Improving Fine-grained Entity Typing with Entity Linking

Hongliang Dai[1], Donghong Du[1,3], Xin Li[2], and Yangqiu Song[1]
[1]Department of CSE, HKUST
[2]Tencent Technology (SZ) Co., Ltd.
[1]{hdai,yqsong}@cse.ust.hk
[2]alonsoli@tencent.com
[3]dduaa@connect.ust.hk

Abstract

Fine-grained entity typing is a challenging problem since it usually involves a relatively large tag set and may require to understand the context of the entity mention. In this paper, we use entity linking to help with the fine-grained entity type classification process. We propose a deep neural model that makes predictions based on both the context and the information obtained from entity linking results. Experimental results on two commonly used datasets demonstrates the effectiveness of our approach. On both datasets, it achieves more than 5% absolute strict accuracy improvement over the state of the art.

1 Introduction

Given a piece of text and the span of an entity mention in this text, fine-grained entity typing (FET) is the task of assigning fine-grained type labels to the mention (Ling and Weld, 2012). The assigned labels should be context dependent (Gillick et al., 2014). For example, in the sentence "Trump threatens to pull US out of World Trade Organization," the mention "Trump" should be labeled as /person and /person/politician, although Donald Trump also had other occupations such as businessman, TV personality, etc.

This task is challenging because it usually uses a relatively large tag set, and some mentions may require the understanding of the context to be correctly labeled. Moreover, since manual annotation is very labor-intensive, existing approaches have to rely on distant supervision to train models (Ling and Weld, 2012; Ghaddar and Langlais, 2018).

Thus, the use of extra information to help with the classification process becomes very important. In this paper, we improve FET with entity linking (EL). EL is helpful for a model to make typing decisions because if a mention is correctly linked to its target entity, we can directly obtain the type information about this entity in the knowledge base (KB). For example, in the sentence "There were some great discussions on a variety of issues facing Federal Way," the mention "Federal Way" may be incorrectly labeled as a company by some FET models. Such a mistake can be avoided after linking it to the city Federal Way, Washington. For cases that require the understanding of the context, using entity linking results is also beneficial. In the aforementioned example where "Trump" is the mention, obtaining all the types of Donald Trump in the knowledge base (e.g., politician, businessman, TV personality, etc.) is still informative for inferring the correct type (i.e., politician) that fits the context, since they narrows the possible labels down.

However, the information obtained through EL should not be fully trusted since it is not always accurate. Even when a mention is correctly linked to an entity, the type information of this entity in the KB may be incomplete or outdated. Thus, in this paper, we propose a deep neural fine-grained entity typing model that flexibly predicts labels based on the context, the mention string, and the type information from KB obtained with EL.

Using EL also introduces a new problem for the training process. Currently, a widely used approach to create FET training samples is to use the anchor links in Wikipedia (Ling and Weld, 2012; Ren et al., 2016a). Each anchor link is regarded as a mention, and is weakly labeled with all the types of its referred entity (the Wikipedia page the anchor link points to) in KB. Our approach, when links the mention correctly, also uses all the types of the referred entity in KB as extra information. This may cause the trained model to overfit the weakly labeled data. We design a variant of the hinge loss and introduce noise during training to address this problem.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing
and the 9th International Joint Conference on Natural Language Processing, pages 6210–6215,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

We conduct experiments on two commonly used FET datasets. Experimental results show that introducing information obtained through entity linking and having a deep neural model both helps to improve FET performance. Our model achieves more than 5% absolute strict accuracy improvement over the state of the art on both datasets.

Our contributions are summarized as follows:

- We propose a deep neural fine-grained entity typing model that utilizes type information from KB obtained through entity linking.
- We address the problem that our model may overfit the weakly labeled data by using a variant of the hinge-loss and introducing noise during training.
- We demonstrate the effectiveness of our approach with experimental results on commonly used FET datasets.

Our code is available at `https://github.com/HKUST-KnowComp/IFETEL`.

2 Related Work

An early effort of classifying named entities into fine-grained types can be found in (Fleischman and Hovy, 2002), which only focuses on person names. Latter, datasets with larger type sets are constructed (Weischedel and Brunstein, 2005; Ling and Weld, 2012; Choi et al., 2018). These datasets are more preferred by recent studies (Ren et al., 2016a; Murty et al., 2018).

Most of the existing approaches proposed for FET are learning based. The features used by these approaches can either be hand-crafted (Ling and Weld, 2012; Gillick et al., 2014) or learned from neural network models (Shimaoka et al., 2017; Xu and Barbosa, 2018; Xin et al., 2018). Since FET systems usually use distant supervision for training, the labels of the training samples can be noisy, erroneous or overly specific. Several studies (Ren et al., 2016b; Xin et al., 2018; Xu and Barbosa, 2018) address these problems by separating clean mentions and noisy mentions, modeling type correction (Ren et al., 2016a), using a hierarchy-aware loss (Xu and Barbosa, 2018), etc.

(Huang et al., 2016) and (Zhou et al., 2018) are two studies that are most related to this paper. Huang et al. (2016) propose an unsupervised FET system where EL is an importat component. But they use EL to help with clustering and type name selection, which is very different from how we use it to improve the performance of a supervised FET model. (Zhou et al., 2018) finds related entities based on the context instead of directly applying EL. The types of these entities are then used for inferring the type of the mention.

3 Method

Let T be a predefined tag set, which includes all the types we want to assign to mentions. Given a mention m and its context, the task is to predict a set of types $\boldsymbol{\tau} \subset T$ suitable for this mention. Thus, this is a multi-class, multi-label classification problem (Ling and Weld, 2012). Next, we will introduce our approach for this problem in detail, including the neural model, the training of the model, and the entity linking algorithm we use.

3.1 Fine-grained Entity Typing Model

Input Each input sample to our FET system contains one mention and the sentence it belongs to. We denote $w_1, w_2, ..., w_n$ as the words in the current sentence, $w_{p_1}, w_{p_2}, ..., w_{p_l}$ as the words in the mention string, where n is the number of words in the sentence, $p_1, ..., p_l$ are the indices of the words in the mention string, l is the number of words in the mention string. We also use a set of pretrained word embeddings.

Our FET approach is illustrated in Figure 1. It first constructs three representations: *context representation*, *mention string representation*, and *KB type representation*. Note that the KB type representation is obtained from a knowledge base through entity linking and is independent of the context of the mention.

Context Representation To obtain the context representation, we first use a special token w_m to represent the mention (the token "[Mention]" in Figure 1). Then, the word sequence of the sentence becomes $w_1, ..., w_{p_l-1}, w_m, w_{p_l+1}, ..., w_n$. Their corresponding word embeddings are fed into two layers of BiLSTMs. Let $\boldsymbol{h}_m^1$ and $\boldsymbol{h}_m^2$ be the output of the first and the second layer of BiLSTMs for w_m, respectively. We use $\boldsymbol{f}_c = \boldsymbol{h}_m^1 + \boldsymbol{h}_m^2$ as the context representation vector.

Mention String Representation Let $\boldsymbol{x}_1, ..., \boldsymbol{x}_l$ be the word embeddings of the mention string words $w_{p_1}, ..., w_{p_l}$. Then the mention string representation $\boldsymbol{f}_s = (\sum_{i=1}^{l} \boldsymbol{x}_i)/l$.

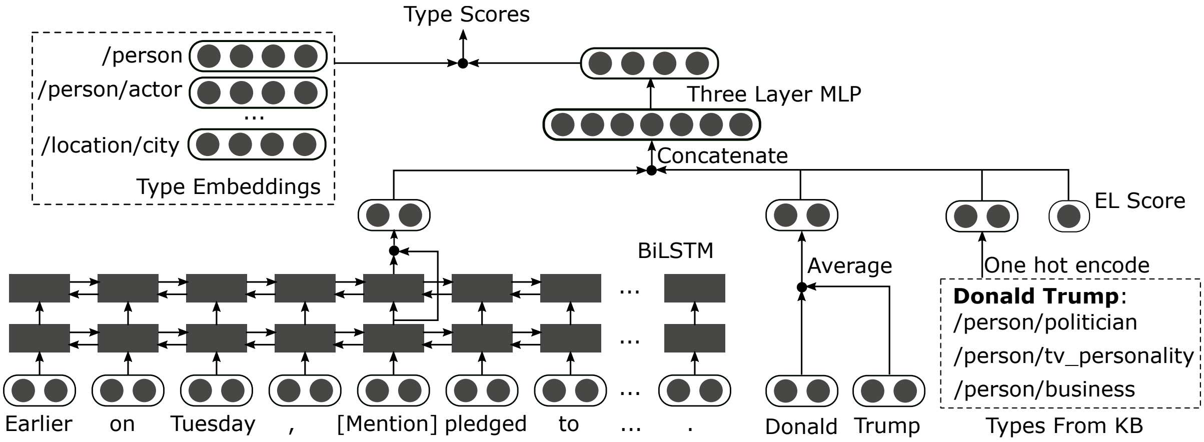

Figure 1: Our approach. The example sentence is "Earlier on Tuesday, *Donald Trump* pledged to help hard-hit U.S. farmers caught in the middle of the escalating trade war." Here, the correct label for the mention *Donald Trump* should be /person, /person/politician. "[Mention]" is a special token that we use to represent the mention.

KB Type Representation To obtain the KB type representation, we run an EL algorithm for the current mention. If the EL algorithm returns an entity, we retrieve the types of of this entity from the KB. We use Freebase as our KB[1]. Since the types in Freebase is different from T, the target type set, they are mapped to the types in T with rules similar to those used in (Zhou et al., 2018). Afterwards, we perform one hot encoding on these types to get the KB Type Representation $\boldsymbol{f}_e$. If the EL algorithm returns NIL (i.e., the mention cannot be linked to an entity), we simply one hot encode the empty type set.

Prediction Apart from the three representations, we also obtain the score returned by our entity linking algorithm, which indicates its confidence on the linking result. We denote it as a one dimensional vector $\boldsymbol{g}$. Then, we get $\boldsymbol{f} = \boldsymbol{f}_c \oplus \boldsymbol{f}_s \oplus \boldsymbol{f}_e \oplus \boldsymbol{g}$, where $\oplus$ means concatenation. $\boldsymbol{f}$ is then fed into an MLP that contains three dense layers to obtain $\boldsymbol{u}_m$, out final representation for the current mention sample m. Let $t_1, t_2, ..., t_k$ be all the types in T, where $k = |T|$. We embed them into the same space as $\boldsymbol{u}_m$ by assigning each of them a dense vector (Yogatama et al., 2015). These vectors are denoted as $\boldsymbol{t}_1, ..., \boldsymbol{t}_k$. Then the score of the mention m having the type $t_i \in T$ is calculated as the dot product of $\boldsymbol{u}_m$ and $\boldsymbol{t}_i$:

$$s(m, t_i) = \boldsymbol{u}_m \cdot \boldsymbol{t}_i. \quad (1)$$

We predict t_i as a type of m if $s(m, t_i) > 0$.

3.2 Model Training

Following existing studies, we also generate training data by using the *anchor links* in Wikipedia. Each anchor link can be used as a mention. These mentions are labeled by mapping the Freebase types of the target entries to the tag set T (Ling and Weld, 2012).

Since the *KB type representations* we use in our FET model are also obtained through mapping Freebase types, they will perfectly match the automatically generated labels for the mentions that are correctly linked (i.e., when the entity returned by the EL algorithm and the target entry of the anchor link are the same). For example, in Figure 1, suppose the example sentence is a training sample obtained from Wikipedia, where "Donald Trump" is an anchor link points to the Wikipedia page of *Donald Trump*. After mapping the Freebase types of *Donald Trump* to the target tag set, this sample will be weakly annotated as /person/politician, /person/tv_personality, and /person/business, which is exactly the same as the type information (the "Types From KB" in Figure 1) obtained through EL. Thus, during training, when the EL system links the mention to the correct entity, the model only needs to output the types in the *KB type representation*. This may cause the trained model to overfit the weakly labeled training data. For most types of entities such as locations and organizations, it is fine since they usually have the same types in different contexts. But it is problematic for person mentions, as their types can be context dependent.

To address this problem, during training, if a

[1] We use Freebase mainly because it is widely used by existing studies. Wikidata is an alternative.

mention is linked to a person entity by our entity linking algorithm, we add a random fine-grained person type label that does not belong to this entity while generating the *KB type representation*. For example, if the mention is linked to a person with types /person/actor and /person/author, a random label /person/politician may be added. This will force the model to still infer the type labels from the context even when the mention is correctly linked, since the *KB type representation* no longer perfectly match the weak labels.

To make it more flexible, we also propose to use a variant of the hinge loss used by (Abhishek et al., 2017) to train our model:

$$\begin{aligned} L = \sum_{m}[\sum_{t\in\tau_m} \max(0, 1 - s(m,t)) \\ + \sum_{t\in\bar{\tau}_m} \lambda(t)\max(0, 1 + s(m,t))] \end{aligned} \quad (2)$$

where τ_m is the correct type set for mention m, $\bar{\tau}_m$ is the incorrect type set. $\lambda(t) \in [1, +\infty)$ is a predefined parameter to impose a larger penalty if the type t is incorrectly predicted as positive. Since the problem of overfitting the weakly annotated labels is more severe for person mentions, we set $\lambda(t) = \lambda_P$ if t is a fine-grained person type, and $\lambda(t) = 1$ for all other types.

During training, we also randomly set the EL results of half of the training samples to be NIL. So that the model can perform well for mentions that cannot be linked to the KB at test time.

3.3 Entity Linking Algorithm

In this paper, we use a simple EL algorithm that directly links the mention to the entity with the greatest commonness score. Commonness (Pan et al., 2015; Medelyan and Legg, 2008) is calculated base on the anchor links in Wikipedia. It estimates the probability of an entity given only the mention string. In our FET approach, the commonness score is also used as the confidence on the linking result (i.e., the $\boldsymbol{g}$ used in the prediction part of Subsection 3.1). Within a same document, we also use the same heuristic used in (Ganea and Hofmann, 2017) to find coreferences of generic mentions of persons (e.g., "Matt") to more specific mentions (e.g., "Matt Damon").

We also tried other more advanced EL methods in our experiments. However, they do not improve the final performance of our model. Experimental results of using the EL system proposed in (Ganea and Hofmann, 2017) is provided in Section 4.

4 Experiments

4.1 Setup

We use two datasets: FIGER (GOLD) (Ling and Weld, 2012) and BBN (Weischedel and Brunstein, 2005). The sizes of their tag sets are 113 and 47, respectively. FIGER (GOLD) allows mentions to have multiple type paths, but BBN does not. Another commonly used dataset, OntoNotes (Gillick et al., 2014), is not used since it contains many pronoun and common noun phrase mentions such as "it," "he," "a thrift institution," which are not suitable to directly apply entity linking on.

Following (Ling and Weld, 2012), we generate weakly labeled datasets for training with Wikipedia anchor links. Since the tag sets used by FIGER (GOLD) and BBN are different, we create a training set for each of them. For each dataset, $2,000$ weakly labeled samples are randomly picked to form a development set. We also manually annotated 50 person mentions collected from news articles for tuning the parameter λ_P.

We use the 300 dimensional pretrained GloVe word vectors provided by (Pennington et al., 2014). The hidden layer sizes of the two layers of BiLSTMs are both set to 250. For the three-layer MLP, the size of the two hidden layers are both set to 500. The size of the type embeddings is 500. λ_P is set to 2.0. We also apply batch normalization and dropout to the input of each dense layer in our three-layer MLP during training.

We use strict accuracy, Macro F1, and Micro F1 to evaluate fine-grained typing performance (Ling and Weld, 2012).

4.2 Compared Methods

We compare with the following existing approaches: AFET (Ren et al., 2016a), AAA (Abhishek et al., 2017), NFETC (Xu and Barbosa, 2018), and CLSC (Chen et al., 2019).

We use **Ours (Full)** to represent our full model, and also compare with five variants of our own approach: **Ours (DirectTrain)** is trained without adding random person types while obtaining the KB type representation, and λ_P is set to 1; **Ours (NoEL)** does not use entity linking, i.e., the KB type representation and the entity linking confidence score are removed, and the model is trained in DirectTrain style; **Ours (NonDeep)** uses one BiLSTM layer and replaces the MLP with a dense layer; **Ours (NonDeep NoEL)** is the NoEL version of *Ours (NonDeep)*; **Ours (LocAttEL)** uses

Dataset	FIGER (GOLD)			BBN		
Approach	Accuracy	Macro F1	Micro F1	Accuracy	Macro F1	Micro F1
AFET	53.3	69.3	66.4	67.0	72.7	73.5
AAA	65.8	81.2	77.4	73.3	79.1	79.2
NFETC	68.9	81.9	79.0	72.1	77.1	77.5
CLSC	-	-	-	74.7	80.7	80.5
Ours (NonDeep NoEL)	65.9	81.7	78.0	69.3	81.4	81.5
Ours (NonDeep)	72.3	85.4	82.6	79.1	87.9	88.4
Ours (DirectTrain)	69.1	85.2	82.2	-	-	-
Ours (NoEL)	69.8	82.7	80.4	80.5	87.5	88.0
Ours (LocAttEL)	75.1	86.3	83.9	**82.8**	88.9	89.5
Ours (Full)	**75.5**	**87.1**	**84.6**	82.5	**89.2**	**89.6**

Table 1: Fine-grained entity typing performance. The performance of "Ours (DirectTrain)" on BBN is omitted since this dataset does not have fine-grained types for person.

the entity linking approach proposed in (Ganea and Hofmann, 2017) instead of our own commonness based approach. *Ours (Full)*, *Ours (DirectTrain)*, and *Ours (NonDeep)* all use our own commonness based entity linking approach.

4.3 Results

The experimental results are listed in Table 1. As we can see, our approach performs much better than existing approaches on both datasets.

The benefit of using entity linking in our approach can be verified by comparing *Ours (Full)* and *Ours (NoEL)*. The performance on both datasets decreases if the entity linking part is removed. Especially on FIGER (GOLD), the strict accuracy drops from 75.5 to 69.8. Using entity linking improves less on BBN. We think this is because of three reasons: 1) BBN has a much smaller tag set than FIGER (GOLD); 2) BBN does not allow a mention to be annotated with multiple type paths (e.g., labeling a mention with both /building and /location is not allowed), thus the task is easier; 3) By making the model deep, the performance on BBN is already improved a lot, which makes further improvement harder.

The improvement of our full approach over *Ours (DirectTrain)* on FIGER (GOLD) indicates that the techniques we use to avoid overfitting the weakly labeled data are also effective.

Ours (LocAttEL), which uses a more advanced EL system, does not achieve better performance than *Ours (Full)*, which uses our own EL approach. After manually checking the results of the two EL approaches and the predictions of our model on FIGER (GOLD), we think this is mainly because: 1) Our model also uses the context while making predictions. Sometimes, if it "thinks" that the type information provided by EL is incorrect, it may not use it. 2) The performances of different EL approaches also depends on the dataset and the types of entities used for evaluation. We find that on FIGER (GOLD), the approach in (Ganea and Hofmann, 2017) is better at distinguishing locations and sports teams, but it may also make some mistakes that our simple EL method does not. For example, it may incorrectly link "March," the month, to an entity whose Wikipedia description fits the context better. 3) For some mentions, although the EL system links it to an incorrect entity, the type of this entity is the same with the correct entity.

5 Conclusions

We propose a deep neural model to improve fine-grained entity typing with entity linking. The problem of overfitting the weakly labeled training data is addressed by using a variant of the hinge loss and introducing noise during training. We conduct experiments on two commonly used dataset. The experimental results demonstrates the effectiveness of our approach.

Acknowledgments

This paper was supported by the Early Career Scheme (ECS, No. 26206717) from Research Grants Council in Hong Kong and WeChat-HKUST WHAT Lab on Artificial Intelligence Technology.

References

Abhishek Abhishek, Ashish Anand, and Amit Awekar. 2017. Fine-grained entity type classification by jointly learning representations and label embeddings. In *Proceedings of EACL*, volume 1, pages 797–807.

Bo Chen, Xiaotao Gu, Yufeng Hu, Siliang Tang, Guoping Hu, Yueting Zhuang, and Xiang Ren. 2019. Improving distantly-supervised entity typing with compact latent space clustering. In *Proceedings of NAACL*, page 28622872.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of ACL*, pages 87–96.

Michael Fleischman and Eduard Hovy. 2002. Fine grained classification of named entities. In *Proceedings of COLING*, pages 1–7. Association for Computational Linguistics.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of EMNLP*, pages 2619–2629.

Abbas Ghaddar and Phillippe Langlais. 2018. Transforming wikipedia into a large-scale fine-grained entity type corpus. In *Proceedings of LREC*.

Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. Context-dependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*.

Lifu Huang, Jonathan May, Xiaoman Pan, and Heng Ji. 2016. Building a fine-grained entity typing system overnight for a new x (x= language, domain, genre). *arXiv preprint arXiv:1603.03112*.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *Proceedings of AAAI*, volume 12, pages 94–100.

Olena Medelyan and Catherine Legg. 2008. Integrating cyc and wikipedia: Folksonomy meets rigorously defined common-sense. *Wikipedia and Artificial Intelligence: An Evolving Synergy*, page 13.

Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. 2018. Hierarchical losses and new resources for fine-grained entity typing and linking. In *Proceedings of ACL*, volume 1, pages 97–109.

Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. 2015. Unsupervised entity linking with abstract meaning representation. In *Proceedings of NAACL*, pages 1130–1139.

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

Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. 2016a. AFET: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *Proceedings of EMNLP*, pages 1369–1378.

Xiang Ren, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, and Jiawei Han. 2016b. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of SIGKDD*, pages 1825–1834. ACM.

S Shimaoka, P Stenetorp, K Inui, and S Riedel. 2017. Neural architectures for fine-grained entity type classification. In *Proceedings of EACL*, volume 15, pages 1271–1280. Association for Computational Linguistics.

Ralph Weischedel and Ada Brunstein. 2005. BBN pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia*.

Ji Xin, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2018. Improving neural fine-grained entity typing with knowledge attention. In *Proceedings of AAAI*.

Ji Xin, Hao Zhu, Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Put it back: Entity typing with language model enhancement. In *Proceedings of EMNLP*, pages 993–998.

Peng Xu and Denilson Barbosa. 2018. Neural fine-grained entity type classification with hierarchy-aware loss. In *Proceedings of NAACL-HLT*, volume 1, pages 16–25.

Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding methods for fine grained entity type classification. In *Proceedings of ACL*, volume 2, pages 291–296.

Ben Zhou, Daniel Khashabi, Chen-Tse Tsai, and Dan Roth. 2018. Zero-shot open entity typing as type-compatible grounding. In *Proceedings of EMNLP*, pages 2065–2076.