

where $\mathbb{I}_t$ is indicator function to indicate whether t is the annotated type of entity e, i.e., $t \in \mathcal{T}_e$.

Experiments and Conclusion

Experiments are carried on two widely used datasets OntoNotes and FIGER(GOLD), and the training dataset of OntoNotes is noisy compared to FIGER(GOLD) (Shimaoka et al. 2016). Evaluation measures are Strict Accuracy (Acc), Loose Macro F1 (MaF1), and Loose Micro F1 (MiF1) (Shimaoka et al. 2016). The baselines are chosen from two aspects: (1) Predicting types in a unified process using raw noisy data, i.e., TLSTM (Shimaoka et al. 2016), and other methods in Table1. (2) Predicting types using clean data by denoising ahead, i.e., H_PLE and F_PLE (Ren et al. 2016). To prove the superiority of path-based attention, we also directly apply the attention mechanism in relation extraction (Lin et al. 2016) without considering type hierarchy (AN).

Table 1: Performance on FIGER(GOLD) and OntoNotes

Metric	OntoNotes			FIGER(GOLD)		
	Acc	MaF1	MiF1	Acc	MaF1	MiF1
HYENA	24.9	49.7	44.6	28.8	52.8	50.6
FIGER	36.9	57.8	51.6	47.4	69.2	65.5
TLSTM	50.8	70.1	64.9	59.7	79.0	75.4
AN	52.3	71.7	65.2	60.0	79.5	75.9
PAN-A	54.9	**72.8**	**66.5**	**60.2**	**79.9**	**76.2**
PAN-M	53.0	71.9	65.3	60.0	79.4	76.0
H_PLE	54.6	69.2	62.5	54.3	69.5	68.1
F_PLE	**57.2**	71.5	66.1	59.9	76.3	74.9

We can observed that: (1) Compared with HYENA, FIGER and TLSTM using the same raw noisy data, PAN performs best on both data sets, which proves the anti-noise ability of PAN. (2) Compared with H_PLE and F_PLE using denoised data, PAN using raw noisy data still achieves highest Ma-F1 and Mi-F1. It makes sense that F_PLE has higher Acc on OntoNotes since the noise is reduced before training, but it needs to learn additional parameters about mentions, context and types, while PAN only needs to learn parameters of attention. Thus, PAN is more efficient to reduce noise. (3) PAN performs better than AN, since the path-based attention utilizes hierarchical structures to enable parameter sharing. (4) The improvements on OntoNotes are higher than FIGER(GOLD), because OntoNotes is more noisy, and the hierarchical structure in OntoNotes is more complex with more layers, which further demonstrates that path-based attention handles well with type hierarchy and noise. (5) PAN-A achieves better performance than PAN-M, which shows that addition operator can better capture type hierarchy.

In conclusion, PAN can reduce noise effectively through an end-to-end process, and achieves better typing performance on datasets with more noise.

Acknowledgments

The work was funded by National Natural Science Foundation of China (No. 61572469, 61402442, 91646120, 61572473, 61402022), the National Key R&D Program of China (No. 2016QY02D0405, 2016YFB1000902), National Grand Fundamental Research 973 Program of China (No. 2013CB329602, 2014CB340401), and 2016 Key Program of National Network Space Security.

References

Lin, Y.; Shen, S.; Liu, Z.; and et al. 2016. Neural relation extraction with selective attention over instances. In *ACL*.

Ren, X.; He, W.; Qu, M.; Voss, C. R.; Ji, H.; and Han, J. 2016. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *KDD*.

Shimaoka, S.; Stenetorp, P.; Inui, K.; and Riedel, S. 2016. Neural architectures for fine-grained entity type classification. In *EACL*.