Ultra-Fine Entity Typing with Weak Supervision from a Masked Language Model

Hongliang Dai, Yangqiu Song
Department of CSE, HKUST
{hdai,yqsong}@cse.ust.hk

Haixun Wang
Instacart
haixun.wang@instacart.com

Abstract

Recently, there is an effort to extend fine-grained entity typing by using a richer and ultra-fine set of types, and labeling noun phrases including pronouns and nominal nouns instead of just named entity mentions. A key challenge for this ultra-fine entity typing task is that human annotated data are extremely scarce, and the annotation ability of existing distant or weak supervision approaches is very limited. To remedy this problem, in this paper, we propose to obtain training data for ultra-fine entity typing by using a BERT Masked Language Model (MLM). Given a mention in a sentence, our approach constructs an input for the BERT MLM so that it predicts context dependent hypernyms of the mention, which can be used as type labels. Experimental results demonstrate that, with the help of these automatically generated labels, the performance of an ultra-fine entity typing model can be improved substantially. We also show that our approach can be applied to improve traditional fine-grained entity typing after performing simple type mapping.

1 Introduction

Fine-grained entity typing (Ling and Weld, 2012) has been long studied in the natural language processing community as the extracted type information is useful for downstream tasks such as entity linking (Ling et al., 2015; Onoe and Durrett, 2020), relation extraction (Koch et al., 2014), coreference resolution (Onoe and Durrett, 2020), etc. Recently, ultra-fine entity typing (Choi et al., 2018) extends the effort to using a richer set of types (e.g., *person*, *actor*, *company*, *victim*) to label noun phrases including not only named entity mentions, but also pronouns and nominal nouns. This task directly uses type words or phrases as tags. Its tag set can contain more than 10,000 types. A challenge is that with the large type set, it is extremely difficult and

time-consuming for humans to annotate samples. As a result, most existing works use weak labels that are automatically generated (Ling and Weld, 2012; Choi et al., 2018; Lee et al., 2020).

There are two main approaches to obtaining weakly labeled training examples. One approach is to find the Wikipedia pages that correspond to entity mentions, which can be done by using hyperlinks to Wikipedia or applying entity linking. Then the entity types can be obtained from knowledge bases. The other approach is to directly use the head words of nominal mentions as ultra-fine type labels. For example, if a nominal mention is "a famous actor," then the head word "actor" can be used as its type label.

Several problems exist when using these weak labels for the ultra-fine typing task. First, in the dataset created by Choi et al. (2018), on average there are fewer than two labels (types) for each sample annotated through either entity linking or head word supervision. On the other hand, a human annotated sample has on average 5.4 labels. As a result, models trained from the automatically obtained labels have a low recall. Second, neither of the above approaches can create a large number of training samples for pronoun mentions. Third, it is difficult to obtain types that are highly dependent on the context. For example, in "I met the movie star Leonardo DiCaprio on the plane to L.A.," the type *passenger* is correct for "Leonardo DiCaprio." However, this type cannot be obtained by linking to knowledge bases.

In this paper, to alleviate the problems above, we propose an approach that combines hypernym extraction patterns (Hearst, 1992; Seitner et al., 2016) with a masked language model (MLM), such as BERT (Devlin et al., 2019), to generate weak labels for ultra-fine entity typing. Given a sentence that contains a mention, our approach adds a short piece of text that contains a "[MASK]" token into it

Input	Top Words for [MASK]
In late 2015, [MASK] such as **Leonardo DiCaprio** starred in The Revenant.	actors, stars, actor, directors, filmmakers
At some clinics, **they** and some other [MASK] are told the doctors don't know how to deal with AIDS, and to go someplace else.	patients, people, doctors, kids, children
Finkelstein says he expects the company to "benefit from some of the disruption faced by **our competitors** and any other [MASK] ."	company, business, companies, group, investors

Table 1: Examples of constructed BERT MLM inputs for obtaining weak entity typing labels. Entity mentions are in bold and underlined. The texts highlighted with blue background are not in the original sentences. They are inserted to create inputs for BERT. The right column lists the five most probable words predicted by a pretrained BERT-Base-Cased MLM.

to construct an input to BERT. Then, the pretrained MLM will predict the hypernyms of the mention as the most probable words for "[MASK]." These words can then be used as type labels. For example, consider the first example in Table 1. The original sentence is "In late 2015, Leonardo DiCaprio starred in The Revenant." We construct an input for the BERT MLM by inserting "[MASK] such as" before the mention "Leonardo DiCaprio." With this input, the pretrained BERT MLM predicts "actors," "stars," "actor," "directors," and "filmmakers" as the five most probable words for "[MASK]." Most of them are correct types for the mention after singularization. This approach can generate labels for different kinds of mentions, including named entity mentions, pronoun mentions, and nominal mentions. Another advantage is that it can produce labels that needs to be inferred from the context. This allows us to generate more context-dependent labels for each mention, such as *passenger*, *patient*, etc.

Then, we propose a method to select from the results obtained through different hypernym extraction patterns to improve the quality of the weak labels. We also use a weighted loss function to make better use of the generated labels for model training. Finally, we adopt a self-training step to further improve the performance of the model. We evaluate our approach with the dataset created by Choi et al. (2018), which to the best of our knowledge, is the only English ultra-fine entity typing dataset currently available. On this dataset, we achieve more than 4% absolute F1 improvement over the previously reported best result. Additionally, we also apply our approach to a traditional fine-grained entity typing dataset: Ontonotes (Gillick et al., 2014), where it also yields better performance than the state of the art.

Our contributions are summarized as follows.

- We propose a new way to generate weak labels for ultra-fine entity typing.

- We propose an approach to make use of the newly obtained weak labels to improve entity typing results.

- We conduct experiments on both an ultra-fine entity typing dataset and a traditional fine-grained entity typing dataset to verify the effectiveness of our method.

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

2 Related Work

The ultra-fine entity typing task proposed by Choi et al. (2018) uses a large, open type vocabulary to achieve better type coverage than the traditional fine-grained entity typing task (Ling and Weld, 2012) that uses manually designed entity type ontologies. There are only limited studies on this newly proposed task: A neural model introduced by (Onoe and Durrett, 2019) filters samples that are too noisy to be used and relabels the remaining samples to get cleaner labels. A graph propagation layer is introduced by (Xiong et al., 2019) to impose a label-relational bias on entity typing models, so as to implicitly capture type dependencies. Onoe et al. (2021) use box embeddings to capture latent type hierarchies. There is also some work on the applications of ultra-fine entity typing: Onoe and Durrett (2020) apply ultra-fine entity typing to learn entity representations for two downstream tasks: coreference arc prediction and named entity disambiguation.

The traditional fine-grained entity typing task (Ling and Weld, 2012; Yosef et al., 2012) is closely

related to ultra-fine entity typing. Automatic annotation (Ling and Weld, 2012; Gillick et al., 2014; Dai et al., 2020) is also commonly used in the studies of this task to produce large size training data. Many different approaches have been proposed to improve fine-grained entity typing performance. For example, denoising the automatically generated labels (Ren et al., 2016), taking advantage of the entity type hierarchies or type inter-dependencies (Chen et al., 2020; Murty et al., 2018; Lin and Ji, 2019), exploiting external resources such as the information of entities provided in knowledge bases (Jin et al., 2019; Dai et al., 2019; Xin et al., 2018), etc.

Our work is also related to recent studies (Petroni et al., 2019; Jiang et al., 2020; Zhang et al., 2020) that probe pretrained language models to obtain knowledge or results for target tasks. Different from them, we use the predictions produced by BERT as intermediate results that are regarded as weak supervision to train better models. (Zhang et al., 2020) also uses Hearst patterns to probe masked language models. However, they target at the entity set expansion task.

3 Methodology

Our methodology consists of two main steps. First, we obtain weak ultra-fine entity typing labels from a BERT masked language model. Second, we use the generated labels in model training to learn better ultra-fine entity typing models.

3.1 Labels from BERT MLM

Given a sentence and a mention of interest in the sentence, our goal is to derive the hypernym or the type of the mention using a BERT MLM. To do this, we insert into the sentence a few tokens to create an artificial Hearst pattern (Hearst, 1992). One of the inserted tokens is a special "[MASK]" token, which serves as the placeholder of the hypernym of the mention. As the BERT MLM predicts the "[MASK]" token, we derive the hypernyms of the mention.

Consider the first sentence in Table 1 as an example: "In late 2015, Leonardo DiCaprio starred in The Revenant." To find the hypernym or the type of "Leonardo DiCaprio", we insert three tokens to create a "such as" pattern: "In late 2015, [MASK] such as Leonardo DiCaprio starred in The Revenant." Applying the BERT MLM on the sentence, we derive hypernyms such as "actors,"

Pattern	F1
M and any other H	25.3
M and some other H	24.8
H such as M	20.7
such H as M	18.1
H including M	17.4
H especially M	11.5

Table 2: Hypernym extraction patterns. M denotes the hyponym; H denotes the hypernym. The F1 score is evaluated with the development set of the ultra-fine dataset (Choi et al., 2018) for the labels generated with the corresponding pattern.

"stars," "directors," "filmmakers." Table 1 shows a few more examples.

We consider the 63 Hearst-like patterns (Hearst, 1992) presented in (Seitner et al., 2016) that express a hypernym-hyponym relationship between two terms. Table 2 lists some of the patterns, wherein H and M denote a hypernym and a hyponym, respectively. For example, "M and some other H" can be used to match "Microsoft and some other companies."

The general procedure to use these patterns to create input samples for BERT MLM and obtain labels from its predictions is as follows. We first regard the mention as M. Then, we insert the rest of the pattern either before or after the mention, and we replace H with the special "[MASK]" token. After applying the BERT MLM on sentences with artificial Hearst patterns, we derive top k type labels from the prediction for "[MASK]." To drive these k labels, we first sigularize the most probable words that are plural. Then, remove those that are not in the type vocabulary of the dataset. Finally, use the most probable k different words as k labels. For example, if we want to obtain 3 labels, and the most probable words are "people," "actors," "celebrities," "famous," "actor," etc. Then the 3 labels should be *person*, *actor*, *celebrity*. Because "actor" is the singluar form of "actors," and "famous" is not in the type vocabulary.

We show the performance of our method for obtaining 10 type labels for each mention with different patterns in Table 2. A pre-trained BERT-Base-Cased MLM is used to obtain the results[1].

For nominal mentions, directly applying the patterns that starts with "M" with the above procedure

---

[1] We use the pretrained model provided in the Transformers library. We also tried using BERT-Large and RoBERTa models. However, they do not yield better performance.

may sometimes be problematic. For example, consider the noun phrase "the factory in Thailand" as a mention. If we use the "M and some other H" pattern and insert "and other [MASK]" after the mention, the BERT MLM will predict the type *country* for Thailand instead of for the entire mention. To avoid such errors, while applying patterns that starts with "M" for nominal mentions, we regard the head word of the mention as M instead.

A more subtle and challenging problem is that the quality of the type labels derived from different patterns for different mentions can be very different. For example, for the mention "He" in sentence "He has won some very tough elections and he's governor of the largest state," the pattern "H such as M" leads to *person*, *election*, *time*, *thing*, *leader* as the top five types. But using the pattern "M and any other H," we get *candidate*, *politician*, *man*, *person*, *governor*. On the other hand, for mention "the Al Merreikh Stadium" in "It was only Chaouchi's third cap during that unforgettable night in the Al Merreikh Stadium," the results of using "H such as M" (the top five types are *venue*, *place*, *facility*, *location*, *area*) is better than using "M and any other H" (the top five types are *venue*, *stadium*, *game*, *match*, *time*).

To address the above problem, we do not use a same pattern for all the mentions. Instead, for each mention, we try to select the best pattern to apply from a list of patterns. This is achieved by using a baseline ultra-fine entity typing model, *BERT-Ultra-Pre*, which is trained beforehand without using labels generated with our BERT MLM based approach. Details of *BERT-Ultra-Pre* can be found in Section 5.2. Denote the pattern list as L. With each pattern in L, we can apply it on the given mention to derive a set of labels from the BERT MLM. Then, we find the set of labels that have the most overlap with the labels predicted by *BERT-Ultra-Pre*. Finally, the given mention is annotated with this set of labels.

It is not necessary to use all the patterns in (Seitner et al., 2016). To construct L, the list of patterns used for annotation, we perform the following procedure.

Step 1: Initialize L to contain the best performing pattern (i.e., "M and any other H") only.

Step 2: From all the patterns not in L, find the one that may bring the greatest improvement in F1 score if it is added to L.

Step 3: Add the pattern found in Step 2 to the L if the improvement brought by it is larger than a threshold.

Step 4: Repeat steps 2-3 until no patterns can be added.

Discussion on Type Coverage Since we only use one [MASK] token to generate labels, the model cannot produce multi-word types (e.g., *football_player*) or single word types that are not present in the BERT MLM vocabulary. The BERT MLM vocabulary covers about 92% of the labels in the human annotated dataset constructed by Choi et al. (2018). Type coverage is a known issue with weak supervision, and is tolerable if the generated labels can be used to achieve our final goal: improving the performance of the ultra-fine entity typing model.

3.2 Training Data

Our approach generates type labels for all three types of mentions: named entity mentions, pronoun mentions, and nominal mentions. For named entity mentions and nominal mentions, existing automatic annotation approaches can already provide some labels for them by using the entity types in knowledge bases or using the head words as types (Ling and Weld, 2012; Choi et al., 2018). Thus, we combine these labels with the labels generated by us. For pronoun mentions, no other labels are used.

Besides the automatically annotated samples, we can also use a small amount of human annotated samples provided by the dataset for model training.

3.3 Model Training

Our ultra-fine entity typing model follows the BERT-based model in (Onoe and Durrett, 2019). Given a sentence that contains an entity mention, we form the sequence "[CLS] sentence [SEP] mention string [SEP]" as the input to BERT. Then, denoting the final hidden vector of the "[CLS]" token as $\boldsymbol{u}$, we add a linear classification layer on top of $\boldsymbol{u}$ to model the probability of each type:

$$\boldsymbol{p} = \sigma(\boldsymbol{W}\boldsymbol{u}), \tag{1}$$

where σ is the sigmoid function, $\boldsymbol{W}$ is a trainable weight matrix. $\boldsymbol{p} \in \mathbb{R}^d$, where d is the number of types used by the dataset. We assign a type t to the mention if p_t, its corresponding element in $\boldsymbol{p}$, is larger than 0.5. If no such types are found, we assign the one with the largest predicted probability to the mention.

To make use of the automatically labeled samples, some existing approaches mix them with high quality human annotated samples while training models (Choi et al., 2018; Onoe and Durrett, 2019). However, we find that better performance can be obtained by pretraining the model on automatically labeled samples, then fine-tuning it on human annotated samples.

Following (Choi et al., 2018), we partition the whole type vocabulary used by the dataset into three non-overlapping sets: general, fine, and ultra-fine types, denoted with T_g, T_f and T_u, respectively. Then, we use the following objective for training:

$$\mathcal{J}(x) = \mathcal{L}(x, T_g)\mathbb{1}(\mathfrak{L}, T_g) + \mathcal{L}(x, T_f)\mathbb{1}(\mathfrak{L}, T_f) \\ + \mathcal{L}(x, T_u)\mathbb{1}(\mathfrak{L}, T_u), \tag{2}$$

where x is a training sample; $\mathfrak{L}$ denotes the set of type labels assigned to x through either human or automatic annotation. The function $\mathbb{1}(\mathfrak{L}, T)$ equals 1 when a type in $\mathfrak{L}$ is in set T and 0 otherwise. This loss can avoid penalizing some false negative labels.

Unlike existing studies, we define the function L differently for human annotated samples and automatically labeled samples. While pretraining with automatically labeled samples, the labels obtained through entity linking and head word supervision are usually of higher precision than those obtained through BERT MLM. Thus, we propose to assign different weights in the training objective to the labels generated with different methods:

$$\mathcal{L}(x, T) = -\sum_{t \in T} \alpha(t)[y_t \cdot \log(p_t) \\ + (1 - y_t) \cdot \log(1 - p_t)], \tag{3}$$

where y_t equals to 1 if t is annotated as a type for x and 0 otherwise; p_t is the probability of whether t should be assigned to x predicted by the model. The value of $\alpha(t)$ indicates how confident we are about the label t for x. Specifically, it equals to a predefined constant value larger than 1 when t is a positive type for x obtained through entity linking or head word supervision, otherwise, it equals to 1.

While fine-tuning with human annotated samples, we directly use the binary cross entropy loss:

$$\mathcal{L}(x, T) = -\sum_{t \in T} [y_t \cdot \log(p_t) + (1 - y_t) \cdot \log(1 - p_t)]. \tag{4}$$

3.4 Self-Training

Denote the ultra-fine entity typing model obtained after pretraining on the automatically labeled data as h, and the model obtained after fine-tuning h with human annotated data as m. A weakness of m is that at the fine-tuning stage, it is trained with only a small number of samples. Thus, we employ self-training to remedy this problem.

By using m as a teacher model, our self-training step fine-tunes the model h again with a mixture of the samples from the automatically labeled data and the human annotated data. This time, for the automatically annotated samples, we use pseudo labels generated based on the predictions of m instead of their original weak labels. The newly fine-tuned model should perform better than m, and is used for evaluation.

Denote the set of human annotated samples as H, the set of automatically labeled samples as A. The training objective at this step is

$$\mathcal{J}_{ST} = \frac{1}{|H|}\sum_{x \in H} \mathcal{J}(x) + \lambda \frac{1}{|A|}\sum_{x \in A} \mathcal{L}_{ST}(x), \tag{5}$$

where λ is a hyperparameter that controls the strength of the supervision from the automatically labeled data.

While computing loss for the samples in A, we only use the types that are very likely to be positive or negative. For a sample x, let p_t be the probability of it belonging to type t predicted by the model m. We consider a type t very likely to be positive if p_t is larger than a threshold P, or if t is a weak label of x and p_t is larger than a smaller threshold P_w. Denote the set of such types as $\hat{Y}^+(x)$. We consider a type t very likely to be negative if p_t is smaller than $1 - P$. Denote the set of such types as $\hat{Y}^-(x)$. Then we have:

$$\mathcal{L}_{ST}(x) = -\sum_{t \in \hat{Y}^+(x)} \log(p_t) \\ - \sum_{t \in \hat{Y}^-(x)} \log(1 - p_t). \tag{6}$$

Thus, we compute the binary cross entropy loss with only the types in $\hat{Y}^+(x)$ and $\hat{Y}^-(x)$.

4 Application to Traditional Fine-grained Entity Typing

Our approach to generating weak entity type labels with BERT MLM can also be applied to the

1794

traditional fine-grained entity typing task. Different from ultra-fine entity typing, traditional fine-grained entity typing uses a manually designed entity type ontology to annotate mentions. The types in the ontology are organized in an hierarchical structure. For example, the ontology used by the Ontonotes dataset contains 89 types including /organization, /organization/company, /person, /person/politician, etc. On this dataset, our automatic annotation approach can mainly be helpful to generate better labels for nominal mentions.

We still use the same method described in Section 3.1 to create input for BERT MLM based on the given mention. But with traditional fine-grained entity typing, most mentions are assigned only one type path (e.g., a company mention will only be assigned labels {/organization, /organization/company}, which includes all the types along the path of /organization/company). Thus, while generating labels, we only use the most probable word predicted by the BERT MLM, which is mapped to the types used by the dataset if possible. For example, the word "company" and its plural form are both mapped to /organization/company. Such a mapping from free-form entity type words to the types used by the dataset can be created manually, which does not require much effort. We mainly construct the mapping with two ways: 1) Check each type used by the dataset, and think of a few words that should belong to it, if possible. For example, for the type /person/artist/author, corresponding words can be "author," "writer," etc. 2) Run the BERT MLM on a large number of inputs constructed with unannotated mentions, then try to map the words that are most frequently predicted as the most probable word to the entity type ontology.

Since only the most probable word predicted by the BERT MLM is used to produce labels, we also only use one hypernym relation pattern: "M and any other H."

For traditional fine-grained entity typing, we use our approach to generate labels for mentions that are not previously annotated with other automatic annotation approaches. While training, all the automatically labeled mentions are used together. The typing model is the same as the model described in 3.3. The binary cross entropy loss is directly employed as the training objective.

5   Experiments

We conduct experiments on our primary task: ultra-fine entity typing. In addition, we evaluate the performance of our approach when applied to traditional fine-grained entity typing.

5.1   Evaluation on Ultrafine

For ultra-fine entity typing, we use the dataset created by Choi et al. (2018). It uses a type set that contains 10,331 types. These types are partitioned into three categories: 9 general types, 121 fine-grained types, and 10,201 ultra-fine types. There are 5,994 human annotated samples. They are split into train/dev/test with ratio 1:1:1. It also provides 5.2M samples weakly labeled through entity linking and 20M samples weakly labeled through head word supervision.

We compare with the following approaches:

- **UFET** (Choi et al., 2018). This approach obtains the feature vector for classification by using a bi-LSTM, a character level CNN, and pretrained word embeddings.

- **LabelGCN** (Xiong et al., 2019). LabelGCN uses a graph propagation layer to capture label correlations.

- **LDET** (Onoe and Durrett, 2019). LDET learns a model that performs relabeling and sample filtering to the automatically labeled samples. Their typing model, which employs ELMo embeddings and a bi-LSTM, is train with the denoised labels.

- **Box** (Onoe et al., 2021). *Box* represents entity types with box embeddings to capture latent type hierarchies. Their model is BERT-based.

We use the BERT-Base-Cased version of BERT for both weak label generation and the typing model in Section 3.3. The hyperparameters are tuned through grid search using F1 on the dev set as criterion. The value of $\alpha(t)$ in Equation (3) is set to 5.0 for positive types obtained through entity linking or head word supervision. λ in Equation (5) is set to 0.01. P and P_w in Section 3.4 are set to 0.9 and 0.7, respectively. Our approach to generate labels through BERT MLM is applied to each weak sample provided in the original dataset. In addition, we also use our approach to annotate about 3.7M pronoun mentions, which are extracted through string matching from the English Gigaword corpus

Method	P	R	F1
UFET	47.1	24.2	32.0
LabelGCN	50.3	29.2	36.9
LDET	51.5	33.0	40.2
Box	52.8	38.8	44.8
Ours	**53.6**	**45.3**	**49.1**

Table 3: Macro-averaged Precision, Recall, and F1 of different approaches on the *test set*.

Method	P	R	F1
BERT-Ultra-Direct	51.0	33.8	40.7
BERT-Ultra-Pre	50.8	39.7	44.6
Ours (Single Pattern)	52.4	44.9	48.3
Ours (Unweighted Loss)	51.5	45.8	48.5
Ours (No Self-train)	53.5	42.8	47.5
Ours	**53.6**	**45.3**	**49.1**

Table 4: Performance of different variants of our approach on the *test set*. BERT-Ultra-Direct and BERT-Ultra-Pre are two baseline approaches that do not use labels generated with our BERT MLM based method in training.

(Parker et al., 2011). We generate 10 types for each sample[2]. With the procedure described in Sectiton 3.1, three hypernym extraction patterns are used while generating labels with BERT MLM: "M and any other H," "H such as M," "M and some other H." Specifically, adding "H such as M" and "M and some other H" improves the F1 score from 0.253 to 0.274, and from 0.274 to 0.279, respectively. Adding any more patterns cannot improve the F1 score for more than 0.007.

Following existing work (Onoe et al., 2021; Onoe and Durrett, 2019), we evaluate the macro-averaged precision, recall, and F1 of different approaches on the manually annotated test set. The results are in Table 3. Our approach achieves the best F1 score. It obtains more than 4% F1 score improvement over the existing best reported performance by Box in (Onoe et al., 2021). This demonstrates the effectiveness of our approach.

5.2 Ablation Study

For ablation study, we verify the effectiveness of the different techniques used in our full entity typing approach by evaluating the performance of the following variants: **Ours (Single Pattern)** only

uses one pattern: M and any other H; **Ours (Unweighted Loss)** removes the $\alpha(t)$ term in Equation (3); **Ours (No Self-train)** does not perform the self-training step. We also evaluate two baseline approaches: **BERT-Ultra-Direct** uses the same BERT based model described in Section 3.3, but is trained with only the human annotated training samples; **BERT-Ultra-Pre** also uses the same BERT based model, but is first pretrained with the existing automatically generated training samples in the dataset provided by Choi et al. (2018), then fine-tuned on the human annotated training data.

First, the benefit of using the labels generated through BERT MLM can be verified by comparing *Ours (No Self-train)* and BERT-Ultra-Pre. Because the techniques employed in *Ours (No Self-train)*, including the use of multiple hypernym extraction patterns and the weighted loss, are both for better utilization of our automatic entity type label generation method.

The effectiveness of the use of multiple hypernym extraction patterns, the weighted loss, and the self-training step can be verified by comparing *Ours* with *Ours (Single Pattern)*, *Ours (Unweighted Loss)* and *Ours (No Self-train)*, respectively. Among them, self-training is most beneficial.

5.3 Evaluation on Different Kinds of Mentions

It is also interesting to see how our approach performs on different kinds of mentions. Table 5 lists the performance of our full approach and two baseline systems on the three kinds of mentions in the dataset: named entity mention, pronoun mentions, and nominal mentions.

Our approach performs much better than BERT-Ultra-Pre on all three kinds of mentions. The improvements in F1 on pronoun and nominal mentions are relatively more substantial.

5.4 Case Study

Table 6 presents several ultra-fine entity typing examples, along with the human annotated labels, and the labels predicted by BERT-Ultra-Pre, BERT MLM, and our full approach.

In the first example, the label *prisoner* is a type that depends on the context, and is usually not assigned to humans in knowledge bases. We think that since we can assign such labels to the training samples with our BERT MLM based approach, our

---

[2]The performance of the trained model is relatively insensitive with respect to the number of labels generated with MLM. The difference between the F1 scores of the models trained using 10 and 15 generated types is less than 0.005.

	Named Entity			Pronoun			Nominal		
Method	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
BERT-Ultra	58.1	45.1	50.8	52.9	42.9	47.4	47.4	26.9	34.3
BERT-Ultra-Pre	54.7	50.5	52.5	51.3	46.1	48.6	45.2	33.7	38.6
Ours	58.3	54.4	56.3	57.2	50.0	53.4	49.5	38.9	43.5

Table 5: Performance on named entity mentions, pronoun mentions, and nominal mentions, respectively.

Sentence	Captured in 1795, **he** was confined at Dunkirk, escaped, set sail for India, was wrecked on the French coast, and condemned to death by the decree of the French Directory.
Human	prisoner, person
BERT-Ultra-Pre	person, soldier, man, criminal
BERT MLM	man, prisoner, person, soldier, officer
Ours	person, soldier, man, prisoner
Sentence	Also in the morning, **a roadside bomb** struck a police patrol on a main road in Baghdad's northern neighborhood of Waziriya, damaging a police vehicle ...
Human	bomb, weapon, object, explosive
BERT-Ultra-Pre	object, event, attack, bomb
BERT MLM	weapon, threat, evidence, device, debris
Ours	object, weapon, bomb
Sentence	In October 1917, **Sutton** was promoted (temporarily) to the rank of major and appointed Officer Commanding No.7 Squadron, a position he held for the remained of the War.
Human	soldier, officer, male, person
BERT-Ultra-Pre	person, politician, male
BERT MLM	officer, pilot, man, unit, aircraft
Ours	person, soldier, male, officer

Table 6: Ultra-fine entity typing examples with the corresponding human annotated labels and predictions of three different systems. Entity mentions are in bold and underlined. For BERT MLM, we list the top five labels.

model is better at predicting them than the baseline model.

The second and third examples demonstrate that our model may not only improve the recall by predicting more correct types, but also reduce incorrect predictions that do not fit the mention or the context well.

5.5 Evaluation on Ontonotes

The Ontonotes dataset uses an ontology that contains 89 types to label entity mentions. We use the version provided by Choi et al. (2018). It includes

11,165 manually annotated mentions, which are split into a test set that contains 8,963 mentions, and a dev set that contain 2,202 mentions. It also provides about 3.4M automatically labeled mentions.

Since existing annotations for named entity mentions may be more accurate than the annotations obtained through our approach, we only apply our method to label nominal mentions. Applying the approach in Section 4, we create 1M new automatically labeled mentions with the head word supervision samples (such samples contain mostly nominal mentions) in the ultra-fine dataset. They are used together with the originally provided 3.4M mentions to train the typing model.

On this dataset, we compare with the following approaches: **UFET** (Choi et al., 2018), **LDET** (Onoe and Durrett, 2019), **DSAM** (Hu et al., 2020), **LTRFET** (Lin and Ji, 2019), **BERT-Direct**. Where BERT-Direct uses the same BERT based model as our approach, but trains with only the weak samples provided in the dataset. LTRFET adopts a hybrid classification method to exploit type inter-dependency. DSAM is a diversified semantic attention model with both mention-level attention and context-level attention.

For our approach and BERT-Direct, we still use the pretrained BERT-Base-Cased model for initialization. Although a very large number of weakly labeled mentions are provided, not all of them are needed for training the models. In our experiments, for both our approach and BERT-Direct, the performance does not increase after training on about 0.3M mentions.

We report strict accuracy, macro-averaged F1, and micro-averaged F1 (Ling and Weld, 2012). The results are in Table 7. As we can see, our approach also achieves the best performance on this dataset. Comparing it with BERT-Direct demonstrates the benefit of the samples automatically labeled with BERT MLM.

However, less improvement is achieved on OntoNotes than on the ultra-fine entity typing

Method	Acc	Macro F1	Micro F1
UFET	59.5	76.8	71.8
LTRFET	63.8	82.9	77.3
LDET	64.9	84.5	79.2
DSAM	66.06	83.07	78.19
BERT-Direct	63.25	80.84	75.90
Ours	**67.44**	**85.44**	**80.35**

Table 7: Performance of different approaches on Ontonotes. We report strict accuracy, macro-averaged F1, and micro-averaged F1.

dataset. We think there are two main reasons. First, OntoNotes uses a much smaller entity type set (89 types) than the ultra-fine entity typing dataset (10,331 types). As a result, some finer grained types that can be produced by our approach become less beneficial. Second, generating type labels that are highly dependent on the context (e.g., types like *criminal*, *speaker*) is an advantage of our approach, and the ultra-fine entity typing dataset contains more such type labels.

6 Conclusion

In this work, we propose a new approach to automatically generate ultra-fine entity typing labels. Given a sentence that contains a mention, we insert a hypernym extraction pattern with a "[MASK]" token in it, so that a pretrained BERT MLM may predict hypernyms of the mention for "[MASK]." Multiple patterns are used to produce better labels for each mention. We also propose to use a weighted loss and perform a self-training step to learn better entity typing models. Experimental results show that our approach greatly outperforms state-of-the-art systems. Additionally, we also apply our approach to traditional fine-grained entity typing, and verify its effectiveness with experiments.

References

Tongfei Chen, Yunmo Chen, and Benjamin Van Durme. 2020. Hierarchical entity typing via multi-level learning to rank. In *Proceedings of ACL*, pages 8465–8475.

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

Hongliang Dai, Donghong Du, Xin Li, and Yangqiu Song. 2019. Improving fine-grained entity typing with entity linking. In *Proceedings of EMNLP-IJCNLP*, pages 6211–6216.

Hongliang Dai, Yangqiu Song, and Xin Li. 2020. Exploiting semantic relations for fine-grained entity typing. In *Automated Knowledge Base Construction*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

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

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING*.

Yanfeng Hu, Xue Qiao, Xing Luo, and Chen Peng. 2020. Diversified semantic attention model for fine-grained entity typing. *IEEE Access*.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Hailong Jin, Lei Hou, Juanzi Li, and Tiansi Dong. 2019. Fine-grained entity typing via hierarchical multi graph convolutional networks. In *Proceedings of EMNLP-IJCNLP*, pages 4970–4979.

Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S. Weld. 2014. Type-aware distantly supervised relation extraction with linked arguments. In *Proceedings of EMNLP*, pages 1891–1901.

Chin Lee, Hongliang Dai, Yangqiu Song, and Xin Li. 2020. A chinese corpus for fine-grained entity typing. In *Proceedings of LREC*, pages 4451–4457.

Ying Lin and Heng Ji. 2019. An attentive fine-grained entity typing model with latent type representation. In *Proceedings of EMNLP-IJCNLP*, pages 6198–6203.

Xiao Ling, Sameer Singh, and Daniel S Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328.

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

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*, pages 97–109.

Yasumasa Onoe, Michael Boratko, and Greg Durrett. 2021. Modeling fine-grained entity types with box embeddings. *arXiv preprint arXiv:2101.00345*.

Yasumasa Onoe and Greg Durrett. 2019. Learning to denoise distantly-labeled data for entity typing. In *Proceedings of NAACL-HLT*, pages 2407–2417.

Yasumasa Onoe and Greg Durrett. 2020. Interpretable entity representations through large-scale typing. In *Proceedings of EMNLP*, pages 612–624.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition. *Linguistic Data Consortium*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of EMNLP-IJCNLP*, pages 2463–2473.

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

Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. 2016. A large database of hypernymy relations extracted from the web. In *Proceedings of LREC*, pages 360–367.

Ji Xin, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2018. Improving neural fine-grained entity typing with knowledge attention. In *Proceedings of AAAI*, volume 32.

Wenhan Xiong, Jiawei Wu, Deren Lei, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Imposing label-relational inductive bias for extremely fine-grained entity typing. In *Proceedings of NAACL-HLT*, pages 773–784.

Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012. Hyena: Hierarchical type classification for entity names. In *Proceedings of COLING*, pages 1361–1370.

Yunyi Zhang, Jiaming Shen, Jingbo Shang, and Jiawei Han. 2020. Empower entity set expansion via language model probing. In *Proceedings of ACL*, pages 8151–8160.