Prompt-Learning for Fine-Grained Entity Typing

细粒度实体类型的即时学习

Abstract摘要

As an effective approach to tune pre-trained language models (PLMs) for specific tasks, prompt-learning has recently attracted much attention from researchers. By using clozestyle language prompts to stimulate the versatile knowledge of PLMs, prompt-learning can achieve promising results on a series of NLP tasks, such as natural language inference, sentiment classification, and knowledge probing. In this work, we investigate the application of prompt-learning on fine-grained entity typing in fully supervised, few-shot and zero-shot scenarios. We first develop a simple and effective prompt-learning pipeline by constructing entity-oriented verbalizer and templates and conducting masked language modeling. Further, to tackle the zero-shot regime, we propose a self-supervised strategy that carries out distribution-level optimization in prompt-learning to automatically summarize the information of entity types. Extensive experiments on three fine-grained entity typing benchmarks (with up to 86 classes) under fully supervised, few-shot and zero-shot settings show that prompt-learning methods significantly outperform fine-tuning baselines, especially when the training data is insufficient.

作为针对特定任务调整预训练语言模型 (PLM) 的有效方法，提示学习最近引起了研究人员的广泛关注。通过使用 clozestyle 语言提示来激发 PLM 的通用知识，提示学习可以在自然语言推理、情感分类和知识探测等一系列 NLP 任务上取得可喜的成果。在这项工作中，我们研究了提示学习在全监督、少样本和零样本场景中的细粒度实体类型中的应用。我们首先通过构建面向实体的语言表达器和模板并进行掩码语言建模来开发一个简单有效的提示学习管道。此外，为了解决零样本制度，我们提出了一种自监督策略，该策略在即时学习中进行分布级别优化，以自动汇总实体类型的信息。在完全监督、少样本和零样本设置下对三个细粒度实体类型基准（最多 86 个类）进行的大量实验表明，即时学习方法显着优于微调基线，尤其是在训练数据不足时。

1 Introduction

In recent years, pre-trained language models (PLMs) have been widely explored and become a key instrument for natural language understanding (Devlin et al., 2019; Liu et al., 2019) and generation (Radford et al., 2018; Raffel et al., 2020). By applying self-supervised learning on large-scale unlabeled corpora, PLMs can capture rich lexical (Jawahar et al., 2019), syntactic (Hewitt and Manning, 2019; Wang et al., 2021), and factual knowledge (Petroni et al., 2019) that well benefits downstream NLP tasks. Considering the versatile knowledge contained in PLMs, many efforts of researchers have been devoted to stimulating taskspecific knowledge in PLMs and adapting such knowledge to downstream NLP tasks. Fine-tuning with extra classifiers has been one typical solution for adapting PLMs to specific tasks and achieves promising results on various NLP tasks (Qiu et al., 2020; Han et al., 2021a).

近年来，预训练语言模型 (PLM) 得到了广泛的探索，并成为自然语言理解（Devlin 等人，2019 年；Liu 等人，2019 年）和生成（Radford 等人，2018 年； 拉菲尔等人，2020 年）。 通过在大规模未标记语料库上应用自监督学习，PLM 可以捕获丰富的词汇（Jawahar 等人，2019 年）、句法（Hewitt 和 Manning，2019 年；Wang 等人，2021 年）和事实知识（Petroni 等人，2019 年） ., 2019) 这对下游 NLP 任务有好处。 考虑到 PLM 中包含的通用知识，研究人员的许多努力都致力于激发 PLM 中的特定任务知识，并将这些知识应用于下游 NLP 任务。 使用额外分类器进行微调一直是使 PLM 适应特定任务的一种典型解决方案，并在各种 NLP 任务上取得了有希望的结果（Qiu 等人，2020 年；Han 等人，2021a）。

Some recent efforts on probing knowledge of PLMs show that, by writing some natural language prompts, we can induce PLMs to complete factual knowledge (Petroni et al., 2019). GPT-3 further utilizes the information provided by prompts to conduct few-shot learning and achieves awesome results (Brown et al., 2020). Inspired by this, prompt-learning has been introduced. As shown arXiv:2108.10604v1 [cs.CL] 24 Aug 2021 in Figure 1, in prompt-learning, downstream tasks are formalized as equivalent cloze-style tasks, and PLMs are asked to handle these cloze-style tasks instead of original downstream tasks. Compared with conventional fine-tuning methods, promptlearning does not require extra neural layers and intuitively bridges the objective form gap between pre-training and fine-tuning. Sufficient empirical analysis shows that, either for manually picking hand-crafted prompts (Liu et al., 2021b; Han et al., 2021b) or automatically building auto-generated prompts (Shin et al., 2020; Gao et al., 2020; Lester et al., 2021), taking prompts for tuning models is surprisingly effective for the knowledge stimulation and model adaptation of PLMs, especially in the low-data regime.

最近在探索 PLM 知识方面的一些努力表明，通过编写一些自然语言提示，我们可以诱导 PLM 完成事实知识（Petroni 等，2019）。 GPT-3 进一步利用提示提供的信息进行小样本学习并取得了惊人的结果（Brown 等，2020）。受此启发，引入了即时学习。如图 1 中 arXiv:2108.10604v1 [cs.CL] 24 Aug 2021 所示，在提示学习中，下游任务被形式化为等效的完形填空式任务，并且要求 PLM 处理这些完形填空式任务而不是原始下游任务.与传统的微调方法相比，即时学习不需要额外的神经层，直观地弥合了预训练和微调之间的目标形式差距。足够的实证分析表明，无论是手动挑选手工制作的提示（Liu 等人，2021b；Han 等人，2021b）还是自动构建自动生成的提示（Shin 等人，2020 年；Gao 等人，2020 年） ; Lester et al., 2021)，提示调整模型对于 PLM 的知识刺激和模型适应非常有效，尤其是在低数据情况下。

Intuitively, prompt-learning is applicable to finegrained entity typing, which aims at classifying marked entities from input sequences into specific types in a pre-defined label set. We discuss this topic with a motivating example, “He is from New York”. By adding a prompt with a masking token [MASK], the sentence becomes “He is from New York. In this sentence, New York is [MASK]”. Due to the wealth of knowledge acquired during pretraining, PLMs can compute a probability distribution over the vocabulary at the masked position, and a relatively higher probability with the word “city” than the word “person”. In other words, with simple prompts, the abstract entity attributes contained in PLMs can be efficiently exploited, which is meaningful for downstream entity-related tasks.

直观地说，提示学习适用于细粒度实体类型，其目的是将输入序列中的标记实体分类为预定义标签集中的特定类型。 我们用一个激励性的例子来讨论这个话题，“他来自纽约”。 通过添加带有屏蔽标记 [MASK] 的提示，句子变为“他来自纽约。 在这句话中，纽约是 [MASK]”。 由于在预训练期间获得了丰富的知识，PLM 可以计算掩码位置词汇表的概率分布，并且“city”这个词的概率比“person”这个词的概率要高。 换句话说，通过简单的提示，可以有效地利用 PLM 中包含的抽象实体属性，这对于下游实体相关的任务是有意义的。

In this work, we comprehensively explore the application of prompt-learning to fine-grained entity typing in fully supervised, few-shot and zeroshot settings. Particularly, we first introduce a naive pipeline, where we construct entity-oriented prompts and formalize fine-grained entity typing as a cloze-style task. This simple pipeline yields promising results in our experiments, especially when supervision is insufficient. Then, to tackle the zero-shot scenario where no explicit supervision exists in training, we develop a self-supervised strategy under our prompt-learning pipeline. Our self-supervised strategy attempts to automatically summarize entity types by optimizing the similarity of the predicted probability distributions of paired examples in prompt-learning.

在这项工作中，我们全面探索了提示学习在全监督、少样本和零样本设置中的细粒度实体类型中的应用。 特别是，我们首先引入了一个简单的管道，在那里我们构建了面向实体的提示，并将细粒度的实体类型化为一个完形填空式的任务。 这个简单的管道在我们的实验中产生了有希望的结果，尤其是在监督不足的情况下。 然后，为了解决在训练中不存在明确监督的零样本场景，我们在我们的即时学习管道下开发了一种自我监督的策略。 我们的自监督策略试图通过优化提示学习中成对示例的预测概率分布的相似性来自动总结实体类型。

Three popular benchmarks are used for our experiments, including FEW-NERD (Ding et al., 2021b), OntoNotes (Weischedel et al., 2013), BBN (Weischedel and Brunstein, 2005). All these datasets have a complex type hierarchy consisting of rich entity types, requiring models to have good capabilities of entity attribute detection. Empirically, our method yields significant improvements on these benchmark datasets, especially under the zero-shot and few-shot settings. We also make an analysis and point out both the superiority and bottleneck of prompt-learning in fine-grained entity typing, which may advance further efforts to extract entity attributes using PLMs. Our source code and pre-trained models will be publicly available.

我们的实验使用了三个流行的基准，包括 FEW-NERD (Ding et al., 2021b)、OntoNotes (Weischedel et al., 2013)、BBN (Weischedel and Brunstein, 2005)。 所有这些数据集都具有复杂的类型层次结构，包含丰富的实体类型，要求模型具有良好的实体属性检测能力。 根据经验，我们的方法对这些基准数据集产生了显着的改进，尤其是在零样本和少样本设置下。 我们还进行了分析并指出了细粒度实体类型中即时学习的优势和瓶颈，这可能会进一步推动使用 PLM 提取实体属性的努力。 我们的源代码和预训练模型将公开提供。

2 Background

In this section, we first give a problem definition of the entity typing task (§ 2.1), followed by an introduction of conventional vanilla fine-tuning (§ 2.2) and prompt-based tuning (§ 2.3) with PLMs.

在本节中，我们首先给出实体输入任务的问题定义（第 2.1 节），然后介绍使用 PLM 的传统 vanilla 微调（第 2.2 节）和基于提示的调优（第 2.3 节）。

2.1 Problem Definition

The input of entity typing is a dataset D = {x1, ..., xn} with n sentences, and each sentence x contains a marked entity mention m. For each input sentence x, entity typing aims at predicting the entity type y ∈ Y of its marked mention m, where Y is a pre-defined set of entity types. Entity typing is typically regarded as a context-aware classification task. For example, in the sentence “London is the fifth album by the rock band Jesus Jones...”, the entity mention London should be classified as Music rather than Location. In the era of PLMs, using pre-trained neural language models (e.g. BERT) as the encoder and performing model tuning for classifying types becomes a standard paradigm.

实体类型的输入是一个数据集 D = {x1, ..., xn} 有 n 个句子，每个句子 x 包含一个标记的实体mention m。 对于每个输入句子 x，实体类型旨在预测其标记提及 m 的实体类型 y ∈ Y，其中 Y 是一组预定义的实体类型。 实体类型通常被视为上下文感知分类任务。 例如，在句子“伦敦是耶稣琼斯摇滚乐队的第五张专辑...”中，实体提及伦敦应归类为音乐而不是位置。 在 PLM 时代，使用预训练的神经语言模型（例如 BERT）作为编码器并执行模型调整以进行类型分类成为标准范式。

2.2 Vanilla Fine-tuning

In the vanilla fine-tuning paradigm of entity typing, for each token ti in an input sequence x = {[CLS], t1, . . . , m, . . . , tT , [SEP]} with a marked entity mention m = {ti , . . . , tj}, the PLM M produces its contextualized representation {h[CLS], h1, . . . , hT , h[SEP]}. Empirically, we choose the embedding of the [CLS] token, h[CLS], as the final representation that is fed into an output layer to predict the probability distribution over the label space



where W and b are learnable parameters. W, b and all parameters of PLMs are tuned by maximizing the objective function 1 n Pn i=1 log(P(yi |si)), where yi is the golden type label of si .

在实体类型的普通微调范式中，对于输入序列中的每个标记 ti x = {[CLS], t1, ... . . , 米, . . . , tT , [SEP]} 带有标记的实体提及 m = {ti , . . . , tj}，PLM M 生成其上下文化表示 {h[CLS], h1, ... . . , hT , h[SEP]}。 根据经验，我们选择 [CLS] 标记 h[CLS] 的嵌入作为输入输出层的最终表示，以预测标签空间上的概率分布，其中 W 和 b 是可学习参数。 W、b 和 PLM 的所有参数通过最大化目标函数 1 n Pn i=1 log(P(yi |si)) 来调整，其中 yi 是 si 的黄金类型标签。

2.3 Prompt-based Tuning

In prompt-based tuning, for each label y ∈ Y, we define a label word set Vy = {w1, . . . , wm}. Vy is a subset of the vocabulary V of the PLM M, i.e., Vy ⊆ V. By taking the union of the dictionary corresponding to each label, we get an overall dictionary V ∗ . For example, in sentiment classification, we could map the label y = POSITIVE into a set Vy = {great, good, wonderful...}. And another primary component of prompt-learning is a prompt template T(·), which modifies the original input x into a prompt input T(x) by adding a set of additional tokens at the end of x. Conventionally, a [MASK] token is added for PLMs to predict the missing label word w ∈ V∗ . Thus, in promptlearning, a classification problem is transferred into a masked language modeling problem,



在基于提示的调优中，对于每个标签 y ∈ Y，我们定义了一个标签词集 Vy = {w1, . . . , wm}。 Vy 是 PLM M 的词汇表 V 的子集，即 Vy ⊆ V。 通过取每个标签对应的字典的并集，我们得到一个整体字典 V ∗ 。 例如，在情感分类中，我们可以将标签 y = POSITIVE 映射到集合 Vy = {great, good, beautiful...}。 提示学习的另一个主要组成部分是提示模板 T(·)，它通过在 x 的末尾添加一组附加标记将原始输入 x 修改为提示输入 T(x)。 传统上，为 PLM 添加 [MASK] 标记以预测丢失的标签词 w ∈ V∗ 。 因此，在提示学习中，分类问题被转换为掩码语言建模问题

3 Prompt-learning for Entity Typing: A Naive Pipeline

After transferred into masked language modeling, the prompt-learning method is applicable to learning and aggregating type information of entities. In this section, we first introduce a naive but empirically strong baseline that utilizes prompts to extract entity types with explicit supervision, including the construction of label words (§ 3.1), templates (§ 3.2) and training (§ 3.3). And such a simple pipeline yields remarkable results on three benchmark datasets. Then we propose a self-supervised prompt-learning method that automatically learns type information from unlabeled data (§ 4).

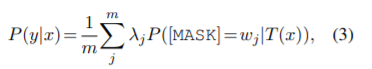
转移到掩码语言建模后，提示学习方法适用于学习和聚合实体的类型信息。 在本节中，我们首先介绍一个朴素但经验丰富的基线，它利用提示提取具有显式监督的实体类型，包括标签词的构建（第 3.1 节）、模板（第 3.2 节）和训练（第 3.3 节）。 这样一个简单的管道在三个基准数据集上产生了显着的结果。 然后我们提出了一种自监督的提示学习方法，可以从未标记的数据中自动学习类型信息（第 4 节）。

3.1 Label Words Set V ∗

For fine-grained entity typing, datasets usually use hierarchical label space such as PERSON/ARTIST (FEW-NERD) and ORGANIZATION/PARTY (OntoNotes). In this case, we use all the words as the label words set V ∗ for this entity type. For example, y = LOCATION/CITY → v = {location, city}. And as the entity types are all well-defined nouns with clear boundaries, it is intuitive to expand the label words set V ∗ with obtainable related nouns. For example, in Related Words1 , the top-10 related words of the label word city is “metropolis, town, municipality, urban, suburb, municipal, megalopolis, civilization, downtown, country”. These words are strongly related to the class CITY, and they are hardly mapped to other entity types even under the same LOCATION class, such as LOCATION/MOUNTAIN, LOCATION/ISLAND, etc.

对于细粒度的实体类型，数据集通常使用分层标签空间，例如 PERSON/ARTIST (FEW-NERD) 和 ORGANIZATION/PARTY (OntoNotes)。 在这种情况下，我们使用所有词作为该实体类型的标签词集 V\*。 例如，y = LOCATION/CITY → v = {location, city}。 并且由于实体类型都是定义明确的名词，边界清晰，用可获得的相关名词扩展标签词集V\*是很直观的。 例如，在Related Words1 中，标签词city 的前10 个相关词是“metropolis、town、 Municipality、urban、郊区、city、megalopolis、civities、downtown、country”。 这些词与CITY类密切相关，即使在同一LOCATION类下，它们也很难映射到其他实体类型，例如LOCATION/MOUNTAIN、LOCATION/ISLAND等。

In masked language modeling, we use confidence scores of all the words in Vy to construct the final score of the particular type y. That is, for an input x (which is mapped to T(x)) and its entity type y (which is mapped to Vy = {w1, ..., wm}), the conditional probability becomes



where λi is a parameter to indicate the importance of the current word wj ∈ Vy. Note that λi could also be learnable or heuristically defined during the training procedure.

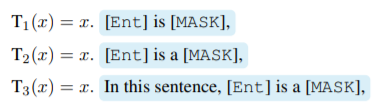
在掩码语言建模中，我们使用 Vy 中所有单词的置信度分数来构建特定类型 y 的最终分数。 也就是说，对于输入 x（映射到 T(x)）及其实体类型 y（映射到 Vy = {w1, ..., wm}），条件概率变为 其中 λi 是一个参数，表示当前词 wj ∈ Vy 的重要性。 请注意，λi 也可以在训练过程中学习或启发式定义。

3.2 Templates

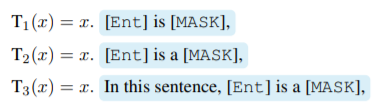
In this section, we construct entity-oriented prompts for the fine-grained entity typing task. We choose hard-encoding templates with natural language and soft-encoding templates with additional special tokens in our work.

在本节中，我们为细粒度实体键入任务构建面向实体的提示。 我们在工作中选择带有自然语言的硬编码模板和带有额外特殊标记的软编码模板。

For the choice of hard-encoding templates, we do not use automatic searching methods for discrete prompts since the fine-grained entity typing task is clearly defined and the prompts are easily purposeful. We select simple declarative templates rather than hypernym templates to avoid grammartical errors. In the template of hard encoding setting, we first copy the marked entity mention in x, then we add a few linking verbs and articles followed by the [MASK] token. With the marked entity mention [Ent], we use the following templates:



对于硬编码模板的选择，我们不使用离散提示的自动搜索方法，因为细粒度的实体输入任务定义明确，提示很容易有目的。 我们选择简单的声明性模板而不是上位词模板来避免语法错误。 在硬编码设置的模板中，我们首先复制 x 中标记的实体mention，然后添加一些链接动词和文章，然后是 [MASK] 标记。 使用标记实体提及 [Ent]，我们使用以下模板：



where [Ent] is the entity mention in x. In § 5, we report the the results of T3(·).

其中 [Ent] 是 x 中的实体提及。 在第 5 节中，我们报告了 T3(·)的结果。

We also adopt the soft-encoding strategy, which introduces some additional special tokens [P1], ..., [Pl ] as the template, where l is a predefined hyper-parameter. The template begins with a delimiter [P] and a copy of the entity mention [M]. The complete template becomes:

我们还采用了软编码策略，它引入了一些额外的特殊标记 [P1], ..., [Pl] 作为模板，其中 l 是一个预定义的超参数。 模板以分隔符 [P] 和实体提及 [M] 的副本开头。 完整的模板变为：



where each embedding of prompts is randomly initialized and optimized during training. Intuitively, these special tokens can represent a cluster of words with similar semantics in the vocabulary.

其中每个提示嵌入在训练期间随机初始化和优化。 直观地，这些特殊标记可以表示词汇表中具有相似语义的一组单词。

3.3 Training and Inference

The strategies of hard or soft encoding provide different initialization of templates, and they both can be parameterized by φ and optimized along with M during training. We train the pre-trained model M (parameterized by θ) along with the additional prompt embeddings by using the cross-entropy loss function:

硬编码或软编码的策略提供不同的模板初始化，它们都可以通过 φ 参数化并在训练过程中与 M 一起优化。 我们使用交叉熵损失函数训练预训练模型 M（由 θ 参数化）以及附加提示嵌入：



For inference, we can directly use Eq. 3 to predict the label of the current input instance based on the predicted words of the [MASK] position.

对于推理，我们可以直接使用方程。 3 根据[MASK]位置的预测词预测当前输入实例的标签。

This pipeline could be applied to entity typing task with explicit supervision, and it is effective even if the training data are insufficient, i.e., the few-shot scenario (§ 5.5). Naturally, we consider a more extreme situation, that is, a scenario without any training data (zero-shot scenario). In this setting, if we directly use an additional classifier to predict the label, the result is equivalent to random guessing, because the parameters of the classifier are randomly initialized. If we use prompts to infer the label based on the predicted words, although its performance is significantly better than guessing, there will also be a catastrophic decline (§ 5.6). At this time, a question emerges: “Is it possible for PLMs to predict entity types without any explicit supervision? ”

该管道可以应用于具有显式监督的实体输入任务，即使训练数据不足，即少样本场景（第 5.5 节）也有效。 自然，我们考虑更极端的情况，即没有任何训练数据的场景（零样本场景）。 在这种设置下，如果我们直接使用一个额外的分类器来预测标签，结果相当于随机猜测，因为分类器的参数是随机初始化的。 如果我们根据预测词使用提示来推断标签，虽然其性能明显优于猜测，但也会出现灾难性的下降（§5.6）。 这时候就出现了一个问题：“PLM 是否可以在没有任何明确监督的情况下预测实体类型？ ”

4 Self-supervised Prompt-learning for Zero-shot Entity Typing

With prompt-learning, the answer is yes, because in the pre-training stage, the contexts of entities have already implied the corresponding type information, which provides an advantageous initialization point for the prompt-learning paradigm. For example, in the input sentence with the T3(·) template: “Steve Jobs found Apple. In this sentence, Steve Jobs is a [MASK] ”. In our observations, the probability of PLMs predicting person at the masked position will be significantly higher than the probability of location. And if we make reasonable use of this superior initialization point, it is possible for PLMs to automatically summarize the type information, and finally extract the correct entity type.

对于提示学习，答案是肯定的，因为在预训练阶段，实体的上下文已经暗示了相应的类型信息，这为提示学习范式提供了有利的初始化点。 例如，在带有 T3(·) 模板的输入句子中：“Steve Jobs found Apple. 在这句话中，史蒂夫乔布斯是一个[MASK]”。 在我们的观察中，PLM 预测蒙面位置的人的概率将显着高于位置的概率。 而如果我们合理利用这个优越的初始化点，PLMs就有可能自动汇总类型信息，最终提取出正确的实体类型。

4.1 Overview

In order to create conditions for PLMs to summarize entity types, we consider a self-supervised paradigm that optimizes the similarity of the probability distribution predicted by similar examples over a projected vocabulary V ∗ . To achieve that in prompt-learning, we need to (1) impose a limit on the prediction range of the model, so that only those words that we need, that is, words that express entity types, participate in the optimization of the gradient; (2) provide an unlabeled dataset, where entity mentions are marked without any types to allow the model to learn the process of inducing type information in a self-supervised manner. The inputs contain a pre-trained model M, a pre-defined label schema Y, and a dataset without labels D = {x1, ..., xn} (entity mentions are marked without any types). our goal is to make M capable to automatically carry out zero-shot entity typing after trained on D and Y. Using promptlearning as the training strategy, we first construct a label words set V ∗ from Y, and for each sentence x in D, we wrap it with hard-encoding template with a [MASK] symbol. The key idea is to make the prediction distributions of the same type of entities on V ∗ as similar as possible. In this way, we can perform contrastive learning by sampling positive and negative examples, while ignoring the impact of other words that are not in V ∗ on optimization during the MLM process.

为了为 PLM 总结实体类型创造条件，我们考虑了一种自监督范式，该范式优化了由相似示例预测的概率分布在投影词汇 V\* 上的相似性。为了在提示学习中实现这一点，我们需要（1）对模型的预测范围施加限制，以便只有我们需要的那些词，即表达实体类型的词，参与梯度的优化; (2) 提供一个未标记的数据集，其中实体mention 被标记为没有任何类型，以允许模型以自监督的方式学习诱导类型信息的过程。输入包含一个预训练的模型 M、一个预定义的标签模式 Y 和一个没有标签的数据集 D = {x1, ..., xn}（实体提及被标记为没有任何类型）。我们的目标是让 M 在 D 和 Y 上训练后能够自动进行零样本实体打字。使用提示学习作为训练策略，我们首先从 Y 构建一个标签词集 V\*，对于 D 中的每个句子 x，我们用带有 [MASK] 符号的硬编码模板包装它。关键思想是使同类型实体在 V\* 上的预测分布尽可能相似。这样，我们可以通过对正负示例进行采样来进行对比学习，同时忽略不在 V\* 中的其他词对 MLM 过程中优化的影响。

4.2 Self-supervised Learning

Although there are no labels in D, we can still develop a sampling strategy based on a simple hypothesis, that is, same entities in different sentences have similar types. For instance, we will sample two sentences contain “Steve Jobs” as a positive pair. Moreover, considering entity typing is context-aware, “Steve Jobs” could be entrepreneur, designer, philanthropist in different contexts, we choose to optimize the similarity between distributions of the words over V ∗ . This strategy not only softens the supervision, but also eliminates the impact of other words in self-supervised learning.

尽管 D 中没有标签，但我们仍然可以基于一个简单的假设制定抽样策略，即不同句子中的相同实体具有相似的类型。 例如，我们将采样包含“Steve Jobs”的两个句子作为正对。 此外，考虑到实体类型是上下文感知的，“史蒂夫乔布斯”在不同的上下文中可能是企业家、设计师、慈善家，我们选择优化单词在 V\* 上的分布之间的相似性。 这种策略不仅软化了监督，而且消除了自监督学习中其他词的影响。

Particularly, we randomly sample c positive pairs, i.e., sentence pairs that share one same entity mention, denoted as Dˆ pos, and c negative pairs, i.e., two sentences with different entity mentions marked, denoted as Dˆ neg from a large-scale entitylinked corpus D. To avoid generating false negative samples, the negative samples are further restricted by a large dictionary that contains common entities and their type information. Only sentence pairs with entities of different types in the dictionary are selected as negative samples. Then we wrap them with hard-encoding T3(·). To avoid overfitting of the entity names, we randomly hide the entity mention (in the original input and the template) with a special symbol [Hide] with a probability of α. Empirically, α is set to 0.4.

特别地，我们从一个大规模的entitylinked中随机抽取了c对正对，即共享一个相同实体mention的句子对，表示为Dˆpos，以及c对负对，即标记了不同实体mention的两个句子，表示为Dˆneg 语料库 D. 为了避免产生假负样本，负样本进一步被包含公共实体及其类型信息的大字典限制。 仅选择字典中具有不同类型实体的句子对作为负样本。 然后我们用硬编码 T3(·) 包裹它们。 为了避免实体名称的过度拟合，我们使用概率为 α 的特殊符号 [Hide] 随机隐藏实体提及（在原始输入和模板中）。 根据经验，α 设置为 0.4。

Since the impact of a pair of examples on training should be measured at the distribution level, we choose Jensen-Shannon divergence as a metric to assess the similarity of two distributions. Thus, in a sentence pair (x, x 0 ), the similarity score of two representations of the the predictions h and h 0 of the [MASK] position is computed by:

由于一对示例对训练的影响应该在分布级别进行衡量，因此我们选择 Jensen-Shannon 散度作为评估两个分布相似性的指标。 因此，在一个句子对 (x, x 0 ) 中，[MASK] 位置的预测 h 和 h 0 的两个表示的相似性分数计算如下：

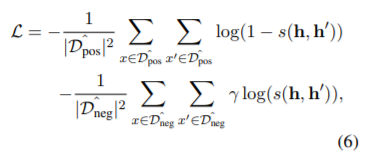


where JS is Jensen-Shannon divergence, PV∗ (w|x) and PV∗ (w|x 0 ) are probability distributions of the predicting token w over V ∗ obtained by h and h 0

其中 JS 是 Jensen-Shannon 散度，PV∗ (w|x) 和 PV∗ (w|x 0 ) 是通过 h 和 h 0 获得的预测标记 w 对 V ∗ 的概率分布

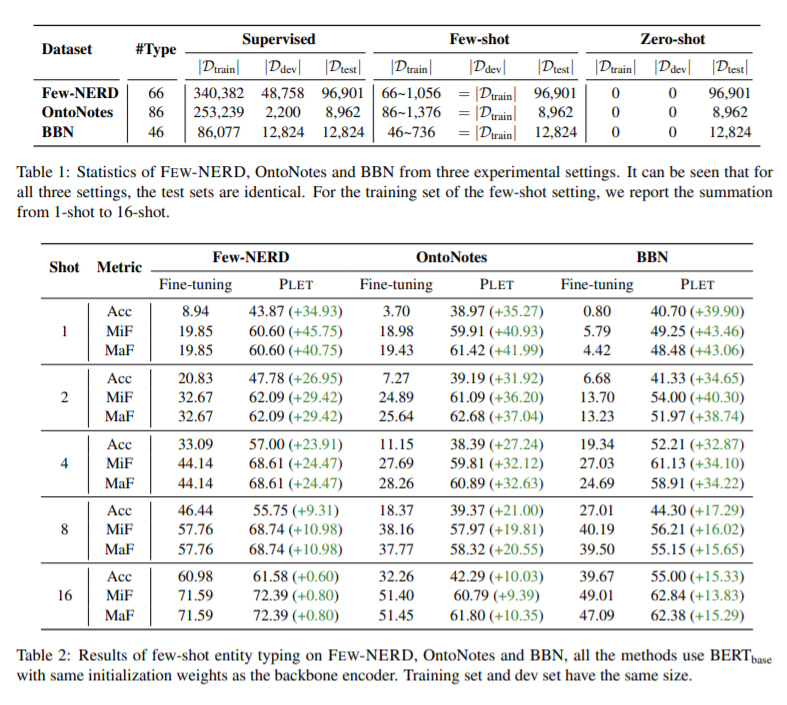
As we attempt to make the predictions of the positive pairs similar, the objective is computed by:

当我们试图使正对的预测相似时，目标是通过以下方式计算的：



where γ is a penalty term, because the assumption is loose in negative pairs. Overall, we use entitylinked English Wikipedia corpus as the raw data and generate about 1 million pairs of data each as Dˆ pos and Dˆ neg.

总的来说，我们使用实体链接的英语维基百科语料库作为原始数据，并生成大约 100 万对数据，分别作为 D^ pos 和 D^ neg。



5 Experiments

In this section, we conduct experiments to evaluate the effectiveness of our methods. We use FT to denote the BERT-based fine-tuning approach, PLET to denote the naive prompt-learning approach for entity typing in § 3, and PLET (S) to denote the self-supervised prompt-learning approach in § 4. Our experiments are carried out on fully supervised (§ 5.4), few-shot (§ 5.5) and zero-shot (§ 5.6) settings on three fine-grained entity typing datasets.

在本节中，我们进行实验以评估我们方法的有效性。 我们使用 FT 表示基于 BERT 的微调方法，PLET 表示第 3 节中实体输入的幼稚提示学习方法，以及第 4 节中的 PLET (S) 表示自监督提示学习方法。我们的 实验是在三个细粒度实体类型数据集的完全监督（第 5.4 节）、少样本（第 5.5 节）和零样本（第 5.6 节）设置上进行的。

5.1 Datasets

We use three fine-grained entity typing datasets: FEW-NERD, OntoNotes, and BBN.

我们使用三个细粒度的实体类型数据集：FEW-NERD、OntoNotes 和 BBN。

FEW-NERD. We use FEW-NERD (Ding et al., 2021b) as the main dataset, which has the following advantages: (1) FEW-NERD is large-scale and fine-grained, which contains 8 coarse-grained and 66 fine-grained entity types. (2) FEW-NERD is manually annotated, thereby we can precisely assess the capability of entity typing models. Specifically, we use the supervised setting of the dataset, FEW-NERD (SUP), and the official split of it to conduct our experiments.

很少-书呆子。 我们使用 FEW-NERD (Ding et al., 2021b) 作为主要数据集，它具有以下优点： (1) FEW-NERD 是大规模细粒度的，包含 8 个粗粒度和 66 个细粒度 实体类型。 (2) FEW-NERD 是手动标注的，因此我们可以精确评估实体类型模型的能力。 具体来说，我们使用数据集的监督设置 FEW-NERD (SUP) 和它的官方拆分来进行我们的实验。

OntoNotes. We also use the OntoNotes 5.0 dataset (Weischedel et al., 2013) in experiments. Following previous works for fine-grained entity typing, we adopt 86-classes version of OntoNotes, while each class has at most 3 levels of the type hierarchy. And the data split is identical to (Shimaoka et al., 2017).

OntoNotes。 我们还在实验中使用了 OntoNotes 5.0 数据集（Weischedel 等人，2013 年）。 继之前的细粒度实体类型工作之后，我们采用了 OntoNotes 的 86 类版本，而每个类最多有 3 级类型层次结构。 并且数据拆分与 (Shimaoka et al., 2017) 相同。

BBN. BBN dataset is selected from Penn Treebank corpus of Wall Street Journal texts and labeled by (Weischedel and Brunstein, 2005). We follow the version processed by (Ren et al., 2016a), and the data split by (Ren et al., 2016b). The dataset contains 46 types and each type has a maximum type hierarchy level of 2.

BBN。 BBN 数据集选自《华尔街日报》文本的 Penn Treebank 语料库，并由 (Weischedel and Brunstein, 2005) 标记。 我们遵循 (Ren et al., 2016a) 处理的版本，以及 (Ren et al., 2016b) 分割的数据。 该数据集包含 46 种类型，每种类型的最大类型层次级别为 2。

5.2 Experimental Settings

The experiments are performed under three different settings to evaluate the effect of the promptlearning method and semi-supervised training. In table 1, we show the statistics of all the settings on the three datasets.

实验在三种不同的设置下进行，以评估即时学习方法和半监督训练的效果。 在表 1 中，我们显示了三个数据集上所有设置的统计数据。

Supervised Setting. In a fully supervised setting, all training data are used in the training phase. FT and PLET are used to train the model. We run the experiments on all three datasets with BERTbase-cased backbone. Both hard and soft encodings are used for PLET.

监督设置。 在完全监督的设置中，所有训练数据都用于训练阶段。 FT 和 PLET 用于训练模型。 我们使用 BERTbase-cased 主干在所有三个数据集上运行实验。 硬编码和软编码都用于 PLET。

Few-shot Setting. In a few-shot setting, we randomly sample 1, 2, 4, 8, 16 instances for each entity type for training. We apply both FT and PLET methods with hard encoding on all the three datasets.

少镜头设置。 在少样本设置中，我们为每个实体类型随机抽取 1、2、4、8、16 个实例进行训练。 我们在所有三个数据集上都应用了硬编码的 FT 和 PLET 方法。

Zero-shot Setting. In zero-shot setting, no training data with labels are available. The model is required to infer the entity type without any supervised training. Since fine-tuning is not applicable in this setting, we only conduct experiments on PLET and PLET (S).

零射击设置。 在零样本设置中，没有带标签的训练数据可用。 该模型需要在没有任何监督训练的情况下推断实体类型。 由于微调不适用于此设置，因此我们仅对 PLET 和 PLET (S) 进行实验。

Metrics. In terms of evaluation metrics, we follow the widely used setting of Ling and Weld (2012), which includes strict accuracy (Acc), loose macro F1-score (MaF) and loose micro F1-score (MiF) to evaluate the performances of models. The loose F1-score calculation concerns type labels by different granularities.

指标。 在评价指标方面，我们沿用了 Ling and Weld (2012) 广泛使用的设置，包括严格精度 (Acc)、松散宏观 F1-score (MaF) 和松散微观 F1-score (MiF) 来评估 楷模。 松散的 F1 分数计算涉及不同粒度的类型标签。

5.3 Experimental Details

We use BERT-base (Devlin et al., 2019) as the backbone structures of our model and initialized with the corresponding pre-trained cased weights2 . The hidden sizes are 768, and the number of layers are 12. Models are implemented by Pytorch framework3 (Paszke et al., 2019) and Huggingface transformers4 (Wolf et al., 2020). BERT models are optimized by AdamW (Loshchilov and Hutter, 2019) with the learning rate of 5e-5. The training batch size used is 16 for all models. In the supervised setting, each model is trained for 10 epochs and evaluated on the dev set every 2000 steps. In the few-shot setting, each model is trained for 30 epochs and evaluated every 10∼50 steps, each time the evaluation is run for 200 steps. For the methods with hard-encoding, we report the experimental results of T3(·). For the soft-encoding method, we report the results of m = 2. Experiments are conducted with CUDA on NVIDIA Tesla V100 GPUs.

我们使用 BERT-base (Devlin et al., 2019) 作为我们模型的主干结构，并使用相应的预训练案例权重进行初始化。 隐藏大小为 768，层数为 12。模型由 Pytorch 框架 3 (Paszke et al., 2019) 和 Huggingface Transformers4 (Wolf et al., 2020) 实现。 BERT 模型由 AdamW（Loshchilov 和 Hutter，2019 年）优化，学习率为 5e-5。 所有模型使用的训练批量大小为 16。 在监督设置中，每个模型都训练 10 个 epoch，并每 2000 步在开发集上进行评估。 在少样本设置中，每个模型训练 30 个 epoch，每 10~50 步评估一次，每次评估运行 200 步。 对于硬编码的方法，我们报告了T3(·)的实验结果。 对于软编码方法，我们报告了 m = 2 的结果。在 NVIDIA Tesla V100 GPU 上使用 CUDA 进行了实验。

5.4 Results of Fully Supervised Entity Typing

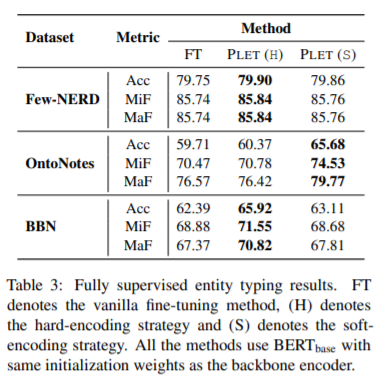


Table 3: Fully supervised entity typing results. FT denotes the vanilla fine-tuning method, (H) denotes the hard-encoding strategy and (S) denotes the softencoding strategy. All the methods use BERTbase with same initialization weights as the backbone encoder.

表 3：完全监督的实体类型结果。 FT 表示普通微调方法，(H) 表示硬编码策略，(S) 表示软编码策略。 所有方法都使用与主干编码器具有相同初始化权重的 BERTbase。

The results on all three datasets across different models are reported in Table 3. Overall, the promptbased methods have shown certain improvements comparing to directly fine-tuned models. It shows that the prompt-based method does help with capturing entity-type information from a given context.

表 3 报告了不同模型的所有三个数据集的结果。总体而言，与直接微调模型相比，基于提示的方法显示出一定的改进。 它表明基于提示的方法确实有助于从给定的上下文中捕获实体类型信息。

It is also observed that the magnitude of the improvement and the preference of prompt encoding strategy may vary with different datasets. The prompt-based method seems less effective on FEWNERD dataset than the other two. It indicates that the effect of the prompt-based method partially depends on the characteristics of the dataset and that different prompt designs may suit different data. Specifically, FEW-NERD is manually annotated and contains much less noise than the other two datasets, benefiting the FT method to learn classification with an extra linear layer. Moreover, for the OntoNotes dataset, soft encoding significantly outperforms hard encoding, while for the other two datasets the effect seems reversed.

还观察到，改进的幅度和提示编码策略的偏好可能因不同的数据集而异。 在 FEWNERD 数据集上，基于提示的方法似乎不如其他两种方法有效。 这表明基于提示的方法的效果部分取决于数据集的特性，不同的提示设计可能适合不同的数据。 具体来说，FEW-NERD 是手动注释的，并且比其他两个数据集包含的噪声要少得多，这有利于 FT 方法通过额外的线性层来学习分类。 此外，对于 OntoNotes 数据集，软编码明显优于硬编码，而对于其他两个数据集，效果似乎相反。

5.5 Results of Few-shot Entity Typing

Table 2 shows the results on few-shot entity typing. It is shown that prompt-based model outperforms fine-tuning by a large margin under few-shot setting, especially when only 1 ∼ 2 training instances per type are available. It should be noted

表 2 显示了少样本实体类型的结果。 结果表明，在少样本设置下，基于提示的模型在很大程度上优于微调，尤其是当每种类型只有 1 ∼ 2 个训练实例可用时。 应该注意

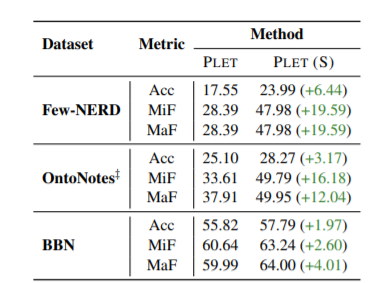


Table 4: Results of zero-shot entity typing on FEWNERD, OntoNotes, and BBN. ‡ means that we remove the “Other” class during testing. PLET denotes the prompt-learning pipeline and PLET (S) denotes self-supervised prompt-learning, both methods use the BERTbase as the backbone encoder.

表 4：在 FEWNERD、OntoNotes 和 BBN 上进行零样本实体输入的结果。 ‡ 表示我们在测试期间删除了“Other”类。 PLET 表示提示学习管道，PLET (S) 表示自监督提示学习，两种方法都使用 BERTbase 作为主干编码器。

that for OntoNotes and BBN datasets, sampling 16 instances for each entity type already amounts to over 0.5% of the total training data. Meanwhile, some of the data in BBN are distantly-supervised and are potentially erroneous. It brings more randomness to few-shot training. The results support the idea that a well-designed prompt has much potential in mining the learned knowledge in pretrained models and thus yields better performance in few-shot settings. The results also indicate that even when the number of entity types is large (46 ∼ 86), the superiority of prompt-learning still holds.

对于 OntoNotes 和 BBN 数据集，每种实体类型采样 16 个实例已经占总训练数据的 0.5% 以上。 同时，BBN 中的一些数据是远程监督的，可能是错误的。 它为少样本训练带来了更多的随机性。 结果支持这样一种观点，即精心设计的提示在挖掘预训练模型中学到的知识方面具有很大潜力，从而在少样本设置中产生更好的性能。 结果还表明，即使实体类型的数量很大（46 ∼ 86），即时学习的优势仍然存在。

5.6 Results of Zero-shot Entity Typing

Table 4 shows the results on zero-shot entity typing task on FEW-NERD dataset. We did not report the performance of the vanilla fine-tuning approach because it cannot produce reasonable results with a randomly initialized classifier. And it also should be noted that the prompt method without fine-tuning already outperforms random guessing. It indicates that adding a prompt is informative for a model pre-trained on masked-language-model task (e.g. BERT) and can induce reasonable predictions in entity typing tasks. Second, the performance of the model improves by a large margin if trained on unlabeled data. It shows the effectiveness of the proposed self-supervised training approach and points to the potential of a pre-trained prompt-based model under the zero-shot setting when no labeled data are available.

表 4 显示了 FEW-NERD 数据集上零样本实体类型任务的结果。 我们没有报告 vanilla 微调方法的性能，因为它无法通过随机初始化的分类器产生合理的结果。 还需要注意的是，没有微调的提示方法已经优于随机猜测。 它表明添加提示对于在掩码语言模型任务（例如 BERT）上预训练的模型是有用的，并且可以在实体输入任务中引起合理的预测。 其次，如果在未标记的数据上训练，模型的性能会有很大的提高。 它显示了所提出的自监督训练方法的有效性，并指出了在没有标记数据可用时在零样本设置下预训练的基于提示的模型的潜力。

To explore the more subtle changes in performance, we carry out case study for the zero-shot entity typing. In Figure 4, we illustrate the zeroshot prediction distribution (the correct prediction and other top-5 predictions) for four entity types in FEW-NERD, which are ORG-SPORTSTEAM, EVENT-ATTACK, MISC-CURRENCY and LOCMOUNTAIN. We could observe that with selfsupervised prompt-learning, PLET (S) could summarize entity type information and infer the related words to a certain extent. In Figure 4 (a) and Figure 4 (b), the PLET model suffers from a severe bias and almost predict no correct labels in the zero-shot setting since such words are low-frequency. And although there is no explicit supervision in the pretraining stage of UNPLET, the model could still find the corresponding words that express the ORGSPORTSLEAGUE and the EVENT-ATTACK types. In Figure 4 (c), self-supervised learning increases the performance of the original encoder. Further, in Figure 4 (d), PLET has been able to make satisfying predictions for this type LOC-MOUNTAIN. In this case, the use of self-supervised learning has hardly weakened the performance, which means that the process of automatically summarizing type information has a little negative impact on highconfidence entity types.

为了探索更细微的性能变化，我们对零样本实体类型进行了案例研究。在图 4 中，我们说明了 FEW-NERD 中四种实体类型（ORG-SPORTSTEAM、EVENT-ATTACK、MISC-CURRENCY 和 LOCMOUNTAIN）的零样本预测分布（正确预测和其他前 5 名预测）。我们可以观察到，通过自监督提示学习，PLET（S）可以在一定程度上总结实体类型信息并推断相关词。在图 4 (a) 和图 4 (b) 中，PLET 模型存在严重的偏差，并且在零样本设置中几乎无法预测正确的标签，因为这些词是低频的。并且尽管在 UNPLET 的预训练阶段没有明确的监督，该模型仍然可以找到表达 ORGSPORTSLEAGUE 和 EVENT-ATTACK 类型的相应单词。在图 4 (c) 中，自监督学习提高了原始编码器的性能。此外，在图 4（d）中，PLET 已经能够对这种类型的 LOC-MOUNTAIN 做出令人满意的预测。在这种情况下，使用自监督学习几乎没有削弱性能，这意味着自动汇总类型信息的过程对高置信度实体类型有一点负面影响。

5.7 Effect of Templates

As stated in previous studies (Gao et al., 2020; Zhao et al., 2021), the choice of templates may have a huge impact on the performance in promptlearning. In this section, we carry out experiments to investigate such influence. Experiments are conducted under the 8-shot setting on FEWNERD dataset, and we use 3 different hard encoding templates and 4 soft encoding templates (by changing the number of prompt tokens m). The results demonstrate that the choice of templates exerts a considerable influence on the performance of prompt-based few-shot learning. For the hard-encoding templates, the phrase that describes the location “in this sentence” contributes a remarkable improvement in performance. For the soft-encoding templates, surprisingly, the promptlearning model yields the best result with the fewest special tokens.

正如之前的研究（Gao 等人，2020 年；Zhao 等人，2021 年）所述，模板的选择可能对即时学习的表现产生巨大影响。 在本节中，我们将进行实验来研究这种影响。 实验在 FEWNERD 数据集的 8-shot 设置下进行，我们使用 3 个不同的硬编码模板和 4 个软编码模板（通过改变提示词的数量 m）。 结果表明，模板的选择对基于提示的小样本学习的性能产生了相当大的影响。 对于硬编码模板，描述“在这句话中”位置的短语有助于显着提高性能。 对于软编码模板，令人惊讶的是，提示学习模型以最少的特殊标记产生了最好的结果。

6 Related Work

After a series of effective PLMs like GPT (Radford et al., 2018), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and T5 (Raffel et al.,

经过一系列有效的 PLM，如 GPT (Radford et al., 2018)、BERT (Devlin et al., 2019)、RoBERTa (Liu et al., 2019) 和 T5 (Raffel et al.,

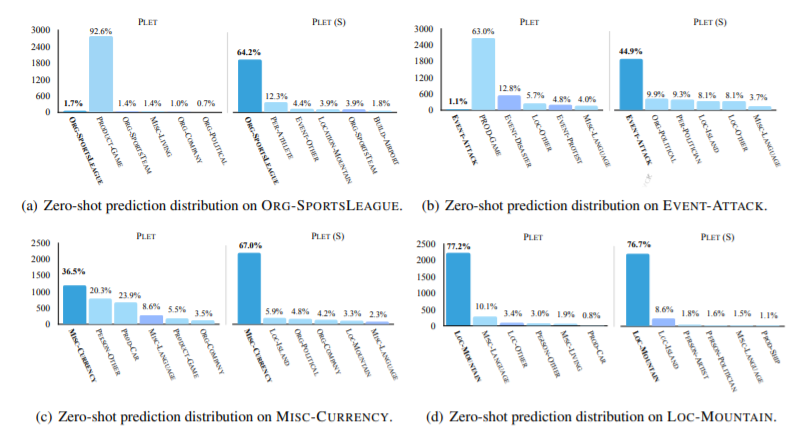


Figure 4: Zero-shot prediction distribution on four types in FEW-NERD, in each subgraph, the left part illustrates the results of PLET and the right part shows the results of PLET (S). denotes the correct predictions, denotes the wrong predictions with correct coarse-grained types, and denotes the wrong predictions with wrong coarsegrained types.

图 4：FEW-NERD 中四种类型的零样本预测分布，在每个子图中，左侧显示了 PLET 的结果，右侧显示了 PLET (S) 的结果。 表示正确的预测，用正确的粗粒度类型表示错误的预测，用错误的粗粒度类型表示错误的预测。

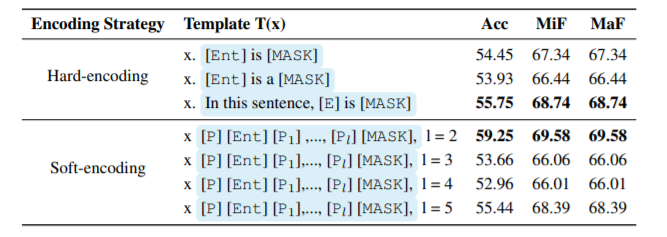


Table 5: Effect of templates. The results are produced under 8-shot setting on FEW-NERD dataset by PLET.

表 5：模板的效果。 结果是在 PLET 的 FEW-NERD 数据集上的 8-shot 设置下产生的。

fine-tuned PLMs have demonstrated their effectiveness on various important NLP tasks, such as dialogue generation (Zhang et al., 2020), text summarization (Zhang et al., 2019; Liu and Lapata, 2019), question answering (Adiwardana et al., 2020), and text classification (Baldini Soares et al., 2019; Peng et al., 2020; Ding et al., 2021a).

微调的 PLM 已经证明了它们在各种重要的 NLP 任务上的有效性，例如对话生成（Zhang 等人，2020）、文本摘要（Zhang 等人，2019 年；Liu 和 Lapata，2019 年）、问答（Adiwardana 等人，2019 年） ., 2020) 和文本分类 (Baldini Soares et al., 2019; Peng et al., 2020; Ding et al., 2021a)。

Despite the success of fine-tuning PLMs, the huge objective form gap between pre-training and fine-tuning still hinders the full use of per-trained knowledge for downstream tasks (Liu et al., 2021b; Han et al., 2021b; Hu et al., 2021). To this end, prompt-learning has been proposed. In promptlearning, by leveraging language prompts as contexts, downstream tasks can be expressed as some cloze-style objectives similar to those pre-training objectives. The seminal work that stimulates the development of prompt-learning is the birth of GPT3 (Brown et al., 2020), which uses hand-crafted prompts for tuning and achieves very impressive performance on various tasks, especially under the setting of few-shot learning.

尽管微调 PLM 取得了成功，但预训练和微调之间巨大的目标形式差距仍然阻碍了将经过训练的知识充分用于下游任务（Liu 等人，2021b；Han 等人，2021b； 胡等人，2021）。 为此，已经提出了即时学习。 在提示学习中，通过利用语言提示作为上下文，下游任务可以表示为一些类似于预训练目标的完形填空式目标。 刺激提示学习发展的开创性工作是 GPT3 (Brown et al., 2020) 的诞生，它使用手工制作的提示进行调优，在各种任务上取得了非常出色的表现，尤其是在少拍的设置下 学习。

Inspired by GPT-3, a series of hand-crafted prompts have been widely explored in knowledge probing (Trinh and Le, 2018; Petroni et al., 2019; Davison et al., 2019), relation classification (Han et al., 2021b), entiment classification and natural language inference (Schick and Schütze, 2021; Liu et al., 2021b). To avoid labor-intensive prompt design, automatic prompt search has also been extensively explored Schick et al. (2020); Schick and Schütze (2021); Shin et al. (2020); Gao et al. (2020); Liu et al. (2021a) to generate language phrases for prompts. Recently, some continuous prompts have also been proposed (Li and Liang, 2021; Lester et al., 2021), which directly use a series of learnable continuous embeddings as prompts rather than discrete language phrases.

受 GPT-3 的启发，一系列手工制作的提示在知识探索（Trinh 和 Le，2018；Petroni 等，2019；Davison 等，2019）、关系分类（Han 等， 2021b）、情感分类和自然语言推理（Schick 和 Schütze，2021；Liu 等，2021b）。 为了避免劳动密集型提示设计，自动提示搜索也得到了广泛的探索 Schick 等人。 (2020); Schick 和 Schütze (2021)； 申等人。 (2020); 高等人。 (2020); 刘等人。 (2021a) 为提示生成语言短语。 最近，还提出了一些连续提示（Li and Liang, 2021; Lester et al., 2021），它们直接使用一系列可学习的连续嵌入作为提示而不是离散的语言短语。

In this paper, we aim to stimulate PLMs with prompt-learning to capture the attribute information of entities. We take fine-grained entity typing, a crucial task in knowledge extraction to assign entity types to entity mentions (Lin et al., 2012), as the foothold to develop prompt-learning strategies. In fact, Dai et al. (2021) use hypernym extraction patterns to enhance the context and apply masked language modeling to tackle the ultra-fine entity typing problem (Choi et al., 2018) with free-form labels, which shares a similar idea with promptlearning. In our work, we mainly emphasize using prompt-learning to extract entity types that have been pre-defined in low-data scenarios.

在本文中，我们旨在通过即时学习来刺激 PLM，以捕获实体的属性信息。 我们采用细粒度实体类型，这是知识提取中将实体类型分配给实体提及的一项关键任务（Lin 等，2012），作为制定快速学习策略的立足点。 事实上，戴等人。 (2021) 使用上位词提取模式来增强上下文并应用掩码语言建模来解决具有自由形式标签的超精细实体类型问题（Choi 等人，2018），这与提示学习有类似的想法。 在我们的工作中，我们主要强调使用提示学习来提取低数据场景中预先定义的实体类型。

7 Conclusion

This work investigates the application of promptlearning on fine-grained entity typing. More specifically, we proposes a framework PLET that could deal with fine-grained entity typing in fully supervised, few-shot and zero-shot scenarios. In PLET, we first introduce a simple and effective promptlearning pipeline that could be used to extract entity types with both sufficient and insufficient supervision. Furthermore, to handle the zero-shot setting, we propose a self-supervised prompt-learning approach that automatically learns and summarizes entity types based on unlabeled corpora and a predefined label schema. PLET utilizes prompts to take advantage of prior knowledge distributed in PLMs, and could learn pre-defined type information without overfitting by performing distributionlevel optimization. In our future work, along the direction of PLET (S), we will explore better promptlearning approaches to automatically learning entity types from unlabeled data.

这项工作研究了提示学习在细粒度实体类型中的应用。 更具体地说，我们提出了一个框架 PLET，可以在完全监督、少样本和零样本场景中处理细粒度实体类型。 在 PLET 中，我们首先引入了一个简单有效的提示学习管道，可用于提取具有充分监督和不充分监督的实体类型。 此外，为了处理零样本设置，我们提出了一种自监督的提示学习方法，该方法基于未标记的语料库和预定义的标签模式自动学习和总结实体类型。 PLET 利用提示来利用分布在 PLM 中的先验知识，并且可以通过执行分布级别优化来学习预定义的类型信息而不会过度拟合。 在我们未来的工作中，沿着 PLET (S) 的方向，我们将探索更好的即时学习方法，以从未标记的数据中自动学习实体类型。