An Attentive Fine-Grained Entity Typing Model with Latent Type Representation

Ying Lin and Heng Ji
Computer Science Department
University of Illinois at Urbana-Champaign, Urbana, IL, USA 61802
{yinglin8,hengji}@illinois.edu

Abstract

We propose a fine-grained entity typing model with a novel attention mechanism and a hybrid type classifier. We advance existing methods in two aspects: feature extraction and type prediction. To capture richer contextual information, we adopt contextualized word representations instead of fixed word embeddings used in previous work. In addition, we propose a two-step mention-aware attention mechanism to enable the model to focus on important words in mentions and contexts. We also present a hybrid classification method beyond binary relevance to exploit type interdependency with latent type representation. Instead of independently predicting each type, we predict a low-dimensional vector that encodes latent type features and reconstruct the type vector from this latent representation. Experiment results on multiple data sets show that our model significantly advances the state-of-the-art on fine-grained entity typing, obtaining up to 6.6% and 5.5% absolute gains in macro averaged F-score and micro averaged F-score respectively.

1 Introduction

Fine-grained entity typing aims to assign one or more types to each entity mention given a certain context. For example, in the following sentence, “If Rogers is in the game, the Huskies will be much better equipped to match the Cougars in that aspect”, the mention “Rogers” should be labeled as athlete in addition to person according to the context (e.g., game, Huskies). These fine-grained entity types are proven to be effective in supporting a wide range of downstream applications such as relation extraction (Yao et al., 2010), question answering (Lin et al., 2012), and coreference resolution (Recasens et al., 2013).

1Code for this paper is available at: https://github.com/limteng-rpi/fet.

Fine-grained entity typing is usually formulated as a multi-label classification problem. Previous approaches (Ling and Weld, 2012; Choi et al., 2018; Xin et al., 2018) typically address it with binary relevance that decomposes the problem into isolated binary classification subproblems and independently predicts each type. However, this method is commonly criticized for its label independence assumption, which is not valid for fine-grained entity typing. For example, if the model is confident at predicting the type artist, it should promote its parent type person but discourage organization and its descendant types. In order to capture inter-dependencies between types, we propose a hybrid model that incorporates latent type representation in addition to binary relevance. Specifically, the model learns to predict a low-dimensional vector that encodes latent type features obtained through Principle Label Space Transformation (Tai and Lin, 2012) and reconstruct the sparse and high-dimensional type vector from this latent representation.

Another major challenge in fine-grained entity typing is to differentiate similar types, such as director and actor, which requires the model to capture slightly different nuances in texts. Previous neural models (Shimaoka et al., 2016; Xin et al., 2018; Choi et al., 2018; Xu and Barbosa, 2018) generally extract features from pre-trained word embeddings. Instead, we adopt contextualized word representations (Peters et al., 2018), which can capture context-aware word semantics and better represent out-of-vocabulary words. We further propose a two-step attention mechanism to actively extract the most relevant information from the sentence. Particularly, we calculate the attention for context words in a mention-aware manner, allowing the model to focus on different parts of the sentence for different mentions. For example, in the following sentence, “In 2005 two fed-
eral agencies, the US Geological Survey and the Fish and Wildlife Service, began to identify fish in the Potomac and tributaries...”, the model should use “federal agencies” to help classify “US Geological Survey” and “Fish and Wildlife Service” as government agencies, but focus on “fish” and “tributaries” to determine that “Potomac” should be a body of water (river) instead of a city.

2 Methodology

Figure 1 illustrates our fine-grained entity typing framework. We represent the input sentence using pre-trained contextualized word representations. Next, we apply a two-step mention-aware attention mechanism to extract the most relevant features from the sentence to form the feature vector. On top of the model, we employ a hybrid classifier to predict the types of each mention.

2.1 Sentence Encoder

Contextual information plays a key role as we often need to determine the types especially subtypes according to the context. Hence, unlike previous neural models that generally use fixed word embeddings, we employ contextualized word representations (ELMo, Peters et al. 2018) that can capture word semantics in different contexts. Furthermore, because ELMo takes as input characters instead of words, it can better represent out-of-vocabulary words that are prevalent in entity mentions by leveraging sub-word information. Given a sentence of $S$ words, the encoder generates a sequence of word vectors \{ $r_1, ..., r_S$ \}, where $r_i \in \mathbb{R}^{d_r}$ is the representation of the $i$-th word.

2.2 Mention Representation

Previous attentive models (Shimaoka et al., 2017; Xu and Barbosa, 2018; Xin et al., 2018; Choi et al., 2018) only apply attention mechanisms to the context. However, some words in an entity mention may provide more useful information for typing, such as “Department” in Figure 1. To allow the model to focus on more informative words, we represent a mention $m$ consisting of $M$ words as a weighted sum of its contextualized word representations with an attention mechanism (Bahdanau et al., 2015) as

$$m = \sum_{i}^{M} a_i^m r_i,$$

where the attention score $a_i^m$ is computed as

$$a_i^m = \text{Softmax}(e_i^m) = \frac{\exp (e_i^m)}{\sum_{k}^{M} \exp (e_k^m)},$$

$$e_i^m = v^T \tanh(W^m r_i),$$

where parameters $W^m \in \mathbb{R}^{d_a \times d_r}$ and $v^m \in \mathbb{R}^{d_a}$ are learned during training, and the hidden attention dimension $d_a$ is set to $d_r$ in our experiments.

2.3 Context Representation

Given the context of mention $m$, we form its representation from involved contextualized word vectors with a mention-aware attention mechanism

$$c = \sum_{i}^{C} a_i^c r_i,$$

where $C$ is the number of contextual words, and $a_i^c$ is defined as $a_i^c = \text{Softmax}(e_i^c)$, where

$$e_i^c = v^c \tanh(W^c(r_i \oplus m \oplus p_i)),$$

where $\oplus$ represents concatenation, and $v^c \in \mathbb{R}^{d_a}$ and $W^c \in \mathbb{R}^{d_a \times (2d_r + 1)}$ are trainable parameters. We introduce a relative position term $p_i$ to indicate the distance from the $i$-th word to the mention as

$$p_i = \left(1 - \mu \left(\min(|i - a|, |i - b|) - 1\right)\right)^+,$$

Finally, the output of the model is determined by a hybrid classifier that takes both contextualized word representations and mention-aware representations as its input.
where \( a \) and \( b \) are indices of the first and last words of the mention, and \( \mu \) is set to 0.1.

Finally, the feature vector of mention \( m \) is formed by concatenating its mention representation \( m \) and context representation \( c \).

### 2.4 Hybrid Classification Model

We propose a hybrid type classification model consisting of two classifiers as Figure 1 shows. We first learn a matrix \( W^b \in \mathbb{R}^{d_l \times 2d_r} \) to predict type scores by

\[
\hat{y}^b = W^b (m \oplus c),
\]

where \( \hat{y}^b \) is the score for the \( i \)-th type and \( d_i \) is the number of types. However, this method independently predicts each type and does not consider their inter-dependencies. To tackle this issue, we introduce an additional classifier inspired by Principle Label Space Transformation (Tai and Lin, 2012). Under the hypercube sparsity assumption that the number of training examples is much smaller than \( 2^{d_i} \), Tai and Lin (2012) project high-dimensional type vectors into a low-dimensional space to find underlying type correlations behind the first order co-occurrence through Singular Value Decomposition (SVD)

\[
Y \approx \tilde{Y} = U \Sigma L^T,
\]

where \( U \in \mathbb{R}^{d_i \times d_i} \), \( \Sigma \in \mathbb{R}^{d_i \times d_i} \), \( L \in \mathbb{R}^{N \times d_i} \), and \( d_l \ll d_i \). This low-dimensional space is similar to the hidden concept space in Latent Semantic Analysis (Deerwester et al., 1990). The \( i \)-th row of \( L \) is the latent representation of the \( i \)-th type vector. After that, we learn to predict the latent type representation from the feature vector using

\[
l = V^l (m \oplus c),
\]

where \( V^l \in \mathbb{R}^{2d_r \times d_i} \) is trainable. We then reconstruct the type vector from \( l \) using a linear projection

\[
\tilde{y}^l = W^l l = U \Sigma l.
\]

Next, by combining scores from both classifiers, we have

\[
\tilde{y} = \sigma(W^b (m \oplus c) + \gamma W^l l),
\]

where \( \gamma \) is a scalar initialized to 0.1 and updated during training. Finally, our training objective is to minimize the following cross-entropy-based loss function

\[
J(\theta) = - \frac{1}{N} \sum_{i} y_i \log \tilde{y}_i + (1 - y_i) \log(1 - \tilde{y}_i).
\]

In the test phase, we predict each type with a probability \( \tilde{y}_i > 0.5 \) or \( \arg \max \tilde{y}_i \) if all probabilities are lower than 0.5.

### 3 Experiments

#### 3.1 Data Sets

In our experiments, we evaluate the proposed model on the following data sets.

**OntoNotes** fine-grained entity typing data set is derived from the OntoNotes corpus (Weischedel et al., 2013) and annotated by Gillick et al. (2014) using a three-layer set of 87 types. We use the augmented data set created by Choi et al. (2018).

**FIGER** (Ling and Weld, 2012) contains 2.7 million automatically labeled training instances from Wikipedia and 434 manually annotated sentences from news reports. We use ground truth mentions in our experiments and sample 0.1M training instances as the development set.

**KNET** (Xin et al., 2018) is another data set derived from Wikipedia. It consists of an automatically annotated subset (WIKI-AUTO) and a manually annotated (WIKI-MAN) test set.

**BBN Pronoun Coreference and Entity Type Corpus** (BBN, Weischedel and Brunstein, 2005) annotates 2,311 Wall Street Journal articles in Treebank-2 (LDC95T7) with fine-grained entity types. We use the version processed by Ren et al. (2016a).

<table>
<thead>
<tr>
<th>Data set</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
<th>Label</th>
<th>Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>OntoNotes</td>
<td>3.4M</td>
<td>2,202</td>
<td>8,963</td>
<td>87</td>
<td>3</td>
</tr>
<tr>
<td>FIGER</td>
<td>2.7M</td>
<td>-</td>
<td>563</td>
<td>113</td>
<td>2</td>
</tr>
<tr>
<td>KNET (Wiki-Auto)</td>
<td>1M</td>
<td>0.1M</td>
<td>0.1M</td>
<td>74</td>
<td>2</td>
</tr>
<tr>
<td>KNET (Wiki-Man)</td>
<td>-</td>
<td>-</td>
<td>100</td>
<td>74</td>
<td>2</td>
</tr>
<tr>
<td>BBN</td>
<td>32,739</td>
<td>-</td>
<td>6,430</td>
<td>56</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1: Data set statistics: Numbers of train/dev/test instances, label set size, max hierarchy depth.

#### 3.2 Experimental Setup

We use the pre-trained original-5.5b ELMo model\(^2\) and freeze its weights during training. We use an Adam optimizer with learning rate of 5e-5, L2 weight decay of 0.01, warmup rate of 0.1, and linear learning rate decay. We use a minibatch size of 200. To reduce overfitting, we apply dropout (Srivastava et al., 2014) to the word representation, relative position term, and the final feature vector with probability 0.5, 0.2, and 0.2.

We evaluate the performance by strict accuracy (Acc), macro-average F-score (Macro F1), and micro-average F-score (Micro F1) (Ling and Weld, 2012).

\(^2\)https://allennlp.org/elmo
3.3 Evaluation Results

We compare the performance of our model with state-of-the-art methods on OntoNotes, FIGER, and KNET in Table 2, 3, and 4.

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc</th>
<th>Macro F1</th>
<th>Micro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shimaoka et al. (2016)</td>
<td>51.7</td>
<td>70.9</td>
<td>64.9</td>
</tr>
<tr>
<td>Ren et al. (2016b)</td>
<td>57.2</td>
<td>71.5</td>
<td>66.1</td>
</tr>
<tr>
<td>Choi et al. (2018)</td>
<td>59.5</td>
<td>76.8</td>
<td>71.9</td>
</tr>
<tr>
<td>Our Model</td>
<td>63.8</td>
<td>82.9</td>
<td>77.3</td>
</tr>
</tbody>
</table>

Table 2: Results on the OntoNotes test set. The first three methods use only KB-based supervision.

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc</th>
<th>Macro F1</th>
<th>Micro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ling and Weld (2012)</td>
<td>53.2</td>
<td>69.9</td>
<td>69.3</td>
</tr>
<tr>
<td>Yogatama et al. (2015)</td>
<td>54.5</td>
<td>74.8</td>
<td>71.6</td>
</tr>
<tr>
<td>Shimaoka et al. (2017)</td>
<td>54.5</td>
<td>74.8</td>
<td>71.6</td>
</tr>
<tr>
<td>+ hand-crafted</td>
<td>59.7</td>
<td>79.0</td>
<td>75.4</td>
</tr>
<tr>
<td>Our Model</td>
<td>62.9</td>
<td>83.0</td>
<td>79.8</td>
</tr>
</tbody>
</table>

Table 3: Results on the FIGER (Gold) test set.

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc</th>
<th>Macro F1</th>
<th>Micro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shimaoka et al. (2016)</td>
<td>42.8</td>
<td>72.4</td>
<td>74.9</td>
</tr>
<tr>
<td>KNET-MA</td>
<td>41.6</td>
<td>72.7</td>
<td>75.7</td>
</tr>
<tr>
<td>KNET-KA*</td>
<td>45.5</td>
<td>73.6</td>
<td>76.2</td>
</tr>
<tr>
<td>KNET-KAD*</td>
<td>47.2</td>
<td>74.9</td>
<td>77.9</td>
</tr>
<tr>
<td>Our Model</td>
<td>45.8</td>
<td>77.4</td>
<td>78.4</td>
</tr>
</tbody>
</table>

Table 4: Results on KNET test sets. KNET-KA and KNET-KAD use additional entity embeddings.

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc</th>
<th>Macro F1</th>
<th>Micro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shimaoka et al. (2016)</td>
<td>18.0</td>
<td>69.4</td>
<td>70.1</td>
</tr>
<tr>
<td>KNET-MA</td>
<td>26.0</td>
<td>71.2</td>
<td>72.1</td>
</tr>
<tr>
<td>KNET-KA*</td>
<td>23.0</td>
<td>71.1</td>
<td>71.7</td>
</tr>
<tr>
<td>KNET-KAD*</td>
<td>34.0</td>
<td>74.9</td>
<td>75.3</td>
</tr>
<tr>
<td>Our Model</td>
<td>29.0</td>
<td>77.6</td>
<td>75.3</td>
</tr>
</tbody>
</table>

Table 5: Results on the BBN test set. Our Model* is a variant of our model that uses Bert contextualized word representations.

We show results on the BBN dataset in Table 5. Our Model* is a variant where we replace ELMo embeddings with Bert (large cased model) contextualized word representations.

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc</th>
<th>Macro F1</th>
<th>Micro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shimaoka et al. (2017)</td>
<td>54.5</td>
<td>74.8</td>
<td>71.6</td>
</tr>
<tr>
<td>+ ELMo</td>
<td>59.3</td>
<td>81.7</td>
<td>79.0</td>
</tr>
<tr>
<td>+ latent</td>
<td>57.7</td>
<td>77.5</td>
<td>75.2</td>
</tr>
<tr>
<td>+ two-step attention</td>
<td>59.7</td>
<td>78.8</td>
<td>76.3</td>
</tr>
<tr>
<td>+ ELMo</td>
<td>62.9</td>
<td>83.0</td>
<td>79.8</td>
</tr>
</tbody>
</table>

Table 6: Ablation study on the FIGER (Gold) test set.

To evaluate the influence of individual components of our model, we conduct an ablation study as shown in Table 6. We implement a baseline model similar to (Shimaoka et al., 2017) (no hand-crafted features), the state-of-the-art on this data set. We observe that each component added to the model improves its performance.

We visualize mention and context attention in Figure 2. The first example shows the impact of mention attention. The baseline model mistakenly classifies the mention as location probably because “Asian” generally appears in location mentions, while our model successfully predicts organization by assigning higher weights to “Student Commission”. In Example #2 and #3, we compare context attention between “Terry Martino” and “APA”. Although they occur in the same sentence, our model is able to focus on different context words for different mentions with the mention-aware attention mechanism.

![Figure 2: Mention and context attention visualization.](image-url)

4 Related Work

As entity types are usually organized as a forest of hierarchies, several models are proposed to leverage this structure. In (Yosef et al., 2012), the authors build a set of classifiers based on the taxon-
omy of YAGO (Hoffart et al., 2013) and perform top-down hierarchical classification. Shimaoka et al. (2017) propose a hierarchical label encoding method to share parameters between types in the same hierarchy. Xu and Barbosa (2018) propose a hierarchy-aware loss function to reduce the penalty when predicted types are related. By contrast, our model automatically find type inter-dependency via matrix factorization and is able to capture inter-dependencies between types regardless of whether they are in the same hierarchy.

Attention mechanisms are widely used in neural fine-grained entity typing models (Shimaoka et al., 2016, 2017; Xu and Barbosa, 2018; Xin et al., 2018; Choi et al., 2018) to weight context words. We also apply it to mention words and introduce a position term to make attentions for context words more mention-aware.

5 Conclusions and Future Work

We propose an attentive architecture for fine-grained entity typing with latent type representation. Experiments on multiple data sets demonstrate that our model achieves state-of-the-art performance. In the future, we will further improve the performance of fine-grained types, which is still lower than that of general types due to less training instances and distant supervision noise. We also plan to utilize fine-grained entity typing results in more downstream applications, such as coreference resolution and event extraction.

Acknowledgement

This work was supported by the U.S. DARPA AIDA Program No. FA8750-18-2-0014, Air Force No. FA8650-17-C-7715, and U.S. ARL NS-CTA No. W911NF-09-2-0053. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily 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 governmental purposes notwithstanding any copyright annotation therein.

References


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 the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016).

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 the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD 2016).


