Put It Back: Entity Typing with Language Model Enhancement

Ji Xin[1,2], Hao Zhu[1], Xu Han[1], Zhiyuan Liu[1]*, Maosong Sun[1]
[1]State Key Laboratory on Intelligent Technology and System
Beijing National Research Center for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, China
[2]David R. Cheriton School of Computer Science, University of Waterloo, Canada
ji.xin@uwaterloo.ca
{zhuhao15,hanxu17}@mails.tsinghua.edu.cn
{liuzy,sms}@tsinghua.edu.cn

Abstract

Entity typing aims to classify semantic types of an entity mention in a specific context. Most existing models obtain training data using distant supervision, and inevitably suffer from the problem of noisy labels. To address this issue, we propose entity typing with language model enhancement. It utilizes a language model to measure the compatibility between context sentences and labels, and thereby automatically focuses more on context-dependent labels. Experiments on benchmark datasets demonstrate that our method is capable of enhancing the entity typing model with information from the language model, and significantly outperforms the state-of-the-art baseline. Code and data for this paper can be found from https://github.com/thunlp/LME.

1 Introduction

Entity typing classifies semantic types of an entity mention in a context sentence, and can be beneficial for a large number of natural language processing tasks (Neelakantan and Chang, 2015), such as entity linking (Chabchoub et al., 2016), relation extraction (Miwa and Sasaki, 2014), and question answering (Yahya et al., 2013). Fine-grained entity typing (FET) (Ling and Weld, 2012; Yosef et al., 2012; Yao et al., 2013; Gillick et al., 2014; Del Corro et al., 2015; Yogatama et al., 2015; Yaghoobzadeh and Schütze, 2015; Ren et al., 2016a; Yuan and Downey, 2018) is based on a large set of fine-grained types and is therefore more challenging. So far, neural models (Dong et al., 2015; Shimaoka et al., 2017; Xin et al., 2018) have achieved state-of-the-art results on FET.

All current FET models rely on distant supervision (DS) (Mintz et al., 2009) to obtain training data, due to the lack of large-scale human-labeled data. Such reliance on DS has been a significant problem for entity typing. In DS, an entity mention in the context sentence is first linked to a named entity in the knowledge base (KB). The entity has type labels[1] stored in the KB, and all labels will be assigned to this entity mention. In other words, these are noisy global labels without considering the specific context of the entity mention. On the other hand, entity typing aims to predict context-dependent types of the entity mention, and test datasets are all human-labeled. The difference between DS and human annotation leads to a huge gap in performances between training/development and test dataset.[2]

Raw	**Schwarzenegger** was elected to be the governor. **Schwarzenegger** acted in the film *Terminator*.
Good	(A) `politician` was elected to be the governor. (An) `actor` acted in the film *Terminator*.
Bad	(An) `actor` was elected to be the governor. (A) `politician` acted in the film *Terminator*.

Table 1: Examples of entity mention—type name replacement.

To address this problem, we propose Entity Typing with **L**anguage **M**odel **E**nhancement (LME). It is able to measure the compatibility between the context sentence and each distantly supervised label, in an *unsupervised* manner using meaning of the label.

In previous works, the hierarchical structure of labels has been considered (Ren et al., 2016b; Karn et al., 2017; Xu and Barbosa, 2018). However, to the best of our knowledge, precious

[*] Corresponding author: Zhiyuan Liu.

[1] Since entities are classified into labels of types, *type* and *label* have the same meaning in this paper.

[2] In the WIKI dataset, strict accuracies and macro-F1 scores are respectively 72.3%/89.2% on the development set and 59.7%/79.0% on the test set, using the model NFGEC from (Shimaoka et al., 2017).

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 993–998
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

information inside names of labels has never been used. For example, whether the label is `/person/actor` or `/foo/bar` makes no difference. We argue that, the meaning of entity mention words can also be expressed by the name of its **context-dependent type**, to some extent. Based on this argument, replacements with context-dependent types make more sense than those with global-but-context-irrelevant ones. We provide an example in Table 1. The entity *Schwarzenegger* has types `/actor` and `/politician`, and we can see that replacements with context-dependent types produce better sentences.

The natural way to evaluate the soundness of sentences is language modeling (Bengio et al., 2003; Mikolov et al., 2010). Our method employs a language model to evaluate the soundness of each synthetic sentence generated by replacing the entity mention with its type's name. It is able to focus more on context-dependent types.

We conduct experiments to compare our model with the state-of-the-art baseline on two widely used datasets. The results demonstrate that LME is capable of improving entity typing systems by considering the meaning of labels, and alleviating the problem of noise in distantly-supervised entity typing.

2 Model

Our model (Figure 1) consists of two parts: an entity typing (ET) module, and a language model enhancement (LME) module.

ET predicts a probability distribution vector $\mathbf{y}$ for an entity mention, where each entry $\mathbf{y}_i$ represents the predicted probability for each type label.

In the training phase, LME optimizes a language model whose input includes $\mathbf{y}$, and also back-propagates gradients through $\mathbf{y}$ to parameters inside ET. In the testing phase, LME is **not** involved and $\mathbf{y}$ is directly used for inference: if $\mathbf{y}_i$ is greater than a threshold 0.5, the i$^{\text{th}}$ type is considered true; if all entries are below the threshold, the type with the greatest entry is considered true.

2.1 Entity Typing Module

Entity typing is defined on an ontology $\mathcal{T}$ (the set of all labels). Given an entity mention e and its context sentence $s = \{l_1, l_2, ..., e, r_1, r_2, ...\}$ (l_i and r_i are left and right context words), the typing model predicts a vector $\mathbf{y}$ indicating the probability distribution over all labels in the ontology:

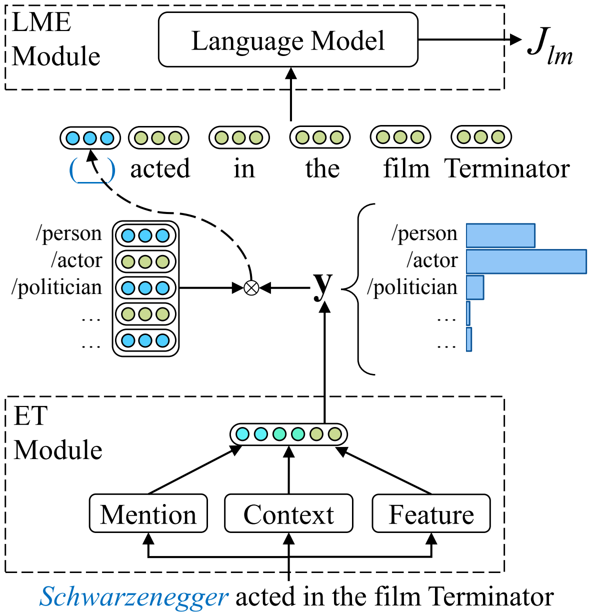

Figure 1: Our model: an entity typing (ET) module and a language model enhancement (LME) module.

$$\mathbf{y} = \sigma(\mathbf{W}_y \ [\mathbf{v}_M; \mathbf{v}_C; \mathbf{v}_F]), \tag{1}$$

where σ is the sigmoid function, $\mathbf{W}_y$ is a parameter matrix, and $[;;]$ denotes concatenation. Three vectors: **M**ention, **C**ontext and **F**eature, are built from e and s as follows:

Entity mention vector There may be multiple words $e_1, e_2, ...$ in the entity mention, and $\mathbf{v}_M$ is the average of word embeddings of these words.

Context vector Two bi-directional LSTMs (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997) are used for left and right context words. The outputs of BiLSTMs further go through a self-attention layer. $\mathbf{v}_C$ is the concatenation of the attention-layer outputs.

Hand-crafted feature vector A sparse feature vector $\mathbf{f}$ is built from the entity mention e. The features are adopted from those used by Gillick et al. (2014) and Yogatama et al. (2015). $\mathbf{v}_F$ is a dense projection of $\mathbf{f}$:

$$\mathbf{v}_F = \mathbf{W}_f \mathbf{f}, \tag{2}$$

where $\mathbf{W}_f$ is the projection matrix.

After $\mathbf{y}$ is calculated, DS provides a label vector $\mathbf{y}^* \in \{0, 1\}^{|\mathcal{T}|}$, where $|\mathcal{T}|$ is the number of labels. The loss function for typing is the cross-entropy between $\mathbf{y}$ and $\mathbf{y}^*$:

$$\begin{aligned} J_{type} &= H(\mathbf{y}^*, \mathbf{y}) \\ &= -\textstyle\sum_i y_i^* \log(y_i) + (1 - y_i^*) \log(1 - y_i), \end{aligned} \tag{3}$$

2.2 Language Model Enhancement Module

The core part of the LME module is an LSTM language model (Sundermeyer et al., 2012). The language model takes a sentence $\{w_1, w_2, ..., w_n\}$ as input, and assigns a probability to this sentence. Concretely, at step i, the LSTM reads the word sub-sequence $\{w_1, ..., w_i\}$, and predicts the probability of w_{i+1} succeeding the sub-sequence. A well trained language model predicts high probability for a reasonable sentence.

Before applying the LME module to enhance the ET module, the language model is pre-trained with sentences from the training set. The loss function for s in the pre-train phase is:

$$J_{pre} = \texttt{LM}(\{\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{e}, \mathbf{r}_1, \mathbf{r}_2, ...\}), \tag{4}$$

where bold face letters are word embeddings for corresponding words. $\texttt{LM}(\cdot)$ is the language model loss function: accumulative step-wise log-probability of each word of the input sequence. A well-trained language model calculates smaller loss for a more reasonable sentence.

After pre-training the language model, the LME module is combined with the ET module. Concretely, we assign an embedding vector $\mathbf{L}_i$ for each label, and take the sum of label embeddings weighted by $\mathbf{y}$. The sum $\mathbf{h}$ replaces $\mathbf{e}$ in the input sequence of the language model:

$$\mathbf{h} = \textstyle\sum_{i=1}^{T} \mathbf{y}_i \mathbf{L}_i, \tag{5}$$

$$J_{lm} = \texttt{LM}(\{\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{h}, \mathbf{r}_1, \mathbf{r}_2, ...\}), \tag{6}$$

where $\mathbf{L}$ is the matrix of all label embeddings, and J_{lm} is loss function of the language model used in the training phase. In order to ensure that label embeddings are in the same semantic space with word embeddings, $\mathbf{L}$ is initialized with word embeddings of the labels' names.

In the training phase, parameters of the ET module are updated w.r.t.

$$J_{train} = J_{type} + \lambda J_{lm}, \tag{7}$$

where λ is the weight to balance the loss.

The ET module has a much smaller parameter space than the language model. In order to make full use of the gradients, we only update parameters of the ET module and **fix** the language model in the training phase. Now that the language model is fixed, when J_{lm} is being minimized, it adjusts the probability distribution in $\mathbf{y}$. If a label i is compatible with the context sentence, its corresponding entry $\mathbf{y}_i$ is expected to have a high value. Gradients are back-propagated through $\mathbf{y}$ and update parameters of the ET module. In this way, $\mathbf{y}$ can learn to be more context-dependent.

3 Experiments

3.1 Dataset

We employ two well-established and widely-used dataset for evaluating our model: WIKI (Ling and Weld, 2012) and ONTONOTES (Gillick et al., 2014).

Training parts of both datasets are labeled with DS, and testing parts are annotated by human. Therefore they are suitable for evaluating how our model can narrow the gap between DS and ground-truth context-dependent labels. Statistics of the two datasets are provided in Table 2.

Dataset	Train	Development	Test
WIKI	2,000,000	10,000	563
ONTONOTES	251,039	2,202	8,963

Table 2: Number of instances in each part of datasets.

3.2 Experiment Settings

The baseline for comparison is the hybrid model NFGEC proposed by Shimaoka et al. (2017). It is described as the ET module of our model. Our own model is referred to as NFGEC+LME.

We implement our model based on the source code of NFGEC.[3] For a fair comparison, the ET module is unchanged, including all hyperparameters and methods of parameter random initialization. Word embeddings are initialized with pre-trained embeddings provided by Pennington et al. (2014).

There are a few additional hyperparameters in our model. The most important one is λ, the weight between two parts of the loss function. Other ones include the learning rate r for pretraining the language model and the hidden size h of LSTM used in the language model. We perform

[3] https://github.com/shimaokasonse/NFGEC

a grid-search based on performances on the development set, and set $r = 0.005$ and $h = 500$. Details of λ will be discussed in Section 3.4.

3.3 Overall Results

We compare vanilla NFGEC and NFGEC+LME in Table 3. The results of NFGEC come from the paper by Shimaoka et al. (2017). For running NFGEC+LME, λ is set to 0.005 in WIKI and 0.001 in ONTONOTES.

Evaluation metrics include strict accuracy, macro-F1 score and micro-F1 score (Ling and Weld, 2012).

Dataset	WIKI			ONTONOTES		
Metric	**Strict**	Macro	Micro	**Strict**	Macro	Micro
NFGEC	59.68	78.97	75.36	50.89	70.80	64.93
+LME	62.88	80.61	76.95	52.90	72.41	65.17

Table 3: Performance of entity typing, evaluated by strict accuracy, macro-F1 and micro-F1 score. (%)

From the results we see that:

(1) In both datasets, LME consistently helps NFGEC to better classify entity mentions into their context-dependent types. We can see improvements in all metrics. This is because LME is capable of evaluating the appropriateness of each label and distinguishing context-dependent ones from global-but-context-irrelevant ones. Therefore LME helps the system to focus on more reasonable types.

(2) Among all metrics, the improvement on strict accuracy is the most significant. Strict accuracy is the proportion of entity mentions whose predicted types are completely identical with human annotation. It is therefore the most important metric for evaluating how robust the system is against noisy labels. The ability of LME alleviating noises from DS contributes to improving strict accuracy most.

3.4 Analysis of λ

We choose the optimal λ values for results in Table 3 according to their performances on the *development* set. After they are chosen, we compare the results on the *test* set under different values of λ in Figure 2.

Conclusions from the previous subsection can be seen again: when λ is set to a proper value, our model can consistently outperform the baseline over all metrics; strict accuracy is the metric with the most significant improvements.

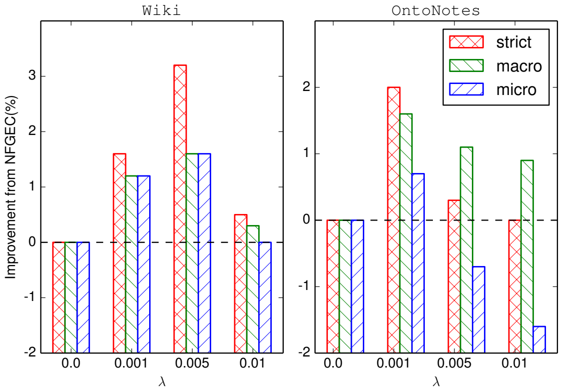

Figure 2: The performance under different λ values. $\lambda = 0$ is the vanilla NFGEC. Note that values of the vertical axis are improvements compared to vanilla NFGEC.

Also, we notice that the performances deteriorate when λ grows too large, and may even be worse than the baseline. The reason is that LME is a kind of *regularization*: its role is only in the training phase, exchanging the performance on training set with that on test set. So λ, as a regularization coefficient, must be carefully chosen.

3.5 Qualitative Analysis

In order to have an intuitive feeling of the model, we provide an example of LME's effect.

In the following sentence (from the test set of WIKI), both models try to predict the type of **Lake Placid** which, in this very context, is a town in New York. We show all labels with at least one score over the threshold 0.5, or is annotated true by human in Table 4.

> *Scaringe dismissed Brian Barrett of* ***Lake Placid*** *as one of his defense attorneys.*

Type	NFGEC	+LME	Human
/person	0.622	0.150	False
/location	0.323	0.627	True
/location/city	0.024	0.208	True

Table 4: An example, showing the scores by two models as well as human annotation.

NFGEC predicts a high score for /person and a low score for /location, probably because both words of the entity mention are first-letter capitalized and thus look like a person's name. LME, however, may consider the sentence structure *person of location* to be more reasonable than *person of person*, and makes the correct judgment between these two labels. As for `/location/city`, LME also shows higher confidence than NFGEC, but it is still regretfully below the threshold. This also demonstrates a weakness of LME: limited by the performance of the ET module. Addressing this limitation can be considered as a future direction for improvement.

4 Conclusion

In this paper, we propose a novel architecture LME to improve entity typing systems. It utilizes a language model and a set of label embeddings to judge the compatibility between labels and context sentences, and reduces noises introduced by DS. Experiments demonstrate that LME is capable of helping NFGEC, a state-of-the-art entity typing model, to alleviate the problem of noisy labels, and reaching a new state-of-the-art performance. Since the LME module does not depend on the ET module, we are confident that LME can be adapted to other entity typing systems as well.

Future Work Utilizing meaning of labels to alleviate the problem of noises from DS is an interesting direction. We make the first attempt in this paper, and we believe the direction is worth further exploring. For example, (1) how to train a language model that is sensitive with incorrect labels; (2) how to combine meaning of labels with the hierarchical structure of types; (3) how to find the optimal λ easily for a new dataset. LME may also be extended to other tasks that also suffer from noises and incompleteness of DS, such as relation extraction (Takamatsu et al., 2012; Ritter et al., 2013; Lin et al., 2016). However, since a relation does not have a specific location in the sentence, it needs more effort than a simple replacement.

Acknowledgment

This work is supported by the 973 Program (No. 2014CB340501), the National Natural Science Foundation of China (NSFC No. 61572273, 61661146007) and Tsinghua University Initiative Scientific Research Program (20151080406). This work is also funded by China Association for Science and Technology (2016QNRC001). We also thank anonymous reviewers for their insightful suggestions.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *JMLR*.

Mohamed Chabchoub, Michel Gagnon, and Amal Zouaq. 2016. Collective disambiguation and semantic annotation for entity linking and typing. In *Proceedings of SWEC*.

Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. 2015. Finet: Context-aware fine-grained named entity typing. In *Proceedings of EMNLP*.

Li Dong, Furu Wei, Hong Sun, Ming Zhou, and Ke Xu. 2015. A hybrid neural model for type classification of entity mentions. In *Proceedings of IJCAI*.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*.

Sanjeev Karn, Ulli Waltinger, and Hinrich Schütze. 2017. End-to-end trainable attentive decoder for hierarchical entity classification. In *Proceedings of EACL*.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *Proceedings of AAAI*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of INTERSPEECH*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL*.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of EMNLP*.

Arvind Neelakantan and Ming-Wei Chang. 2015. Inferring missing entity type instances for knowledge base completion: New dataset and methods. *Proceedings of NAACL*.

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

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*.

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 KDD*.

Alan Ritter, Mausam, Luke Zettlemoyer, and Oren Etzioni. 2013. Modeling missing data in distant supervision for information extraction. *TACL*.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017. Neural architectures for fine-grained entity type classification. In *Proceedings of EACL*.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Proceedings of INTERSPEECH*.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2012. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of ACL*.

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

Peng Xu and Denilson Barbosa. 2018. Neural fine-grained entity type classification with hierarchy-aware loss. In *Proceedings of NAACL*.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2015. Corpus-level fine-grained entity typing using contextual information. In *Proceedings of EMNLP*.

Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, and Gerhard Weikum. 2013. Robust question answering over the web of linked data. In *Proceedings of CIKM*.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2013. Universal schema for entity type prediction. In *Proceedings of Workshop on AKBC*.

Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding methods for fine grained entity type classification. In *Proceedings of ACL*.

Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012. HYENA: Hierarchical type classification for entity names. In *Proceedings of COLING*.

Zheng Yuan and Doug Downey. 2018. Otyper: A neural architecture for open named entity typing. In *Proceedings of AAAI*.