Ultra-fine Entity Typing with Indirect Supervision from Natural Language Inference

Bangzheng Li[◇†*], Wenpeng Yin[‡], and Muhao Chen[◇]

[◇]University of Southern California, USA
[†]University of Illinois at Urbana-Champaign, USA
[‡]Temple University, USA
`vincentleebang@gmail.com; wenpeng.yin@temple.edu;`
`muhaoche@usc.edu`

Abstract

The task of ultra-fine entity typing (UFET) seeks to predict diverse and free-form words or phrases that describe the appropriate types of entities mentioned in sentences. A key challenge for this task lies in the large number of types and the scarcity of annotated data per type. Existing systems formulate the task as a multi-way classification problem and train directly or distantly supervised classifiers. This causes two issues: (i) the classifiers do not capture the type semantics because types are often converted into indices; (ii) systems developed in this way are limited to predicting within a pre-defined type set, and often fall short of generalizing to types that are rarely seen or unseen in training.

This work presents LITE 🍺, a new approach that formulates entity typing as a natural language inference (NLI) problem, making use of (i) the indirect supervision from NLI to infer type information meaningfully represented as textual hypotheses and alleviate the data scarcity issue, as well as (ii) a learning-to-rank objective to avoid the pre-defining of a type set. Experiments show that, with limited training data, LITE obtains state-of-the-art performance on the UFET task. In addition, LITE demonstrates its strong generalizability by not only yielding best results on other fine-grained entity typing benchmarks, more importantly, a pre-trained LITE system works well on new data containing unseen types.[1]

1 Introduction

Entity typing, inferring the semantic types of the entity mentions in text, is a fundamental and long-lasting research problem in natural language understanding, which aims at inferring the semantic types of the entities mentioned in text. The resulted type information can help with grounding human language components to real-world concepts (Chandu et al., 2021), and provide valuable prior knowledge for natural language understanding tasks such as entity linking (Ling et al., 2015; Onoe and Durrett, 2020), question answering (Yavuz et al., 2016), and information extraction (Koch et al., 2014). Prior studies have mainly formulated the task as a multi-way classification problems (Wang et al., 2021; Zhang et al., 2019; Chen et al., 2020a; Hu et al., 2020).

However, earlier efforts for entity typing are far from enough for representing real-world scenarios, where types of entities can be extremely diverse. Accordingly, the community has recently paid much attention to more fine-grained modeling of types for entities. One representative work is the Ultra-fine Entity Typing (UFET) benchmark created by Choi et al. (2018). The task seeks to search for the most appropriate types for an entity among over ten thousand free-form type candidates. The drastic increase of types forces us to question whether the multi-way classification framework is still suitable for UFET. In this context, two main issues are noticed from prior work. First, prior studies have not tried to understand the target types since most classification systems converted all types into indices. Without knowing the semantics of types, it is hard to match an entity mention to a correct type, especially when there is not sufficient annotated data for each type. Second, existing entity typing systems are far behind the desired capability in real-world applications in which any open-form types can appear. Specifically, those pre-trained multi-way

---

[1]Our models and implementation are available at `https://github.com/luka-group/lite`.

classifiers cannot recognize types that are unseen in training, especially when there is no reasonable mapping from existing types to unseen type labels, unless the classifiers are re-trained to include those new types.

To alleviate the aforementioned challenges, we propose a new learning framework that seeks to enhance ultra-fine entity typing with indirect supervision from natural language inference (NLI) (Dagan et al., 2006). Specifically, our method LITE 🍺, (Language Inference based Typing of Entities), treats each entity-mentioning sentence as a premise in NLI. Using simple, template-based generation techniques, a candidate type is transformed into a textual description and is treated as the hypothesis in NLI. Based on the premise sentence and a hypothesis description of a candidate type, the entailment score given by an NLI model is regarded as the confidence of the type. On top of the pre-trained NLI model, LITE conducts a learning-to-rank objective, which aims at scoring hypotheses of positive types higher than the hypotheses of sampled negative types. Finally, the label candidates whose hypotheses obtain scores above a threshold are given as predictions by the model.

Technically, LITE benefits ultra-fine entity typing from three perspectives. First, the inference ability of a pre-trained NLI model can provide effective indirect supervision to improve the prediction of type information. Second, the hypothesis, as a type description, also provides a semantically rich representation of the type, which further benefits few-shot learning with insufficient labeled data. Moreover, to handle the dependency of type labels in different granularities, we also utilize the inference ability of an NLI model to learn that the finer label hypothesis of an entity mention entails its general label hypothesis. Experimental results on the UFET benchmark (Choi et al., 2018) show that LITE drastically outperforms the recent state-of-the-art (SOTA) systems (Dai et al., 2021; Onoe et al., 2021; Liu et al., 2021) without any need of distantly supervised data as they do. In addition, our LITE also yields the best performance on traditional (less) fine-grained entity typing tasks.[2] What's more, because we adopt a learning-to-rank objective to

optimize the inference ability of LITE rather than classification on a specified label space, it is feasible to apply the trained model across different typing data sets. We therefore test its transferability by training on UFET and evaluate on traditional fine-grained benchmarks to get promising results. Moreover, we also examined the time efficiency of LITE, and discussed about the trade-off between training and inference costs in comparison with prior methods.

To summarize, the contributions of our work are three fold. First, to our knowledge, this is the first work that uses NLI formulation and NLI supervision to handle entity typing. As a result, our system is able to retain the labels' semantics and encode the label dependency effectively. Second, our system offers SOTA performance on both ultra-fine entity typing and regular fine-grained typing tasks, being particularly strong at predicting zero-shot and few-shot cases. Finally, we show that our system, once trained, can also work on different test sets that are free to have unseen types.

2   Related Work

Entity Typing.   Traditional entity typing was introduced and thoroughly studied by Ling and Weld (2012). One main challenge that earlier efforts have focused on was to obtain sufficient training data to develop the typing model. To do so, automatic annotation has been commonly used in the a series of studies (Gillick et al., 2014; Ling and Weld, 2012; Yogatama et al., 2015). Later research was developed for further improvement by modeling the label dependency with a hierarchy-aware loss (Ren et al., 2016; Xu and Barbosa, 2018). External knowledge from knowledge bases has also been introduced to capture the semantic relations or relatedness of type information (Jin et al., 2019; Dai et al., 2019; Obeidat et al., 2019). Ding et al. (2021) adopt prompts to model the relationship between entities and type labels, which is similar to our template-based type description generation. However, their prompts are intended for label generation from masked language models whereas our templates realize the supervision from NLI.

More recently, Choi et al. (2018) proposed the ultra-fine entity typing (UFET) task, which involved free-form type labeling to realize the

---

[2]Note that although these more traditional entity typing tasks are termed ''fine-grained entity typing'', their typing systems are much less fine-grained than that of UFET.

open-domain label space with much more comprehensive coverage of types. As the UFET tasks non-trivial learning and inference problems, several methods have been explored by more effectively modeling the structure of the label space. Xiong et al. (2019) utilized a graph propagation layer to impose label-relation bias in order to capture type dependencies implicitly. Onoe and Durrett (2019) trained a filtering and relabeling model with the human annotated data to denoise the automatically generated data for training. Onoe et al. (2021) introduced box embeddings (Vilnis et al., 2018) to represent the dependency among multiple levels of type labels as topology of axis-aligned hyper-rectangles (*boxes*). To further cope with insufficient training data, Dai et al. (2021) used pre-trained language model for augmenting (noisy) training data with masked entity generation. Different from their strategy of augmenting training data, our approach generates type descriptions to leverage indirect supervision from NLI which requires no more data samples.

Natural Language Inference and Its Applications. Early approaches towards NLI problems were based on studying lexical semantics and syntactic relations (Dagan et al., 2006). Subsequent research then introduced deep-learning methods to this task to capture contextual semantics. Parikh et al. (2016) utilized Bi-LSTM (Hochreiter and Schmidhuber, 1997) to encode the input tokens and use attention mechanism to capture substructures of input sentences. Most recent work develops end-to-end trained NLI models that leverage pre-trained language models (Devlin et al., 2019; Liu et al., 2019) for sentence pair representation and large learning resources (Bowman et al., 2015; Williams et al., 2018) for training.

Specifically, because pre-trained NLI models benefit with generalizable logical inference, current literature has also proposed to leverage NLI models to improve prediction tasks with insufficient training labels, including zero-shot and few-shot text classification (Yin et al., 2019). Shen et al. (2021) adopted *RoBERTa-large-MNLI* (Liu et al., 2019) to calculate the document similarity for document multi-class classification. Chen et al. (2021) proposed to verify the output of a QA system with NLI models by converting the question and answer into a hypothesis and extracting textual evidence from the reference document as the premise.

Recent work by Yin et al. (2020) and White et al. (2017) is particularly relevant to this topic, which utilizes NLI as a unified solver for several text classification tasks such as co-reference resolution and multiple choice QA in few-shot or fully-supervised manner. Yet our work handles a learning-to-rank objective for inference in a large candidate space, which not only enhances learning under a data-hungry condition, but is also free to be adapted to infer new labels that are unseen to training. Yin et al. (2020) also proposed an approach to transform co-reference resolution task into NLI manner and we modified this as one of our template generation methods, which is discussed in §3.2.

3 Method

In this section, we introduce the proposed method for (ultra-fine) entity typing with NLI. We start with problem definition and an overview of our NLI-based entity typing framework (§3.1), followed by technical details of type description generation (§3.2), label dependency modeling (§3.3), learning objective (§3.4), and inference (§3.5).

3.1 Preliminaries

Problem Definition. The input of an entity typing task is a sentence s and an entity mention of interest $e \in s$. This task aims at typing e with one or more type labels from the label space L. For instance, in ''*Jay is currently working on his Spring 09 collection, which is being sponsored by the YKK Group.*'', the entity ''*Jay*'' should be labeled as *person*, *designer*, or *creator* instead of *organization* or *location*.

The structure of the label space L can vary. For example, in some benchmarks like OntoNotes (Gillick et al., 2014), labels are provided in canonical form and strictly depend on their ancestor types. In this case, a type label *bridge* appears as */location/transit/bridge*. However, in benchmarks like FIGER (Ling and Weld, 2012), partial labels have a dependency with their ancestors while the others are free-form and uncategorized. For instance, the label *film* is given as */art/film* but *currency* appears as a single word. For our primary task, for ultra fine-grained entity typing, the UFET benchmark (Choi et al., 2018) provides no ontology of the labels and the label vocabulary consists of free-form words only. In this
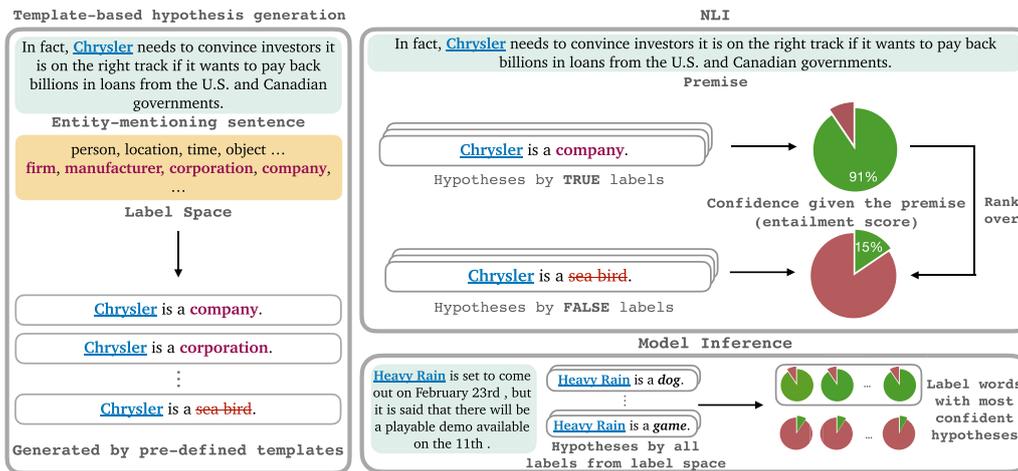
609

Figure 1: Entity typing by LITE with indirect supervision from NLI.

case, *film star* and *person* can appear independently in an annotation set with no dependency information provided.

Overview of LITE. Given a sentence with at least an entity mention, LITE treats the sentence as the premise in NLI, and then learns to type the entity in three consecutive steps (Figure 1). First, LITE employs a simple, low-cost template-based technique to generate a natural language description for a type candidate. This type description is treated as the hypothesis in NLI. For this step, we explore three different description generation templates (§3.2). Second, to capture label dependency, whether or not the type ontology is provided, LITE consistently generates type descriptions for any ancestors of the original type label on the previous sentence and learns their logical dependencies (§3.3). These two steps create positive cases of type descriptions for the entity mention in the previous sentence. Last, LITE fine-tunes a pre-trained NLI model with a learning-to-rank objective that ranks the positive case(s) over negative-sampled type descriptions according to the entailment score (§3.5). During the inference phase, given another sentence that mentions an entity to be typed, our model predicts type that leads to the hypothetical type description with the highest entailment score. In this way, LITE can effectively leverage indirect supervision signals of a (pre-trained) NLI model to infer the type information of a mentioned entity.

We describe the technical details of training and inference steps of LITE in the rest of the section.

3.2 Type Description Generation

Given each sentence s with an annotated entity mention e, LITE first generates a natural language type description $T(a)$ for the type label annotation a. The description will later act as a hypothesis in NLI. Specifically, we consider several generation technique to obtain such type descriptions, for which the details are described as follows.

- *Taxonomic statement.* The first template directly connects the entity mention and the type label with an ''is-a'' statement, namely, ''[ENTITY] is a [LABEL]''.

- *Contextual explanation.* The second template generates a declarative sentence that adds a context-related connective. The generated type description is in the form of ''In this context, [ENTITY] is referring to [LABEL]''.

- *Label substitution.* Yin et al. (2020) proposed to transform co-reference resolution problem into NLI manner by replacing the pronoun mentions with candidate entities. Inspired by their transformation, this technique directly replaces the [ENTITY] in the original sentence with [LABEL]. Therefore, the NLI model will treat the modified sentence with a ''type mention'' as the hypothesis of the original sentence with the entity mention.

As shown in Table 1, each template provides a semantically meaningful way to connect the entity

Templates	Type Descriptions	Premise-Hypothesis Pairs for NLI
Taxonomic Statement	**Jay** is a **producer**.	Premise: "**Jay** is currently working on his Spring 09 collection, . . . " Hypothesis: "**Jay** is a **producer**."
Contextual Explanation	In this context, **career at a company** is referring to **duration**.	Premise: "No one expects a **career at a company** any more, . . . " Hypothesis: "In this context, **career at a company** is referring to **duration**."
Label Substitution	**Musician** knows how to make a hip-hop record sound good.	Premise: "**He** knows how to make a hip-hop record sound good." Hypothesis: "**Musician** knows how to make a hip-hop record sound good."

Table 1: Type description instances of three templates. **Entity mentions** are boldfaced and underlined whereas **label words** are only boldfaced.

with a label. In this way, the inference ability of an NLI model can be leveraged to capture the relationship of entity and label, given the original entity-mentioning sentence as the premise.

In particular, we have also tried the automatic template generation method proposed by Gao et al. (2021), which has led to the adoption of the contextual explanation template. Such a template technique adopts the pre-trained text-to-text Transformer T5 (Raffel et al., 2020) to generate prompt sentences for fine-tuning language models. In our case, T5 mask tokens are added between the sentence, the entity, and the label. Since T5 is trained to fill in the blanks within its input, the output tokens can be used as the template for our type description. For example, given the sentence *"Anyway, Nell is their new singer, and I would never interrupt her show."*, the entity *Nell* and the annotations *(singer, musician, person)*, we can formulate the input to T5 as *"Anyway, Nell is their new singer, and I would never interrupt her show. <X> Nell <Y> singer <Z>"*. T5 will then fill in the placeholders *<X>, <Y>, <Z>* and output ". . . I would never interrupt her show. In fact, Nell is a singer." We observe that most of the generated templates given by T5 have appeared as the format where a prepositional phrase (*in fact*, *in this context*, *in addition*, etc.) followed by a statement such as *"[ENTITY] is a [LABEL]"* or *"[ENTITY] became [LABEL]"*. Accordingly, we select the above contextual explanation template, which is the most representative pattern observed in the generations.

In the training process, we use one of the three templates to generate the hypotheses, for which the same template will also be used to obtain the candidate hypotheses in inference. According to our preliminary results on the dev set, the taxonomic statement generation generally gives better performance than the others under most settings, for which the analysis is presented in

§4.3. Thus, the main experimentation is reported as the configuration where LITE uses the type descriptions based on taxonomic statement.

3.3 Modeling Label Dependency

The rich entity type vocabulary may form hierarchies that enforce logical dependency among labels of different specificity. Hence, we extend the generation process of type description to better capture such a label dependency. In detail, for a specific type label for which LITE has generated a type description, if there are ancestor types, we not only generate descriptions for each of the ancestor types, but also conduct learning among these type descriptions. The descendant type description would act as the premise and the ancestor type description would act as the hypothesis. For instance, in OntoNotes (Gillick et al., 2014) or FIGER (Ling and Weld, 2012), suppose a sentence mentions the entity *London* and is labeled as */location/city*, if the taxonomic statement based description generation is used, LITE will yield descriptions for both levels of types, that is, "London is a city" and "London is a location". In such a case, the more fine-grained type description "London is a city" can act as the premise of the more coarse-grained description "London is a location", so as to help capture the dependency between two labels "city" and "location". Such paired type descriptions are added to training and will be captured by the dependency loss $\mathcal{L}_d$ as described in §3.4.

This technique to capture label dependency can be easily adapted to tasks where a type ontology is unavailable, but each instance is directly annotated with multiple type labels of different specificity. Particularly for the UFET task (Choi et al., 2018), while no ontology is provided for the label space, the task separates the type label vocabulary into different specificity, namely,

general, *fine*, and *ultra-fine* ones. Because its annotation to an entity from a sentence includes multiple labels of different specificity, we can still utilize the aforementioned dependency modeling method. For instance, an entity *Mike Tyson* may be simultaneously labeled as *person* (general), *sportsman* (fine), and *boxer* (ultra-fine). Similar to using an ontology, each pair of descendant and ancestor descriptions among the three generations ''Mike Tyson is a sportsman'', ''Mike Tyson is a person'', and ''Mike Tyson is a sportsman'' are also added to training.

3.4 Learning Objective

Let L be the type vocabularies, and the learning objective of LITE is to conduct learning-to-rank on top of the NLI model. Given a sentence s with mentioned entity e, we use P to denote all true type labels of e that may include the original label and any induced ancestor labels as described in §3.3. Then, for each label $p \in P$ whose type description is generated as $H(p)$ by one of the techniques in §3.2, the NLI model calculates the entailment score $\varepsilon(s, H(p)) \in [0, 1]$ for the premise s and hypothesis $H(p)$. Meanwhile, negative sampling randomly selects a false label $p' \in L \setminus P$. Following the same procedure as above, the entailment score $\varepsilon(s, H(p'))$ is obtained for the premise s and the negative-sample hypothesis $H(p')$. The margin ranking loss for an annotated training case is then defined as

$$\mathcal{L}_t = [\varepsilon(s, H(p')) - \varepsilon(s, H(p)) + \gamma]_+.$$

$[x]_+$ denotes the positive part of the input x (i.e., $max(x, 0)$) and γ is a non-negative constant.

We also similarly define a ranking loss to model the label dependency. Still given the above annotated sentence s and the set of all true type labels P, as described in §3.3, for any exiting pair of ancestor type p_{an} and descendant type p_{de} from P, the training phase also captures the entailment relation between their descriptions. This process regards $H(p_{de})$ as the premise and $H(p_{an})$ as the hypothesis, and the NLI model therefore yields an entailment score $\varepsilon(H(p_{de}), H(p_{an}))$. The label dependency loss is then defined as the following ranking loss:

$$\mathcal{L}_d = [\varepsilon(H(p_{de}), H(p'_{an})) \\ - \varepsilon(H(p_{de}), H(p_{an})) + \gamma]_+,$$

where p'_{an} is negative-sampled type label.

The eventual learning objective is to optimize the following joint loss:

$$\mathcal{L} = \frac{1}{|S|} \sum_{s \in S} \frac{1}{|P_s|} \sum_{p \in P_s} \mathcal{L}_t + \lambda \mathcal{L}_d$$

where S denotes the dataset containing sentences with typed entities, and P_s denotes the set of true labels on an entity of the sentence instance s. In this way, all annotations of each entity mention will be involved in training. λ here is a non-negative hyperparameter that controls the influence of dependency modeling.

3.5 Inference

The inference phase of LITE performs ranking on descriptions for all type labels from the vocabulary. For any given sentence s mentioning an entity e, LITE accordingly generates a type description for each candidate type label. Then, taking the sentence s as the premise, the fine-tuned NLI model ranks the hypothetical type descriptions according to their entailment scores. Finally, LITE selects the type label whose description receives the highest entailment score, or predicts with a threshold of entailment scores in cases where multi-label prediction is required.

4 Experiment

In this section, we present the experimental evaluation for LITE framework, based on both UFET (§4.1) and traditional (less) fine-grained entity typing tasks (§4.2). In addition, we also conduct comprehensive ablation studies to understand the effectiveness of the incorporated techniques (§4.3).

4.1 Ultra-Fine Entity Typing

We use the UFET benchmark created by Choi et al. (2018) for evaluation. The UFET dataset consists of two parts. (i) Human-labeled data (L): 5,994 instances split into train/dev/test by 1:1:1 (1,998 for each); (ii) Distant supervision data (D): including 5.2M instances that are automatically labeled by linking entity to KB, and 20M instances generated by headword extraction. We follow the original design of the benchmark to evaluate loose macro-averaged precision (P), recall (R), and F1.

Training Data. In our approach, the supervision can come from the **MNLI** data (NLI) (Williams et al., 2018), **d**istant supervision data (D), and the

human-labeled data (L). Therefore, we investigate the best combination of training data by exploring the following different training pipelines:

- LITE_{NLI}: Pre-train on MNLI[3], then predict directly, without any tuning on D or L;

- LITE_{L}: Only fine-tune on L;

- $\text{LITE}_{\text{NLI+L}}$: Pre-train on MNLI, then fine-tune on L;

- $\text{LITE}_{\text{D+L}}$: Pre-train on D, then fine-tune on L;

- $\text{LITE}_{\text{NLI+D+L}}$: First pre-train on MNLI, then on D, finally fine-tune on L.

Model Configurations. Our system is first initialized as RoBERTa-large (Liu et al., 2019) and AdamW (Loshchilov and Hutter, 2018) is used to optimize the model. The hyperparameters as well as the output threshold are tuned on the dev set: batch size 16, pre-training (D) learning rate 1e-6, fine-tuning (L) learning rate 5e-6, margin $\gamma = 0.1$ and $\lambda = 0.05$. The pre-training epochs are limited to 5, which is enough considering the large size of pre-training data. The fine-tuning epochs are limited to 2,000; models are evaluated every 30 epochs on dev and the best model is kept to conduct inference on test.

Baselines. We compare LITE with the following strong baselines. Except for LRN which is merely trained on the human annotated data, all the other baselines incorporate the distant supervision data as extra training resource.

- **UFET-biLSTM** (Choi et al., 2018) represents words using the GloVe embedding (Pennington et al., 2014) and captures semantic information of sentences, entities, as well as labels with a bi-LSTM and a character-level CNN. It also learns a type label embedding matrix to operate inner product with the context and mention representation for classification.

- **LabelGCN** (Xiong et al., 2019) improves UFET-biLSTM by stacking a GCN layer on the top to capture the latent label dependency.

- **LDET** (Onoe and Durrett, 2019) applies ELMo embeddings (Peters et al., 2018) for

word representation and adopts LSTM as its sentence and mention encoders. Similar to UFET-biLSTM, it learns a matrix to compute inner product with each input representation for classification. Additionally, LDET also trains a filter and relabeler to fix the label inconsistency in the distant supervision training data.

- **BOX4Types** (Onoe et al., 2021) introduces box embeddings to handle the type dependency problems. It uses BERT-large-uncased (Devlin et al., 2019) as the backbone and projects the hidden classification vector to a hyper-rectangular (box) space. Each type from the label space is also represented as a box and the classification is fulfilled by computing the intersection of the input text and type boxes.

- **LRN** (Liu et al., 2021) encodes the context and entity with BERT-base-uncased. Then two LSTM-based auto-regression networks capture the context-label relation and the label-label relation via attention mechanisms, respectively, in order to generate labels. They simultaneously construct bipartite graphs for sentence tokens, entities, and generated labels to perform relation reasoning and predict more labels.

- **MLMET** (Dai et al., 2021), the prior SOTA system, first generates additional distant supervision data by the BERT Masked Language Model, then stacks a linear layer on BERT to learn the classifier on the union label space.

Results. Table 2 compares LITE with baselines, in which LITE adopts the taxonomic statement template (i.e., ''*[ENTITY]* is a *[LABEL]*'').

Overall, $\text{LITE}_{\text{NLI+L}}$ demonstrates SOTA performance over other baselines, outperforming the prior top system MLMET (Dai et al., 2021) with 1.5% absolute improvement on F1. Recall that MLMET built a multi-way classifier on the its newly collected distant supervision data and the human-labeled data, our LITE optimizes a textual entailment scheme on the entailment data (i.e., MNLI) and the human-labeled entity typing data. This comparison verifies the effectiveness

---

[3]This is obtained from `huggingface.co/roberta -large-mnli`.

Model	P	R	F1
UFET-biLSTM (Choi et al., 2018)	48.1	23.2	31.3
LabelGCN (Xiong et al., 2019)	50.3	29.2	36.9
LDET (Onoe and Durrett, 2019)	51.5	33.0	40.1
Box4Types (Onoe et al., 2021)	52.8	38.8	44.8
LRN (Liu et al., 2021)	**54.5**	38.9	45.4
MLMET (Dai et al., 2021)	53.6	45.3	49.1
LITE NLI	1.5	7.1	2.5
L	48.7	45.8	47.2
D+L	27.5	**56.4**	37.0
NLI+D+L	45.4	49.9	47.4
NLI+L	52.4	48.9	**50.6**
–w/o label dependency	53.3	46.6	49.7

Table 2: Results on the ultra-fine entity typing task. $\texttt{LITE}$ series are equipped with the Taxonomic Statement template. ''w/o label dependency'' is applied to the ''NLI+L'' setting. The F1 result by $\texttt{LITE}_{\text{NLI+L}}$ is statistically significant (p-value < 0.01 in t-test) in comparison with the best baseline result by MLMET.

of using the entailment scheme and the indirect supervision from NLI.

The bottom block in Table 2 further explores the best combination of available training data. First, training on MNLI (i.e., $\texttt{LITE}_{\text{NLI}}$) alone does not provide promising results. This could be because the MNLI does not generalize well to this UFET task. $\texttt{LITE}_{\text{L}}$ removes the supervision from NLI as compared to $\texttt{LITE}_{\text{NLI+L}}$, causing a noticeable performance drop. In addition, the comparison between $\texttt{LITE}_{\text{NLI+L}}$ and $\texttt{LITE}_{\text{D+L}}$ illustrates that the MNLI data, as an out-of-domain resource, even provides more beneficial supervision than the distant annotations. To our knowledge, this is already the first work that shows rather than relying on gathering distant supervision data in the (entity-mentioning context, type) style, it is possible to find more effective supervision from other tasks (e.g., from entailment data) to boost the performance. However, when we incorporate the distant supervision data (D) into $\texttt{LITE}_{\text{NLI+L}}$, the new system $\texttt{LITE}_{\text{NLI+D+L}}$ performs worse. We present more detailed analyses in §4.3.

In addition, we also investigate the contribution of label dependency modeling by removing it from $\texttt{LITE}_{\text{NLI+L}}$. As results show in Table 2, incorporating label dependency helps improve the recall with a large margin (from 46.6 to 48.9) despite a minor drop for the precision, leading to notable overall improvement in F1.

4.2 Fine-grained Entity Typing

In addition to UFET, we are also interested in (i) the effectiveness of our $\texttt{LITE}$ to entity typing tasks with much fewer types, and (ii) seeing if our learned $\texttt{LITE}$ model from the ultra-fine task can be used for inference on other entity typing tasks, *which often have unseen types*, even without further tuning. To that end, we evaluate $\texttt{LITE}$ on OntoNotes (Gillick et al., 2014) and FIGER (Ling and Weld, 2012), two popular fine-grained entity typing benchmarks.

OntoNotes contains 3.4M automatically labeled entity mentions for training and 11k manually annotated instances that are split into 8k for dev set and 2k for test set. Its label space consists of 88 types and one more *other* type. In inference, $\texttt{LITE}$ outputs *other* if none of the 88 types is scored over the threshold described in §3.5. **FIGER** contains 2M data samples labeled with 113 types. The dev set and test set include 1,000 and 562 samples, respectively. Within its label space, 82 types have a dependency relation with their ancestor or descendant types while the other 30 types are uncategorized free-form words.

Results. Table 3 reports baseline results as well as results of two variants of $\texttt{LITE}$: One is pre-trained on UFET and directly transfers to predict on the two target benchmarks, the other conducts task-specific training on the target benchmark after pre-training on MNLI. The task-specific training variant outperforms respective prior SOTA on both benchmarks (OntoNotes: 86.4 vs. 85.4 in macro-F1, 80.9 vs. 80.4 in micro-F1; FIGER: 86.7 vs. 84.9 in macro-F1, 83.3 vs. 81.5 in micro-F1).

An interesting advantage of $\texttt{LITE}$ lies in its **transferability** across benchmarks. Table 3 demonstrates that our $\texttt{LITE}$ (pre-trained on UFET) offers competitive performance on both OntoNotes and FIGER even with only zero-shot transfer (it even exceeds the ''task-specific training'' version on OntoNotes).[4] Although there are disjoint type labels between these two datasets and UFET, there exist manually crafted mappings from UFET labels to them (e.g., ''musician'' to

---

[4] $\texttt{LITE}$ pre-trained on UFET performs worse on FIGER than $\texttt{LITE}$ with task-specific training. The main reason could be that a larger portion of FIGER test data comes with an entity of proper noun to be labeled with more compositional types, such as *government agency, athlete, sports facility*, which have appeared much less often on UFET.

614

Model	OntoNotes		FIGER	
	macro-F1	micro-F1	macro-F1	micro-F1
Hierarchy-Typing (Chen et al., 2020b)	73.0	68.1	83.0	79.8
Box4Types (Onoe and Durrett, 2020)	77.3	70.9	79.4	75.0
DSAM (Hu et al., 2020)	83.1	78.2	83.3	81.5
SEPREM (Xu et al., 2021)	–	–	86.1	82.1
MLMET (Dai et al., 2021)	85.4	80.4	–	–
LITE pre-trained on NLI+UFET	**86.6**	**81.4**	80.1	74.7
LITE NLI+task-specific training	86.4	80.9	**86.7**	**83.3**

Table 3: Results for fine-grained entity typing. All LITE model results are statistically significant (p-value < 0.05 in t-test) in comparison with the best baseline results by MLMET on OntoNotes and by SEPREM on FIGER.

Data Source	Sentence	Labels
Entity Linking	(a) From 1928-1929 , he enrolled in graduate coursework at Yale University in New Haven , **Connecticut**.	**location**, author, **province**, cemetery, person
	(b) Once Upon Andalasia is a video game based on the **film** of the same name.	**art**, **film**
Head Word	(c) You can also use them in casseroles and they can be grated and fried if you want to make **hash browns**.	brown
	(d) He has written **a number of short stories** in different fictional worlds, including Dragonlance, Forgotten Realms, Ravenloft and Thieves' World.	number
	(e) Despite obvious parallels and relationships , video art is not **film**.	**film**

Table 4: Examples of two sources of distant supervision data (one from entity linking, the other from head word extraction). In the right ''Labels'' column, **correct types** are boldfaced while incorrect ones are in gray.

''/person/artist/music''). In this way, traditional multi-way classifiers still work across the datasets after type mapping though we do not prefer human-involvement in real-world applications. To further test the transferability of LITE, a more challenging experimental setting for zero-shot type prediction is conducted and analyzed in §4.3.

4.3 Analysis

Through the following analyses, we try to answer following questions: (i) Why did the distant supervision data not help (as Table 2 indicates)? (ii) How effective is each type description template (Table 1)? (iii) With the NLI-style formulation and the indirect supervision, does LITE generalize better for zero-shot and few-shot prediction? Is trained LITE transferable to new benchmarks with unseen types? (iv) On which entity types

does our model perform better, and which ones remain challenging? (vi) How efficient is LITE?

Distant Supervision Data. As Table 2 indicates, adding distant supervision data in LITE$_{NLI+D+L}$ even leads to a drop of 3.2% absolute score in F1 from LITE$_{NLI+L}$. This should be due to the fact that the distant supervision data (D) are overall noisy (Onoe and Durrett, 2019). Table 4 lists some *frequent and typical* problems that exist in D based on entity linking and head-word extraction. In general, they will lead to two problems.

On the one hand, a large number of false positive types are introduced. Considering example (a) in Table 4, the state *Connecticut* is labeled as *author*, *cemetery*, and *person*. For example (c), *hash brown* is labeled as *brown*, turning the concept of food into color. Additionally, the head-word

Templates	LITE$_{NLI+L}$			LITE$_{NLI+D+L}$			LITE$_{D+L}$		
	P	R	F1	P	R	F1	P	R	F1
Taxonomic Statement	52.4	48.9	**50.6**	45.4	49.9	**47.4**	27.5	56.4	**37.0**
Contextual Explanation	50.8	49.2	50.2	45.3	48.5	46.8	26.9	55.4	36.2
Label Substitution	47.4	49.3	48.3	42.5	50.7	46.2	24.8	59.3	35.0

Table 5: Behavior of different type description templates under three training settings.

method is short in capturing the semantics. In example (d), *number* is falsely extracted as the type for *a number of short stories* because of the preposition ''of''.

On the other hand, such distant supervision may not comprehensively recall positive types. For instance, examples (b) and (e) are both about the entity ''film'' where the recalled types are correct. However, in the human annotated data, entity ''film'' may also be labeled as (''film'', ''art'', ''movie'', ''show'', ''entertainment'', ''creation''). In this situation, those missed positive types (i.e., ''movie'', ''show'', ''entertainment'', and ''creation'') will be selected by the negative sampling process of LITE and therefore negatively influence the performance. The comparison between LITE$_{NLI+L}$ and LITE$_{D+L}$ can further justify the superiority of the indirect supervision from NLI over that from the distant supervision data.

Type Description Templates. Table 5 reveals how template choices affect the typing performance. It is obvious that taxonomic statement outperforms the other two under all of the three training settings. The contextual explanation template yields close yet worse results, but the label substitution leads to more noticeable F1 drop. This may result from the absence of an entity mention in the hypothesis by label substitution. For instance, in ''*Soft eye shields are placed on the babies to protect their eyes.*'', LITE with label substitution generates related but incorrect type labels such as *treatment*, *attention*, or *tissue*.

Few- and Zero-shot Prediction. In §4.2, we discussed transferring LITE trained on UFET to other fine-grained entity typing benchmarks. Nevertheless, because UFET labels are still inclusive of them with mapping, we conducted a further experiment in which portions of UFET training labels are randomly filtered out so that 40% of the testing labels are unseen in training. We then investigated the LITE$_{NLI+L}$ performance on test
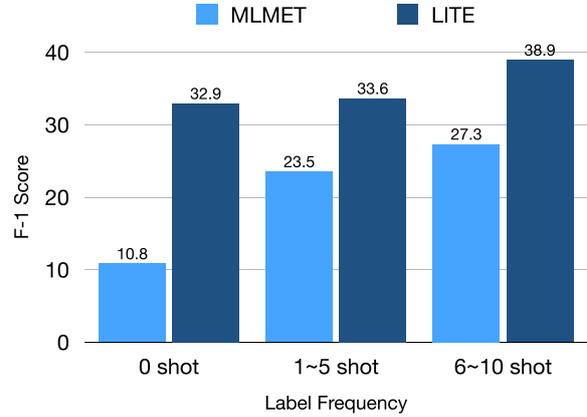

Figure 2: Performance comparison of our system LITE and the prior SOTA system, MLMET, on the filtered version of UFET for zero-shot and few-shot typing. The zero-shot labels correspond to the 40% test set type labels that are unseen in training. We also report the performance on other few-shot type labels.

types that have zero or a few labeled examples in the training set. Figure 2 shows the results of LITE$_{NLI+L}$ and the strongest baseline, MLMET. Note that while the held-out set of type labels is completely unseen to LITE, the full type vocabulary is provided for MLMET during its LM-based data augmentation process in this experiment.

As shown in the results, it is as expected that the performance on more frequent labels is better than on rare labels. LITE$_{NLI+L}$ outperforms MLMET on all the listed sets of zero- and few-shot labels; this reveals the strong low-shot prediction performance of our model. Particularly, on the extremely challenging zero-shot labels, LITE$_{NLI+L}$ drastically exceeds MLMET by 32.9% vs. 10.8% in F1. Hence, it is demonstrated that the NLI-based entity typing succeeds in more reliably representing and inferring rare and unseen entity types.

The main difference between the NLI framework and multi-way classifiers is that NLI makes use of the semantics of input text as well as the label text; conventional classifiers, however, only model the semantics of input text. Encoding the

616

	Input	True Labels	Prediction
LITE exceeds MLMET	(a) The University of California communications major gave **her** mother a fitting present, surprising herself by winning the 50-meter backstroke gold medal.	athlete, person, swimmer*, contestant, scholar, child	LITE: athlete, person, swimmer*, female, student, winner MLMET: athlete, person, child, adult, female, mother, woman
	(b) The apology is being viewed as a watershed in Australia , with major television networks airing **it** live and crowd gathering around huge screens in the city.	event, apology*, plea, regret	LITE: event, apology*, ceremony, happening, concept MLMET: event, message
	(c) A drawing table is also sometimes called a mechanical desk because, for several centuries, most **mechanical desks** were drawing tables.	object, desk, furniture*, board, desk, table	LITE: object, desk, furniture* MLMET: object, desk, computer
MLMET exceeds LITE	(d) He attended the University of Virginia, where he played **basketball and baseball**; his brother Bill also played baseball for the University.	basketball*, baseball, fun, action, activity, contact sport, game, sport, athletics, ball game, ball, event	LITE: activity, game, sport, event, ball game, ball, athletics MLMET: activity, game, sport, event, basketball*
	(e) The **manner in which it was confirmed** however smacked of an acrimonious end to the relationship between club and player with Chelsea.	manner*, way, concept, style, method	LITE: event MLMET: manner*, event

Table 6: Case study of labels on which LITE improves MLMET or MLMET outperforms LITE. Correct predictions are in blue and * indicates the representative label words for the discussed pattern.

semantics of on the label side is particularly beneficial when the type set is super large and many types lack training data. When some test labels are filtered out in the training process, LITE still performs well with its inference manner but classifiers (like MLMET) fail to recognize the semantics of unseen labels merely with their features. In this way, LITE maintains high performance when transferring across benchmarks with disjoint type vocabularies.

Case Study. We randomly sampled 100 labels on which LITE improves MLMET by at least 50% in F1 and here are the recognized typical patterns:

- **Contextual inference (28%)**: In case (a) of Table 6, considering the information ''winning the 50-meter backstroke gold medal'', LITE successfully types **her** with *swimmer* in addition to *athlete* that is given by MLMET.

- **Coreference (20%)**: In case (b), LITE correctly refers the pronoun entity **it** to ''apology'' but MLMET merely captures local information ''tv network airing'' to obtain the label words *event, message*.

- **Hypernym (19%)**: In case (c), even if there is no mention of *furniture* in the text, LITE gives a high confidence score to this type that is a hypernym of **mechanical desks**. Nevertheless, MLMET only obtains trivial answers such as *desk, object*.

On the other hand, we also sampled 100 labels on which MLMET performs better and it can be concluded that LITE falls short mainly in the following scenarios:

- **Multiple nominal words (30%)**: In sample (d) of Table 6, due to the ambiguous meaning of the type hypothesis ''**basketball and baseball** is a *basketball*'', LITE fails to predict the groundtruth label *basketball*.

- **Clause (28%)** Instance (e) illustrates a common situation when clauses are included in

617

	Named Entity			Pronoun			Nominal		
	P	R	F1	P	R	F1	P	R	F1
LITE$_{NLI+L}$	**58.6**	**55.5**	**57.0**	51.2	**57.5**	**54.2**	45.3	**47.1**	**46.2**
MLMET	58.3	54.4	56.3	**57.2**	50.0	53.4	**49.5**	38.9	43.5

Table 7: Performance comparison of LITE and prior SOTA, MLMET, on named entity, pronoun, and nominal entities, respectively.

the entity mention, where the effectiveness of type descriptions is harmed. The clausal information distracts LITE from focusing on the key part of the entity.

Prediction on Different Categories of Entity Mentions. We also investigated the prediction of LITE on three different categories of entity mentions from the UFET test data: named entities, pronouns, and nominals. For each category of mentions, we randomly sample 100 instances; the performance comparison against MLMET is reported in Table 7.

According to the results, LITE consistently outperforms MLMET on all three categories of entities and the improvement on nominal phrases (46.2% vs. 43.5% in F1) is most significant. This partly aligns with the capability of making inferences based on noun hypernyms, as discussed in the Case Study. Meanwhile, typing on nominals seeks to be more challenging than on the other two categories of entities, which, from our observation, is mainly due to two reasons. First, Nominal phrases with multiple words are more difficult to capture by the language model in general. Second, nominals are sometimes less concrete than pronouns and named entities, hence LITE also generates more abstract type labels. For example, LITE has labeled **the drink** in an instance as **substance**, which is too abstract and is not recognized by human annotators.

Time Efficiency. In general, LITE has much less training cost, of around 40 hours, than the previous strongest (data-augmentation-based) model MLMET, which requires over 180 hours, on the UFET task.[5] During the inference step, it takes about 35 seconds per new sentence for our model to do inference with a fixed type vocabulary of

over 10,000 different labels while a common multi-way classifier merely requires around 0.2 seconds. In fact, such a big difference in inference cost results from encoding longer texts and multiple encoding calculation time for the same text. It can be accelerated by modifying the encoding model structure which will be discussed in §5. However, LITE is much more efficient on dynamic type vocabulary. It requires almost no re-calculation when new, un-mappable labels are added to an existing type set but multi-way classifiers need re-training with an extended classifier every time (e.g., over 180 hours by the previous SOTA).

5 Conclusion and Future Work

We propose a new model, LITE, that leverages indirect supervision from NLI to type entities in texts. Through template-based type hypothesis generation, LITE formulates the entity typing task as a language inference task and meanwhile the semantically rich hypothesis remedies the data scarcity problem in the UFET benchmark.

Additionally, the learning-to-rank objective further helps LITE with generalized prediction across benchmarks with disjoint type sets. Our experimental results illustrate that LITE promisingly offers SOTA on UFET, OntoNotes, and FIGER, and yields strong performance on zero-shot and few-shot types. LITE pretrained on UFET also yields strong transferability by outperforming SOTA baselines when directly make predictions on OntoNotes and FIGER.

For future research, as mentioned in §4.3, we first plan to investigate ways to accelerate LITE by utilizing a late-binding cross-encoder (Pang et al., 2020) for linear-complexity NLI, and incorporating high-dimensional indexing techniques like ball trees in inference. To be specific, the premise and hypotheses can first be encoded respectively and the resulting representations can later be used to evaluate the confidence score of

---

[5]All time estimations are given by experiments performed on a commodity server with a TITAN RTX. Training and evaluation batch sizes are maximized to 16 or 128 for LITE and MLMET, respectively.

premise-hypothesis representation pairs through a trained network. With little expected loss in performance, `LITE` can still maintain its feature of strong transferability and zero-shot prediction.

In addition, we plan to extend NLI-based indirect supervision to information extraction tasks such as relation extraction and event extraction. Incorporating abstention-awareness (Dhamija et al., 2018) for handling unknown types is another meaningful direction. Additionally, Poliak et al. (2018) recast diverse types of reasoning datasets including NER, relation extraction, and sentiment analysis into the NLI structure, which we plan to incorporate as extra indirect supervision for `LITE` to further enhance the robustness of entity typing.

Acknowledgments

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Khyathi Raghavi Chandu, Yonatan Bisk, and Alan W. Black. 2021. Grounding 'grounding' in NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4283–4305, Online. Association for Computational Linguistics.

Jifan Chen, Eunsol Choi, and Greg Durrett. 2021. Can NLI models verify QA systems' predictions? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3841–3854, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tongfei Chen, Yunmo Chen, and Benjamin Van Durme. 2020a. Hierarchical entity typing via multi-level learning to rank. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8465–8475, Online. Association for Computational Linguistics.

Tongfei Chen, Yunmo Chen, and Benjamin Van Durme. 2020b. Hierarchical entity typing via multi-level learning to rank. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8465–8475, Online. Association for Computational Linguistics.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hongliang Dai, Donghong Du, Xin Li, and Yangqiu Song. 2019. Improving fine-grained entity typing with entity linking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6210–6215, Hong Kong, China. Association for Computational Linguistics.

Hongliang Dai, Yangqiu Song, and Haixun Wang. 2021. Ultra-fine entity typing with weak supervision from a masked language model. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long*

Papers), pages 1790–1799, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Akshay Raj Dhamija, Manuel Günther, and Terrance E. Boult. 2018. Reducing network agnostophobia. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 9175–9186.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021. Prompt-learning for fine-grained entity typing. *arXiv preprint arXiv:2108.10604*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

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*, 9(8):1735–1780.

Yanfeng Hu, Xue Qiao, Luo Xing, and Chen Peng. 2020. Diversified semantic attention model for fine-grained entity typing. *IEEE Access*, 9:2251–2265.

Hailong Jin, Lei Hou, Juanzi Li, and Tiansi Dong. 2019. Fine-grained entity typing via hierarchical multi graph convolutional networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*

(EMNLP-IJCNLP), pages 4969–4978, Hong Kong, China. Association for Computational Linguistics.

Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S. Weld. 2014. Type-aware distantly supervised relation extraction with linked arguments. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1891–1901, Doha, Qatar. Association for Computational Linguistics.

Xiao Ling, Sameer Singh, and Daniel S. Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328. https://doi.org/10.1162/tacl_a_00141

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Qing Liu, Hongyu Lin, Xinyan Xiao, Xianpei Han, Le Sun, and Hua Wu. 2021. Fine-grained entity typing via label reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4611–4622, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Rasha Obeidat, Xiaoli Fern, Hamed Shahbazi, and Prasad Tadepalli. 2019. Description-based zero-shot fine-grained entity typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 807–814, Minneapolis, Minnesota. Association for Computational Linguistics.

Yasumasa Onoe, Michael Boratko, Andrew McCallum, and Greg Durrett. 2021. Modeling fine-grained entity types with box embeddings. In *Proceedings of the 59th Annual Meeting of*

620

the *Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2051–2064, Online. Association for Computational Linguistics.

Yasumasa Onoe and Greg Durrett. 2019. Learning to denoise distantly-labeled data for entity typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2407–2417, Minneapolis, Minnesota. Association for Computational Linguistics.

Yasumasa Onoe and Greg Durrett. 2020. Interpretable entity representations through large-scale typing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 612–624, Online. Association for Computational Linguistics.

Shuai Pang, Jianqiang Ma, Zeyu Yan, Yang Zhang, and Jianping Shen. 2020. FAST-MATCH: Accelerating the inference of BERT-based text matching. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6459–6469, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 67–81, Brussels, Belgium. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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 the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1825–1834.

Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. 2021. TaxoClass: Hierarchical multi-label text classification using only class names. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4239–4249, Online. Association for Computational Linguistics.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic embedding of knowledge graphs with box lattice measures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 263–272, Melbourne, Australia. Association for Computational Linguistics.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-Adapter: Infusing knowledge into pre-trained models with adapters. In *Findings of the Association*

for Computational Linguistics: ACL-IJCNLP 2021, pages 1405–1418, Online. Association for Computational Linguistics.

Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. 2017. Inference is everything: Recasting semantic resources into a unified evaluation framework. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 996–1005, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 773–784, Minneapolis, Minnesota. Association for Computational Linguistics.

Peng Xu and Denilson Barbosa. 2018. Neural fine-grained entity type classification with hierarchy-aware loss. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 16–25, New Orleans, Louisiana. Association for Computational Linguistics.

Zenan Xu, Daya Guo, Duyu Tang, Qinliang Su, Linjun Shou, Ming Gong, Wanjun Zhong, Xiaojun Quan, Daxin Jiang, and Nan Duan. 2021. Syntax-enhanced pre-trained model. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5412–5422, Online. Association for Computational Linguistics.

Semih Yavuz, Izzeddin Gur, Yu Su, Mudhakar Srivatsa, and Xifeng Yan. 2016. Improving semantic parsing via answer type inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 149–159, Austin, Texas. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Wenpeng Yin, Nazneen Fatema Rajani, Dragomir Radev, Richard Socher, and Caiming Xiong. 2020. Universal natural language processing with limited annotations: Try few-shot textual entailment as a start. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8229–8239, Online. Association for Computational Linguistics.

Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding methods for fine grained entity type classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 291–296, Beijing, China. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.