Hindawi
International Journal of Intelligent Systems
Volume 2024, Article ID 1941221, 14 pages
https://doi.org/10.1155/2024/1941221

Research Article

Leveraging Pretrained Language Models for Enhanced Entity Matching: A Comprehensive Study of Fine-Tuning and Prompt Learning Paradigms

Yu Wang[1], **Luyao Zhou**[1], **Yuan Wang**[2,3] and **Zhenwan Peng**[1]

[1]*School of Biomedical Engineering, Anhui Medical University, Hefei 230001, China*
[2]*Institute of Advanced Technology, University of Science and Technology of China, Hefei 230001, China*
[3]*Anhui HYJK Medical Technology Co., Ltd, Hefei 230001, China*

Correspondence should be addressed to Yu Wang; briskyu@mail.ustc.edu.cn

Received 5 October 2023; Revised 17 March 2024; Accepted 27 March 2024; Published 15 April 2024

Academic Editor: Surya Prakash

Pretrained Language Models (PLMs) acquire rich prior semantic knowledge during the pretraining phase and utilize it to enhance downstream Natural Language Processing (NLP) tasks. Entity Matching (EM), a fundamental NLP task, aims to determine whether two entity records from different knowledge bases refer to the same real-world entity. This study, for the first time, explores the potential of using a PLM to boost the EM task through two transfer learning techniques, namely, fine-tuning and prompt learning. Our work also represents the first application of the soft prompt in an EM task. Experimental results across eleven EM datasets show that the soft prompt consistently outperforms other methods in terms of $F1$ scores across all datasets. Additionally, this study also investigates the capability of prompt learning in few-shot learning and observes that the hard prompt achieves the highest $F1$ scores in both zero-shot and one-shot context. These findings underscore the effectiveness of prompt learning paradigms in tackling challenging EM tasks.

1. Introduction

In the era of big data, extensive Knowledge Bases (KBs) or Knowledge Graphs (KGs) have been constructed, serving as structured repositories of knowledge about the world [1, 2]. However, the entities coming from different KBs are often heterogeneous and presented using different attributes. Figure 1 illustrates the disparities in attribute values for a same product in two different online shopping KBs. When integrating KBs to build recommendation systems or question-answering systems [3–5], these disparities can lead to increased redundancy and reduced performance in downstream tasks. Entity Matching (EM), as a fundamental knowledge extraction task in Natural Language Processing (NLP), aims to determine whether two entity records from different KBs refer to the same real-world entity, thereby addressing the aforementioned challenge [6].

Early EM methods are based on editing distance, which is convenient but less practical. Machine learning-based approaches transform EM into a binary classification problem using classifiers like Support Vector Machine (SVM) [7]. However, given that these methods require manual feature engineering, their generalization is limited. With the rise of deep learning, researchers also attempt to tackle the matching problem leveraging techniques like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) [8, 9]. However, these deep learning-based approaches could only capture semantic knowledge implicit in the training set, and obtaining labelled training data is challenging.

In light of the aforementioned drawbacks associated with deep learning, researchers have proposed Pretrained Language Models (PLMs) consisting of multiple layers of Transformer blocks [10], such as BERT [11] and ERNIE [12].

Initially, these models acquire prior semantic knowledge from extensive unlabelled text corpora through pretraining tasks like Masked Language Model (MLM) [11] and Next Sentence Prediction (NSP) [11]. Subsequently, this semantic knowledge can be employed to enhance a variety of downstream NLP tasks [13–15]. This can be regarded as a form of transfer learning, and there are currently two popular paradigms: fine-tuning and prompt learning [16].

The fine-tuning paradigm involves modifying the model structure for downstream tasks, such as adding an additional classifier on top of the PLM's encoder and discarding the decoder part [17]. Therefore, fine-tuning will introduce discrepancies of the training goal between downstream and pretraining tasks, making it challenging for the model to fully leverage the semantic knowledge acquired during pretraining [18]. In contrast, prompt learning reformulates downstream tasks based on the pretraining task of a PLM, utilizing all the parameters in a PLM, including both the encoder and decoder, rather than only using the encoder. Taking the most representative BERT-series PLMs as an example, when employing these PLMs for prompt learning, they bridge the gap between downstream tasks and the pretraining task by wrapping the raw input with prompt templates containing [MASK] tokens, stimulating better PLM semantic understanding capability by reproducing the MLM process. At this point, downstream tasks are transformed into predictions for these placeholders, resulting in performance enhancement [19–21]. For example, Jin et al. [17] proposed a "Word Transferred LM" for sentiment analysis, transferring the target words of a sentence into pivot tokens via MLM. Zhao et al. [22] developed a series of prompt learning approaches, called PromptMR, investigating how prompt learning could improve metonymy resolution. The methods they proposed achieved competitive accuracy compared to baseline models.

Evidently, the key to applying prompt learning lies in the textual prompt templates. Depending on how templates are generated, prompt learning can be categorized as hard prompt (or discrete prompt) and soft prompt (or continuous prompt) [18]. For the former one, templates consist of fixed tokens [23], while for the latter one, templates are vectors that can be learned in a continuous space [24]. Therefore, although the templates generated by the soft prompt may not be understandable as natural language by humans, they have the capability to discover more suitable template embeddings. However, there is no published work on comparing the differences between the fine-tuning and prompt learning paradigms in EM tasks comprehensively, and the properties of prompt learning in EM remain unexplored.

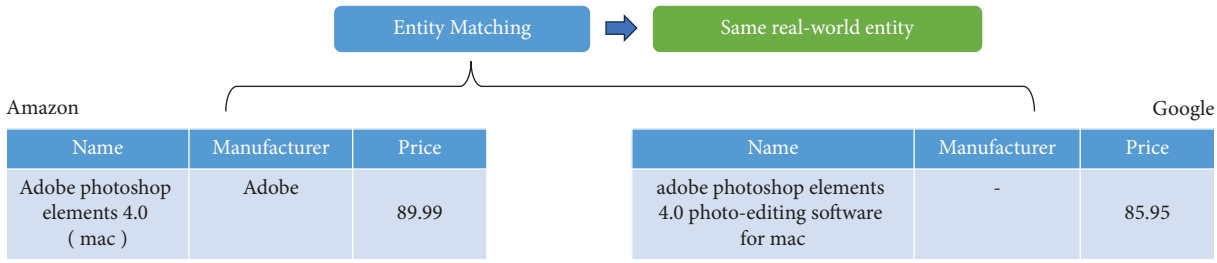
The present research provides, for the first time, comprehensive comparison between fine-tuning and prompt learning paradigms for the EM task and explores the capabilities of prompt learning in the context of few-shot learning. This is also the first study on how to apply soft prompts to EM tasks. The main contributions of this study can be summarized as follows:

(1) We conduct a comprehensive comparison of fine-tuning and prompt learning paradigms when applied to EM. Specifically, we transform the structured attribute values of two entity records into textual descriptions. Given that the BERT-series PLMs are widely used and more representative, we chose ERNIE-2.0-en, which shares the same architecture and pretraining tasks as BERT, as the backbone model. For fine-tuning, we train a binary classifier using the representation of [CLS] to determine whether the two entity records are "consistent." For hard prompt, we utilize the template consisting of fixed tokens to convert the original input into sequences with [MASK] tokens and predict these placeholders. Through this way, the downstream EM task is transformed into the pretraining task MLM. The approach for soft prompt is similar to that of the hard one, but the template consists of pseudo tokens and searches for their embeddings in a continuous space using a Multilayer Perceptron (MLP). Additionally, this is the first exploration of how to apply the soft prompt to an EM task.

(2) We perform a comparative analysis of $F1$ scores between fine-tuning and prompt learning on eleven datasets. Notably, our findings reveal that soft prompts consistently exhibit superior performance across all datasets. Nevertheless, it is worth noting that, in datasets with a substantial number of training samples, prompt learning demonstrates comparatively modest improvements in $F1$ scores. Additionally, we also tracked the loss values throughout the training process, and our observations indicate that hard prompts consistently yield the lowest loss values in the initial training phases, suggesting their potential for few-shot learning.

(3) Consequently, we also undertake experiments to validate the capability of prompt learning for few-shot learning. We observe the performance of both fine-tuning and prompt learning in the contexts of zero-shot learning and one-shot learning, utilizing structured iTunes-Amazon and DBLP-Scholar datasets. The outcomes indicate that prompt learning consistently outperforms fine-tuning, with hard prompt learning displaying the best performance in the context of few-shot learning. The raw datasets used to support the findings of this study are available at https://github.com/anhaidgroup/deepmatcher. We also have made the source code publicly available on GitHub: https://github.com/Briskyu/entity_matching.

The overall structure of this study takes the form of six sections. A brief review of the related work is presented in Section 2. Section 3 deals with the methodology used in this study. The experimental results are presented in Section 4, while the discussion is provided in Section 5. Finally, Section 6 concludes this study with a summary.



Name	Manufacturer	Price
Adobe photoshop elements 4.0 (mac)	Adobe	89.99

Name	Manufacturer	Price
adobe photoshop elements 4.0 photo-editing software for mac	-	85.95

FIGURE 1: The illustration of entity matching. The tables display the product (entity) records of "Adobe Photoshop" in Amazon and Google. The EM task involves determining whether these two entity records represent the same real-world entity.

2. Related Works

Early research into EM primarily utilized methods based on edit distance or machine learning, such as [25–27]. However, these methods either proved to be impractical or exhibited poorer generalization. Therefore, the majority of current EM research is based on deep learning or pretrained language models.

2.1. Deep Learning. Deep learning has achieved remarkable results in the field of EM, driven by the development of computer hardware, especially the Graphics Processing Unit (GPU) [8]. For example, Di Cicco et al. [28] introduced a methodology to produce explainable deep learning models for the EM task. Nie et al. [29] proposed a deep sequence-to-sequence entity matching model, denoted as Seq2Seq-Matcher, which can effectively solve the heterogeneous problems by modelling ER as a token-level sequence-to-sequence matching task. Koolin et al. [30] proposed an EM approach, which is mainly based on a record linkage process and detects records that refer to the same entity. Gottapu et al. [31] used a single-layer convolutional neural network to perform an EM task. Kasai et al. [32] attempted to explore the performance of deep learning in the low-resource RM task. They designed an architecture that can learn a transferable model from a high-resource setting to a low-resource one. These methods based on deep learning can learn features from training data, eliminating the need for manual feature engineering. However, the semantic knowledge they acquire is limited to the training set, which constrains the performance of deep learning-based EM models, especially considering that obtaining labelled training data is challenging.

2.2. Pretrained Language Models. PLMs consisting of multiple transformer blocks, such as BERT [11] and ERNIE [12], can acquire prior semantic knowledge from large-scale unlabelled corpora through pretraining phase and apply this knowledge to downstream tasks. Consequently, PLM-based approaches outperform deep learning-based methods in various NLP tasks. There has been research focusing on the application of PLMs to EM. For example, Chen et al. [33] proposed a transfer-learning EM approach, leveraging a knowledge base constructed through PLMs. Mehdi et al. [34] investigated whether PLM-based EM models can be trusted in real-world applications where data distribution is different from that of training. Due to the existing differences in training goal between fine-tuning and pretraining, recent efforts have focused on employing prompt learning to bridge the gap between pretraining and downstream tasks, namely, utilizing all the parameters both in the encoder and decoder of a PLM. The key to conducting prompt learning lies in reformulating the downstream target task based on the textual prompts. There are two types of textual prompts: cloze prompts, which fill in the blanks of a textual string, and prefix prompts, which continue a string prefix [18]. In addition, prompt learning can generally be categorized into two types: the hard prompt [23, 35] and soft prompt [24, 36]. The difference lies in the fact that the hard prompt has fixed templates, whereas the soft prompt allows the template to be learned in a continuous space. According to the literature review, there has been no comprehensive analysis of fine-tuning and prompt learning specifically for EM.

3. Methods

This section first provides a detailed introduction to the problem definition of the EM task, followed by a thorough presentation of the specific model structures for the two paradigms: fine-tuning and prompt learning.

3.1. Problem Definition. The EM task aims to determine whether two entity mentions or records refer to one real-world entity. Specifically, given a dataset $D = \{(E_1, E_2), A, Y\}$, where E_1 and E_2 are sets of entity mentions, A denotes the set of attributes, and Y denotes the set of true labels. For any $e_i \in E_1$ and $e_j \in E_2$, both are composed of n attributes, i.e., $e_i = \{a_{i1}, a_{i2}, a_{i3} \ldots a_{in}\}$ and $e_j = \{a_{j1}, a_{j2}, a_{j3} \ldots a_{jn}\}$, where $(a_{i1}, a_{i2}, a_{i3} \ldots a_{in}, a_{j1}, a_{j2}, a_{j3} \ldots a_{jn}) \in A$. Assuming the relation between two entity mentions is represented by $y_k \in Y$, a mapping function $f(e_i, e_j) \longrightarrow y_k$ is calculated through D. For another dataset $D' = \{(E_1', E_2'), A', Y'\}$ with the same distribution as D, there exist $e_i' \in E_1'$ and $e_j' \in E_2'$. $y' = f(e_i', e_j')$ should be the same as the true label $y_k' \in Y'$. The goal of this study is to construct an appropriate model to represent the mapping function $f(\cdot)$.

3.2. Fine-Tuning. For the fine-tuning paradigm, we follow the method outlined in Figure 2. First, structured key-value pairs are transformed into unstructured textual data denoted as $X = \{x_1, x_2, \ldots, x_n\}$, where n denotes the length of X.

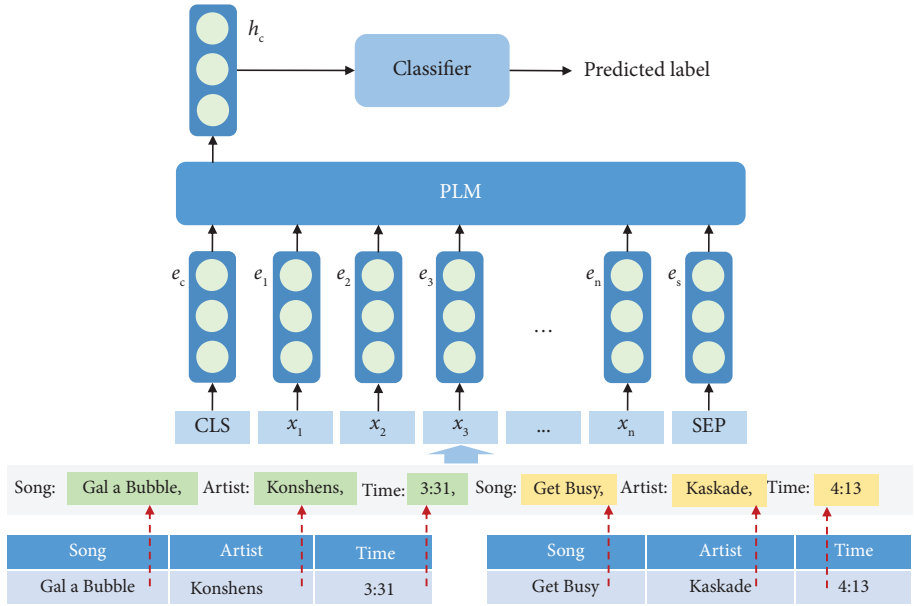


Figure 2: The model architecture of fine-tuning.

Then, the embeddings denoted as $E = \{e_{\text{cls}}, e_1, e_2, \ldots, e_n, e_{\text{sep}}\}$ are acquired by incorporating the [CLS] and [SEP] tokens surrounding X and executing an embedding table lookup. The PLM generates the representations, which are denoted as $H = \{h_{\text{cls}}, h_1, h_2, \ldots, h_n, h_{\text{sep}}\}$, for each token within the input sequence. In this context, the EM task can be regarded as a binary classification task, and the objective is to ascertain whether two entities are identical or dissimilar. Finally, the representation corresponding to [CLS] is used to calculate the predicted label y_p through the following equation:

$$y_p = \arg\max(\text{soft}\max(W_c * h_c + b_c)), \tag{1}$$

where $W_c \in R^{M*H}$ and $b_c \in R^M$ are the learnable weight matrix and bias and initialized with random values. M is the number of labels ($M = 2$ in this study), and H is the dimension of the hidden layer.

3.3. Prompt Learning. In contrast to fine-tuning, prompt learning transforms the downstream task into the form of the pretraining task, aligning the objectives of pretraining and downstream tasks. In this study, the downstream task is transformed into the MLM task since we select ERNIE-2.0-en, which is a BERT-like PLM, as the backbone model. According to the construction method of prompt templates, it can be categorized as either the hard prompt or the soft prompt.

3.3.1. Hard Prompt. For the hard prompt, structured key-value pairs are first transformed into two unstructured text chunks, as shown in Figure 3(a). Then, the new input sequence $X = \{x_1, x_2, \ldots, x_n\}$ is constructed based on the template "<sentence_1> and <sentence_2> they are [MASK]." It is obvious that x_n corresponds to the [MASK] token. Subsequently, $E = \{e_{\text{cls}}, e_1, e_2, \ldots, e_n, e_{\text{sep}}\}$ is obtained as the same method as fine-tuning and input into the PLM. Finally, $h_m \in H = \{h_{\text{cls}}, h_1, h_2, \ldots, h_n, h_{\text{sep}}\}$ generated by the PLM is passed into the MLM head. The most likely word w, which can represent the predicted label y_p, is selected from a dictionary through the following equation:

$$w = \arg\max(\text{soft}\max(W_m * h_m + b_m)), \tag{2}$$

where $W_m \in R^{K*H}$ and $b_c \in R^K$ are the learnable weight matrix and bias in the MLM head, but initialized with the values learned by pretraining process. K is the size of the word dictionary, and H is the dimension of the hidden layer.

3.3.2. Soft Prompt. For the soft prompt, structured key-value pairs are also transformed into two text chunks, as illustrated in Figure 3(b). Then, a new input sequence $X = \{x_1, x_2, \ldots, x_{p1}, x_{p2}, x_n\}$ is constructed based on the template "<sentence_1> and <sentence_2> pseudo pseudo [MASK]," and x_n corresponds to the [MASK] token. It is worth noting that the template contains pseudo tokens, which can be represented using [UNK], and the number of pseudo tokens is a hyperparameter (set to 10 in this study). $E = \{e_{\text{cls}}, e_1, e_2, \ldots, e_{p1}, e_{p2}, e_n, e_{\text{sep}}\}$ is still acquired through an embedding lookup operation. The final input sequence to the PLM is $\{e_{\text{cls}}, e_1, e_2, \ldots, r_{p1}, r_{p2}, e_n, e_{\text{sep}}\}$, where r_{p1} and r_{p2} are obtained using the following equations:

$$r_{p1} = W_p * e_{p1} + b_p, \tag{3}$$

$$r_{p2} = W_p * e_{p2} + b_p, \tag{4}$$

where $W_p \in R^{L*L}$ and $b_p \in R^L$ are the learnable parameters of a Multilayer Perceptron (MLP) layer and L is the embedding dimension. Finally, h_m generated by the PLM is passed into the MLM head, and w is selected from a dictionary through the following equation:

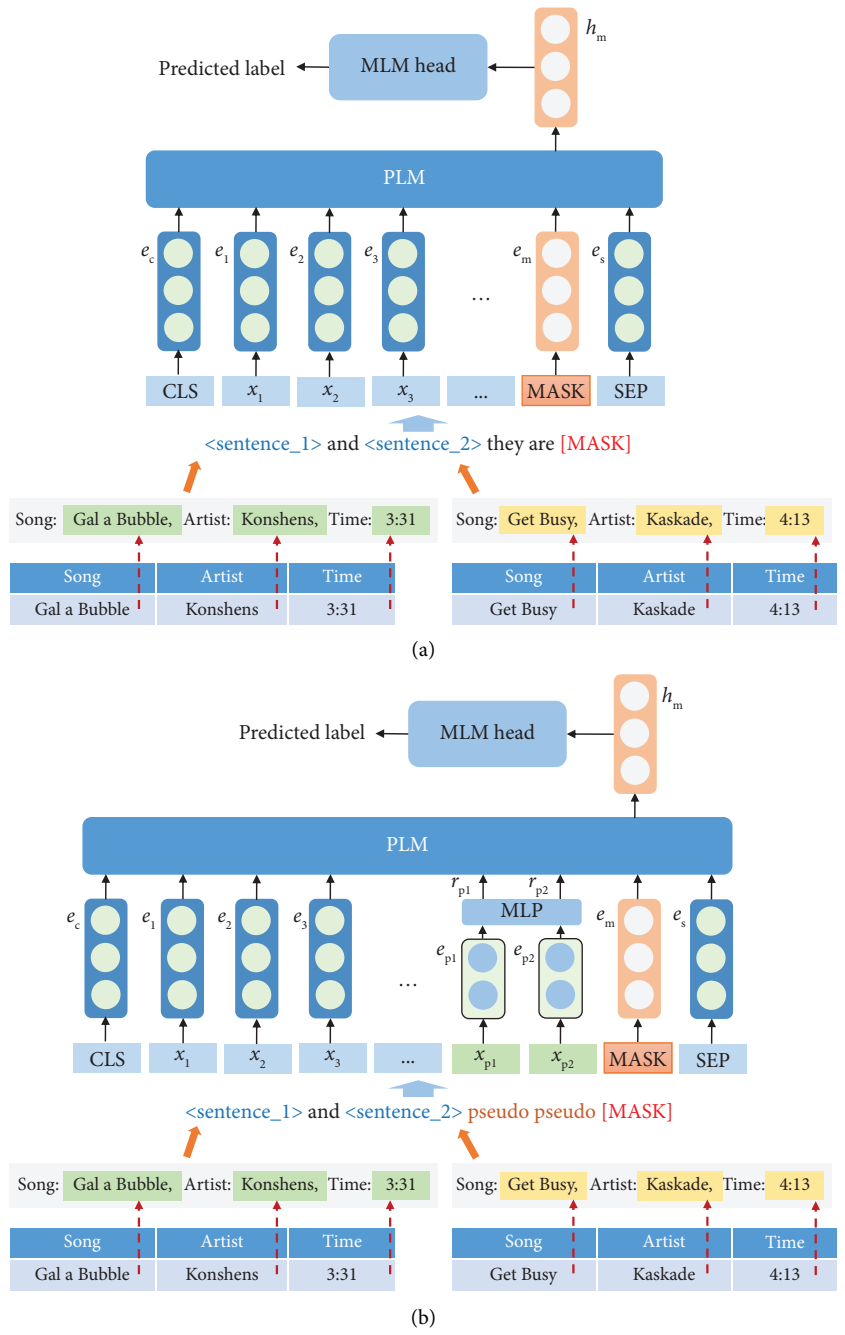


FIGURE 3: The model architecture of hard prompt and soft prompt. (a) The entity matching task with a hard prompt. (b) The entity matching task with a soft prompt.

$$w = \arg\max\left(\text{soft}\max\left(W_m * h_m + b_m\right)\right), \tag{5}$$

where $W_m \in R^{K*H}$ and $b_c \in R^K$ are the learnable weight matrix and bias, but initialized with the values learned by pretraining process, too. K is the size of the word dictionary, and H is the dimension of the hidden layer. Through the aforementioned approach, the soft prompt can find more appropriate template embeddings in continuous space.

4. Experiments

This section aims to evaluate the method proposed in Section 3 through experiments and presents the selected datasets, evaluation metrics, hyperparameters, and experimental results. For the software environment of the experiments, we utilized the Paddlepaddle deep learning framework, introduced by Baidu (https://github.com/paddlepaddle/paddle). As for the hardware environment, we employed a 2-core CPU, a RAM with 16 GB, and an NVIDIA V100 GPU with 16 GB of memory for the experiments. Additionally, given that this study marks the first application of the soft prompt to EM, we refer to the EM approach based on soft prompt as "our model."

4.1. Datasets. We evaluated the proposed entity matching method in this study using the datasets provided by Mudgal et al. [37]. These datasets differ in terms of type, domain, and size, allowing us to assess the generalizability of the entity matching model. Table 1 presents an overview of the datasets, indicating that the datasets consist of two types: structured and dirty. The dirty datasets are obtained by modifying the structured dataset and are differentiated using indices 1 and 2. Specifically, for each attribute except "title," there is a 50% chance that it will be randomly moved to the "title" attribute. This simulates a common kind of dirty data seen in the real-life scenarios while keeping the modifications simple. The "Size" column represents the total number of labelled samples for each dataset. We split all the dataset into three parts with ratio of 3 : 1 : 1, for training, validation, and evaluation, respectively. "Positive instances" represent the number of positive samples in the dataset, indicating two entities that are the same in the real world. "Attributes" indicate the number of attributes corresponding to each entity in the dataset. The more detailed descriptions of each dataset are provided below:

(1) BeerAdvo-RateBeer: This dataset contains beer data from BeerAdvocate and RateBeer. The attributes include beer name, brewery name, beer type, and alcohol by volume.

(2) iTunes-Amazon: This dataset contains music data from iTunes and Amazon. The attributes include song name, artist name, album name, genre, price, copyright information, and release date.

(3) DBLP-ACM: This dataset contains bibliographic data from DBLP and ACM. The attributes include title, author, venue, and year.

(4) DBLP-Scholar: This dataset contains bibliographic data from DBLP and Google Scholar. The attributes include title, author, venue, and year.

(5) Amazon-Google: This dataset contains product data from Amazon and Google. The attributes include title, manufacturer, and price.

(6) Walmart-Amazon: This dataset contains product data from Walmart and Amazon. The attributes include title, category, brand, model number, and price.

(7) Abt-Buy: This dataset contains product data from Abt.com and Buy.com. The attributes include name, description, and price.

4.2. Hyperparameters. The seven hyperparameters involved in training the model are presented in Table 2. It should be noted that for BeerAdvo-RateBeer and iTunes-Amazon datasets, the epoch and batch size are set to 8 and 10, respectively, while for the other datasets, these two hyperparameters are set to 16 and 5. For all datasets, we use AdamW as the optimizer with an initial learning rate of $1e-5$, a maximum gradient norm of 1.0, and a maximum input length of 512.

4.3. Evaluation Metrics. The evaluation metric used in the experiment is "$F1$," calculated according to the following formulation, where precision represents the ratio of correctly predicted number among the predicted positive samples, and recall represents the ratio of correctly predicted positive samples in the evaluation set.

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{6}$$

4.4. Experiment Results

4.4.1. F1 Scores of Different Models. As shown in Tables 3 and 4, we first compare the $F1$ scores of our model on both the structured and dirty datasets with four previously popular entity matching models: DeepER, DeepMatcher, Magellan, and Multicontext Attention (MCA). The DeepER proposed by Ebraheem et al. [38] utilizes GloVe [39] to obtain word embeddings. These embeddings are used to generate the representations of tuples through Long Short-Term Memory (LSTM) and then employ cosine similarity to determine whether they represent the same entity. The DeepMatcher proposed by Mudgal et al. [37] uses a bi-directional RNN with decomposable attention to implement attribute summarization. Magellan proposed by Konda et al. [40] is an EM system that provides a step-by-step guide, instructing users on how to operate in each EM scenario. Zhang et al. [41] proposed an integrated multicontext attention framework that takes into account self-attention, pair-attention, and global-attention from three types of contexts. Therefore, this model is referred to as MCA. In summary, the four methods mentioned above are either based on deep learning models such as RNN or LSTM, or they incorporate attention mechanisms. Moreover, they do not use PLMs to generate the representations that contain prior semantic knowledge. In contrast, our model uses the ERNIE-2.0-base-en to generate representations and employs the soft prompt method introduced in Section 3.3.2 to train the EM model. The column "$\Delta F1$" is set to indicate the improvement in $F1$ scores achieved by our model compared to the previous best result. It is evident that our proposed EM method outperforms the others, both on the structured and dirty datasets.

Table 1: Overview of the eleven EM datasets.

Type	Name	Domain	Size	Positive instances	Attributes
Structured	BeerAdvo-RateBeer	Beer	450	68	4
	iTunes-Amazon$_1$	Music	539	132	8
	DBLP-ACM$_1$	Citation	12,363	2,220	4
	DBLP-Scholar$_1$	Citation	28,707	5,347	4
	Amazon-Google	Software	11,460	1,167	3
	Walmart-Amazon$_1$	Electronics	10,242	962	5
	Abt-Buy	Product	9,575	1,028	3
Dirty	iTunes-Amazon$_2$	Music	539	132	8
	DBLP-ACM$_2$	Citation	12,363	2,220	4
	DBLP-Scholar$_2$	Citation	28,707	5,347	4
	Walmart-Amazon$_2$	Electronics	10,242	962	5

Table 2: Overview of the hyperparameters.

Name	Batch	Epoch	Learning rate	Optimizer	Max gradient norm	Max length
BeerAdvo-RateBeer	8	10	$1e-5$	AdamW	1.0	512
iTunes-Amazon	8	10	$1e-5$	AdamW	1.0	512
DBLP-ACM	16	5	$1e-5$	AdamW	1.0	512
DBLP-Scholar	16	5	$1e-5$	AdamW	1.0	512
Amazon-Google	16	5	$1e-5$	AdamW	1.0	512
Walmart-Amazon	16	5	$1e-5$	AdamW	1.0	512
Abt-Buy	16	5	$1e-5$	AdamW	1.0	512

Table 3: $F1$ scores of different EM models on structured datasets.

Datasets	DeepER	Magellan	DeepMatcher	MCA	Our model	$\Delta F1$
BeerAdvo-RateBeer	72.7	78.8	72.7	80.0	86.7	+6.7
iTunes-Amazon$_1$	—	91.2	88.0	—	92.9	+1.7
DBLP-ACM$_1$	97.6	98.4	98.4	98.9	99.2	+0.3
DBLP-Scholar$_1$	92.3	92.3	94.7	95.2	95.7	+0.4
Amazon-Google	62.1	49.1	69.3	71.4	76.2	+4.8
Walmart-Amazon$_1$	39.0	71.9	66.9	74.7	84.1	+9.4
Abt-Buy	36.1	43.6	62.8	69.3	80.0	+10.7

Table 4: $F1$ scores of different EM models on dirty datasets.

Datasets	DeepER	Magellan	DeepMatcher	MCA	Our model	$\Delta F1$
iTunes-Amazon$_2$	—	46.8	79.4	—	92.6	+13.4
DBLP-ACM$_2$	94.9	91.9	98.1	98.5	99.4	+0.9
DBLP-Scholar$_2$	92.3	92.3	94.7	95.2	95.3	+0.1
Walmart-Amazon$_2$	33.6	37.4	59.2	74.7	84.6	+9.9

4.4.2. F1 Scores of Different Paradigms. We also conduct a comparative analysis between fine-tuning and prompt learning paradigms on both structured and dirty datasets. The corresponding results are presented in Tables 5 and 6, with the abbreviations "FT," "HP," and "SP" denoting "fine-tuning," "hard prompt," and "soft prompt," respectively. "$\Delta F1$" quantifies the enhancement in $F1$ scores. Based on the experimental findings, it is evident that the two prompt learning approaches consistently outperform the fine-tuning paradigm across the majority of datasets, with the exception of the structured iTunes-Amazon and DBLP-Scholar datasets. For these particular datasets, the fine-tuning and the adoption of a hard prompt yield nearly identical $F1$ scores. Notably, the utilization of a soft prompt consistently exhibited superior performance, manifesting an enhancement in $F1$ scores across all datasets.

It is noteworthy that our observations indicate a potential correlation between the magnitude of $F1$ scores' improvement achieved through prompt learning and the scale of the training dataset. Specifically, prompt learning demonstrates a propensity for generating higher enhancements in $F1$ scores for smaller datasets. To illustrate, consider the structured datasets BeerAdvo-RateBeer, DBLP-ACM, and DBLP-Scholar, all having the number of attributes with four. However, the BeerAdvo-RateBeer dataset comprises a modest size of 450 instances, significantly

Table 5: $F1$ scores of different paradigms on structured datasets.

Datasets	FT	HP/Δ$F1$	SP/Δ$F1$
BeerAdvo-RateBeer	80.0	83.9/+3.9	86.7/+2.8
iTunes-Amazon$_1$	91.5	91.5/0.0	92.9/+1.4
DBLP-ACM$_1$	98.6	98.8/+0.2	99.2/+0.4
DBLP-Scholar$_1$	95.7	95.6/−0.1	95.7/0.0
Amazon-Google	73.8	74.1/+0.3	76.2/+2.1
Walmart-Amazon$_1$	79.9	83.5/+3.6	84.1/+0.6
Abt-Buy	74.1	79.0/+4.9	80.0/+1.0

Table 6: $F1$ scores of different paradigms on dirty datasets.

Datasets	FT	HP/Δ$F1$	SP/Δ$F1$
iTunes-Amazon$_2$	90.9	91.5/+0.6	92.6/+1.1
DBLP-ACM$_2$	98.5	99.1/+0.6	99.4/+0.3
DBLP-Scholar$_2$	94.9	95.0/+0.1	95.3/+0.3
Walmart-Amazon$_2$	81.8	83.2/+1.4	84.6/+1.4

smaller compared to the more extensive DBLP-ACM (12,363 instances) and DBLP-Scholar (28,707 instances) datasets. Supporting this observation, Table 5 exhibits that, for the structured BeerAdvo-RateBeer dataset, the deployment of hard prompt leads to a notable increase of 3.6 percentage points in the $F1$ scores, whereas for the structured DBLP-ACM and DBLP-Scholar datasets, the $F1$ scores attained by hard prompt are almost equivalent to those achieved through fine-tuning.

4.4.3. Loss Values of Different Paradigms. Considering Section 4.4.2 shows the comparable performance between prompt learning and fine-tuning on the structured iTunes-Amazon and DBLP-Scholar datasets, we recorded the loss values of the EM models employing these paradigms at each training epoch, with the intention of conducting a detailed investigation into their performance in the EM task. The outcomes are presented in Tables 7 and 8. Furthermore, Figures 4 and 5 provide a visual depiction of the descending trend of the loss values. Notably, during the initial phases of training on the iTunes-Amazon dataset, hard prompt demonstrated the most favourable performance. As the training progressed, fine-tuning manifested a rapid reduction in loss. However, upon reaching complete convergence, its loss value is higher than that of two prompt learning methods. Our model, which is based on the soft prompt paradigm, ultimately achieved the most remarkable outcome, with the lowest loss value of $6.27e-5$. In order to provide a clearer representation of the descending trends of loss values for different methods at the end of training, we took the logarithm of the loss values using a base of 10, as shown in the lower part of Figure 4. For the DBLP-Scholar dataset, the observations in the first epoch are consistent with those of the iTunes-Amazon dataset. Nevertheless, as the training advanced, all three methods converged to nearly identical loss values.

4.4.4. F1 Scores of Few-Shot Learning under Different Paradigms. The aforementioned experiment underscores that the prompt learning exhibits lower loss values in the initial phases of training when compared to the fine-tuning-based approach. This can be attributed to the narrowing of the gap between downstream and pretraining tasks. To further substantiate this finding, we systematically investigated the performance of few-shot learning under different paradigms. Specifically, we conducted zero-shot and one-shot learning using the test sets of the structured iTunes-Amazon and DBLP-Scholar datasets as the query sets. Within the context of zero-shot learning, we appraised the performance of different paradigms on the query set without prior training. In the scenario of one-shot learning, we randomly selected an individual sample labelled as "different" and another labelled as "consistent" from the training dataset, thus constituting a support set for training. Subsequently, we evaluated the performance of diverse paradigms on this constructed support set. Figures 6 and 7 depict the $F1$ scores for zero-shot and one-shot learning on the iTunes-Amazon and DBLP-Scholar datasets, respectively. It becomes evident that, in the context of the iTunes-Amazon dataset, the $F1$ scores yielded by the fine-tuning are notably inferior in both zero-shot and one-shot learning when contrasted with the outcomes of the prompt learning. It is worth noting that in any few-shot learning scenario, the hard prompt consistently attains the highest $F1$ score. The outcomes derived from the DBLP-Scholar dataset substantiate a similar assertion, wherein the prompt learning surpasses the performance of fine-tuning. This congruity echoes the observations drawn from the experiment detailed in Section 4.4.3, particularly during the early training stages, underscoring the efficacy of the hard prompt paradigm in the context of few-shot learning.

5. Discussion

5.1. Performance of Different Models. The experimental results presented in Tables 3 and 4 provide a comprehensive assessment of the proposed EM model in comparison with four established methods: DeepER [38], DeepMatcher [37], Magellan [40], and MCA [41]. Among these four models, some are based on deep learning architectures such as RNN

Table 7: The loss values of different methods at each training epoch on the structured iTunes-Amazon dataset.

Epochs	FT	HP	SP
1	0.663	0.563	0.624
2	0.367	0.366	0.404
3	0.123	0.156	0.225
4	0.0493	0.0683	0.0430
5	0.0464	0.0469	0.0301
6	0.0227	0.0244	$7.97e-4$
7	0.0457	0.0346	$5.12e-5$
8	0.0129	$2.45e-4$	$7.64e-5$
9	0.0106	$1.01e-4$	$6.42e-5$
10	$1.35e-3$	$2.34e-4$	$6.27e-5$

Table 8: The loss values of different methods at each training epoch on the structured DBLP-Scholar dataset.

Epochs	FT	HP	SP
1	0.1561	0.1158	0.1314
2	0.0711	0.0663	0.0720
3	0.0477	0.0483	0.0508
4	0.0325	0.0299	0.0307
5	0.0204	0.0203	0.0203

or LSTM. Even with the incorporation of attention mechanisms, these models can only capture semantic knowledge from the training set, constraining their potential for performance enhancement. Some models introduce word embeddings like GloVe [39], but the semantic knowledge embedded in them falls short of the richness found in PLMs. In contrast, our model leverages the advantages of the PLM, namely, ERNIE-2.0-base-en, to generate enriched representations with contextual information, greatly benefiting the EM task. Furthermore, we incorporate prompt learning to train the EM model. Prompt learning utilizes both the parameters in the PLM's encoder and decoder. Therefore, it narrows the gap between the pretraining task and downstream task (in this case, entity matching), enabling the model to conduct entity matching similar to the pretraining task. "$\Delta F1$" column clearly demonstrates the improvement in $F1$ scores achieved by our model compared to the previous methods. This improvement underscores the efficacy of our approach across diverse datasets (structured and dirty), reaffirming its robustness in various data contexts.

5.2. Performance of Different Paradigms

5.2.1. Comparison of F1 Scores. We also compared the performance of fine-tuning and prompt learning paradigms on both structured and dirty datasets, and the corresponding results are presented in Tables 5 and 6. Evidently, the prompt learning consistently exhibits superior performance over the fine-tuning across the majority of datasets, highlighting its robustness and universality. However, exceptions were observed in the case of structured iTunes-Amazon and DBLP-Scholar datasets, where the adoption of fine-tuning and hard prompt yielded $F1$ scores that were almost indistinguishable. As mentioned earlier, this could be attributed to dataset characteristics, including the number of attributes and the size of training sets. Given that the prompt learning diminishes the gap between pretraining tasks and downstream tasks, it is more suitable for small-scale datasets with fewer attributes. For datasets with ample training samples, as training progresses, the model can acquire more task-specific semantic knowledge from the training set. Thus, for the structured iTunes-Amazon and DBLP-Scholar datasets, fine-tuning and hard prompt learning achieved nearly equivalent $F1$ scores. However, soft prompt, compared to the hard one, allows the model to search for a prompt template in the continuous vector space, which is more conducive to prompt learning. Therefore, it consistently obtains the highest $F1$ scores across all datasets.

5.2.2. Comparison of Loss Values. Considering the similarity in $F1$ scores obtained by fine-tuning and hard prompt on the structured iTunes-Amazon and DBLP-Scholar datasets, we also recorded the average loss values at each training epoch for different paradigms, as listed in Tables 7 and 8, to explore the fitting capabilities of different paradigms on the training set over the entire training phrase. The results for the structured iTunes-Amazon dataset indicate that compared to prompt learning, fine-tuning consistently yields higher loss values throughout the entire training process. However, hard prompt, although starting with the lowest loss value, performs less effectively than soft prompt at the end of training. This phenomenon reaffirms the prior analysis that prompt learning, by adopting the MLM for downstream tasks, can leverage the prior semantic knowledge embedded in PLMs more effectively. As a result, prompt learning fits the training set better, resulting in lower loss values than fine-tuning. Additionally, the soft prompt searches for suitable templates in a continuous space. Thus, although it exhibits higher loss values than the hard prompt in the early stages of training, it ultimately achieves the lowest loss value.

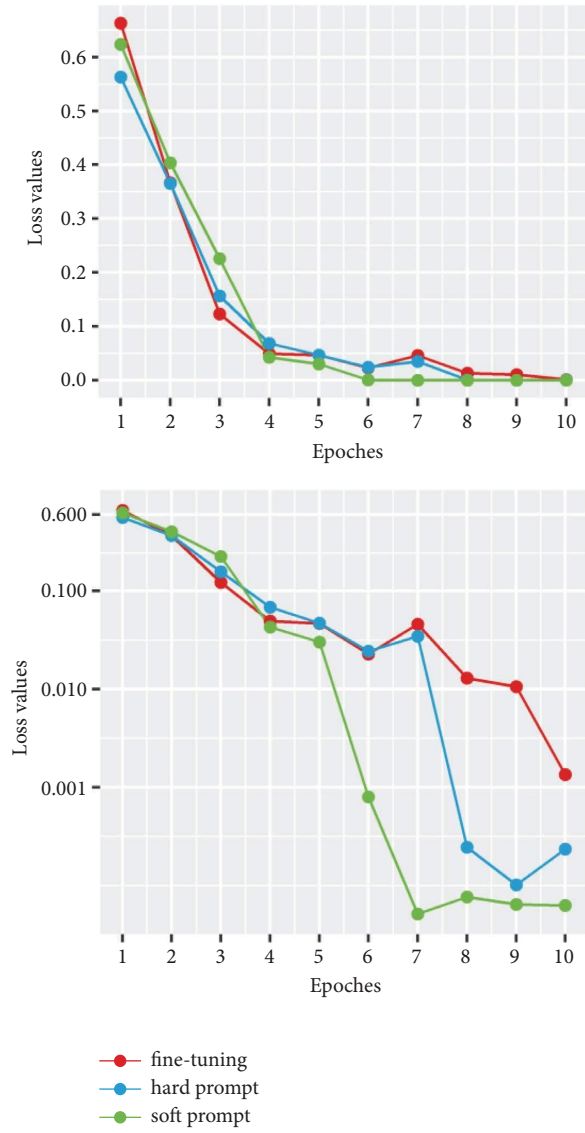


Figure 4: Loss values at each epoch on the structured iTunes-Amazon dataset using different methods. The figure below has taken the logarithm of the loss values using a base of 10.

The experiments conducted on the DBLP-Scholar dataset also demonstrated similar results, indicating that in the early stages of training, fine-tuning exhibited lower fitting capacity to the training set compared to prompt learning, and hard prompt achieved the lowest loss value. However, the final loss value attained by fine-tuning aligns with those of prompt learning. This may still be attributed to the size of dataset, where for a larger number of training samples, fine-tuning can acquire more latent semantic knowledge as training progresses, compensating for its structural differences from prompt learning.

5.2.3. Comparison of the Performance of Few Shots. The preceding discussion elucidates how prompt learning can effectively harness the prior semantic knowledge embedded in PLMs. To further substantiate this assertion, we systematically explored the capabilities of few-shot learning under different paradigms. Experimental results indicate that, compared to prompt learning, fine-tuning yields lower $F1$ scores in both zero-shot and one-shot learning. Regardless of the type of few-shot learning, hard prompts exhibit an advantage in terms of $F1$ scores. This result aligns with the observations detailed in Section 4.4.3, particularly during the initial training phase. Fundamentally, the phenomena observed in few-shot learning can be attributed to the efficacy of the prompt learning in bridging the gap between pretraining and downstream tasks, enabling both soft and hard prompt methods to obtain superior $F1$ scores. Considering that soft prompts require to optimize the template embeddings in continuous space, the experimental outcome further underscores the effectiveness of the hard prompt in the domain of few-shot learning.

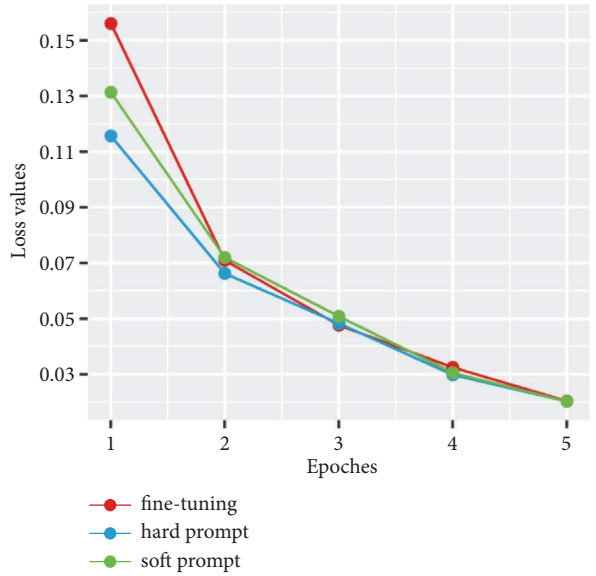


Figure 5: Loss values at each epoch on the structured DBLP-Scholar dataset using different methods.

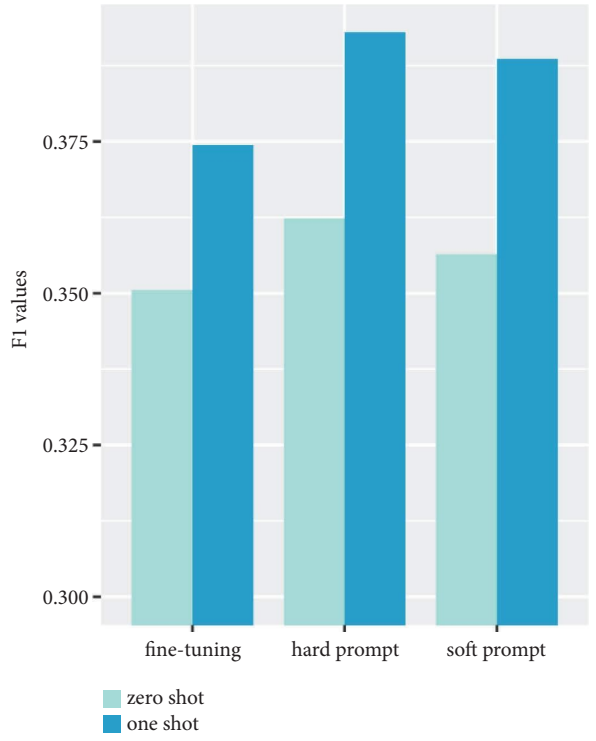


Figure 6: *F*1 scores of zero-shot and one-shot learning on the structured iTunes-Amazon dataset using different methods.

5.3. Limitations and Shortcomings. This study extensively analysed and compared the performance of different paradigms for the EM task. However, our research still has limitations. We investigated the performance of prompt learning based on the BERT-series models, but did not contrast it with large generative language models such as GPT (Generative Pretrained Transformer) [42] and GLM (Generative Language Model). Considering the higher hardware computing resource requirements of the LLM (Large Language Model), we plan to introduce them into the

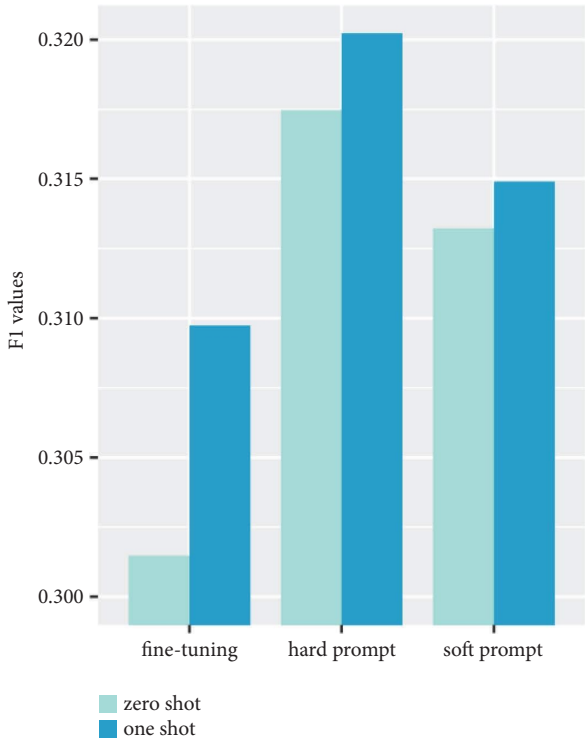


Figure 7: *F*1 scores of zero-shot and one-shot learning on the structured DBLP-Scholar dataset using different methods.

EM task in future work using PEFT (Parameter-Efficient Fine-Tuning) or ICL (In-Context Learning) and CoT (Chain of Thought) techniques.

6. Conclusions

In this study, we have explored the potential of leveraging PLMs to enhance EM. Our investigation involves a comprehensive analysis of two transfer learning paradigms: fine-tuning and prompt learning, across eleven EM datasets. The results indicate that the soft prompt consistently outperforms other approaches across all datasets, demonstrating that generating template embeddings in a continuous space can enhance the performance of EM. Furthermore, our exploration into the realm of few-shot learning unveiled the potential of the hard prompt, showing its effectiveness in both zero-shot and one-shot context. In summary, this research contributes to our understanding of how PLMs can be harnessed to augment EM task. For future work, we will continue to delve into the application of large language models in the EM task. By integrating EM tasks with language models, we aim to enhance knowledge extraction and data integration in various NLP applications [43–45].

Data Availability

The data used to support the findings of this study are available at https://github.com/anhaidgroup/deepmatcher/. The source code is publicly available at https://github.com/Briskyu/entity_matching.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the Medical Big Data Supercomputing Center System of Anhui Medical University. This study was also supported by the Natural Science Foundation of Anhui Province of China (nos. 2108085MH303 and 2108085QF274) and the Key Program of Natural Science Project of Educational Commission of Anhui Province (no. 2023AH050589).

References

[1] D. Van Assche, T. Delva, G. Haesendonck, P. Heyvaert, B. De Meester, and A. Dimou, "Declarative RDF graph generation from heterogeneous (semi-) structured data: a systematic literature review," *Journal of Web Semantics*, vol. 75, 2023.

[2] L. Asprino, E. Daga, A. Gangemi, and P. Mulholland, "Knowledge Graph Construction with a façade: a unified method to access heterogeneous data sources on the Web," *ACM Transactions on Internet Technology*, vol. 23, pp. 1–31, 2023.

[3] B. Hui, L. Zhang, X. Zhou, X. Wen, and Y. Nian, "Personalized recommendation system based on knowledge embedding and historical behavior," *Applied Intelligence*, vol. 13, 2023.

[4] F. Zhao, Y. Li, J. Hou, and L. Bai, "Improving question answering over incomplete knowledge graphs with relation prediction," *Neural Computing & Applications*, vol. 18, 2022.

[5] Q. Guo, S. Cao, and Z. Yi, "A medical question answering system using large language models and knowledge graphs," *International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 8548–8564, 2022.

[6] Y. Li, J. Li, Y. Suhara, J. Wang, W. Hirota, and W. C. Tan, "Deep entity matching: challenges and opportunities," *Journal of Data and Information Quality*, vol. 13, pp. 1–17, 2021.

[7] S. N. Minton, C. Nanjo, C. A. Knoblock, M. Michalowski, and M. Michelson, "A heterogeneous field matching method for record linkage," in *Proceedings of Fifth IEEE International Conference on Data Mining (ICDM'05)*, pp. 226–233, Washington, DC, USA, November 2005.

[8] N. Barlaug and J. A. Gulla, "Neural networks for entity matching: a survey," *ACM Transactions on Knowledge Discovery from Data*, vol. 15, no. 3, pp. 1–37, 2021.

[9] C. Fu, X. Han, J. He, and L. Sun, "Hierarchical matching network for heterogeneous entity resolution," in *Proceedings of the International Conference on International Joint Conferences on Artificial Intelligence*, pp. 3665–3671, Yokohama, Japan, January 2021.

[10] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the Advances in Neural Information Processing Systems*, Long Beach, CA, USA, June 2017.

[11] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186, Minneapolis, MI, USA, June 2019.

[12] Y. Sun, S. Wang, Y. Li et al., "ERNIE: enhanced representation through knowledge integration," in *Proceedings of the 57th*

Annual Meeting of the Association for Computational Linguistics, pp. 1441–1451, Florence, Italy, July 2019.

[13] C. Jia, Y. Shi, Q. Yang, and Y. Zhang, "Entity enhanced BERT pre-training for Chinese NER," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6384–6396, Online, November 2020.

[14] S. Chatterjee and D. Laura, "BERT-ER: query-specific BERT entity representations for entity ranking," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1466–1477, Madrid, Spain, July 2022.

[15] K. Xue, Y. Zhou, Z. Ma, T. Ruan, H. Zhang, and P. He, "Fine-tuning BERT for joint entity and relation extraction in Chinese medical text," in *Proceedings of the IEEE international conference on bioinformatics and biomedicine (BIBM)*, pp. 892–897, San Diego, CA, USA, August 2019.

[16] L. Fichtel, J. C. Kalo, and W. T. Balke, "Prompt tuning or fine-tuning-investigating relational knowledge in pre-trained language models," in *Proceedings of the Conference on Automated Knowledge Base Construction*, Irvine, IR, USA, October 2021.

[17] W. Jin, B. Zhao, Y. Zhang, J. Huang, and H. Yu, "WordTransABSA: enhancing Aspect-based Sentiment Analysis with masked language modeling for affective token prediction," *Expert Systems with Applications*, vol. 238, 2024.

[18] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[19] L. Li, Y. Zhang, and L. Chen, "Personalized prompt learning for explainable recommendation," *ACM Transactions on Information Systems*, vol. 41, no. 4, pp. 1–26, 2023.

[20] G. Jiang, S. Liu, Y. Zhao, Y. Sun, and M. Zhang, "Fake news detection via knowledgeable prompt learning," *Information Processing & Management*, vol. 59, 2022.

[21] X. Wang, K. Zhou, J. R. Wen, and W. X. Zhao, "Towards unified conversational recommender systems via knowledge-enhanced prompt learning," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1929–1937, Washington, DC, USA, August 2022.

[22] B. Zhao, W. Jin, Y. Zhang, S. Huang, and G. Yang, "Prompt learning for metonymy resolution: enhancing performance with internal prior knowledge of pre-trained language models," *Knowledge-Based Systems*, vol. 279, 2023.

[23] T. Schick and H. Schütze, "Few-shot text generation with pattern-exploiting training," in *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, pp. 390–402, Abu Dhabi, UAE, December 2020.

[24] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3045-–3059, Abu Dhabi, UAE, December 2020.

[25] A. E. Monge and E. Charles, "The field matching problem: algorithms and applications," *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining*, vol. 2, pp. 267–270, 1996.

[26] M. Bilenko, R. Mooney, W. Cohen, P. Ravikumar, and S. Fienberg, "Adaptive name matching in information integration," *IEEE Intelligent Systems*, vol. 18, no. 5, pp. 16–23, 2003.

[27] M. Bilenko and R. J. Mooney, "Adaptive duplicate detection using learnable string similarity measures," in *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 39–48, Washington, DC, USA, July 2003.

[28] V. Di Cicco, D. Firmani, N. Koudas, P. Merialdo, and D. Srivastava, "Interpreting deep learning models for entity resolution: an experience report using LIME," in *Proceedings of the International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*, pp. 1–4, Amsterdam, Netherlands, June 2019.

[29] H. Nie, X. Han, B. He et al., "Deep sequence-to-sequence entity matching for heterogeneous entity resolution," in *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 629–638, Beijing, China, October 2019.

[30] N. Kooli, R. Allesiardo, and E. Pigneul, "Deep learning based approach for entity resolution in databases," in *Proceedings of the Asian conference on intelligent information and database systems*, pp. 3–12, Dong Hoi City, Vietnam, February 2018.

[31] R. D. Gottapu, C. Dagli, and B. Ali, "Entity resolution using convolutional neural network," *Procedia Computer Science*, vol. 95, pp. 153–158, 2016.

[32] J. Kasai, K. Qian, S. Gurajada, Y. Li, and L. Popa, "Low-resource deep entity resolution with transfer and active learning," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5851–5861, Florence, Italy, August 2019.

[33] C. Zhao and H. Yeye, "Auto-em: end-to-end fuzzy entity-matching using pre-trained deep models and transfer learning," in *Proceedings of the World Wide Web Conference*, pp. 2413–2424, San Francisco, CA, USA, April 2019.

[34] M. Akbarian, K. Ehsan, and R. Davood, "Probing the robustness of Pre-trained Language Models for entity matching," in *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 3786–3790, Atlanta, GA, USA, October 2022.

[35] T. Schick and H. Schütze, "It's not just size that matters: small language models are also few-shot learners," in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 2339–2352, Washington, DC, USA, July 2020.

[36] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "Autoprompt: eliciting knowledge from language models with automatically generated prompts," in *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4222–4235, Abu Dhabi, UAE, December 2020.

[37] S. Mudgal, H. Li, T. Rekatsinas et al., "Deep learning for entity matching: a design space exploration," in *Proceedings of the International Conference on Management of Data*, pp. 19–34, Houston, HL, USA, June 2018.

[38] M. Ebraheem, S. Thirumuruganathan, S. Joty, M. Ouzzani, and N. Tang, "Distributed representations of tuples for entity resolution," *Proceedings of the VLDB Endowment*, vol. 11, pp. 1454–1467, 2018.

[39] P. Jeffrey, S. Richard, and M. Christopher, "GloVe: global vectors for word representation," in *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, UAE, December 2014.

[40] P. Konda, S. Das, P. Suganthan G C et al., "Magellan: toward building entity matching management systems," *Proceedings of the VLDB Endowment*, vol. 9, no. 12, pp. 1197–1208, 2016.

[41] D. Zhang, Y. Nie, S. Wu, Y. Shen, and K. L. Tan, "Multi-context attention for entity matching," in *Proceedings of the Web Conference (WWW)*, pp. 2634–2640, Taipei, Taiwan, April 2020.

[42] B. Zhao, W. Jin, J. Del Ser, and G. Yang, "ChatAgri: exploring potentials of ChatGPT on cross-linguistic agricultural text classification," *Neurocomputing*, vol. 557, 2023.

[43] W. Cohen, "Data integration using similarity joins and a word-based information representation language," *ACM Transactions on Information Systems*, vol. 18, no. 3, pp. 288–321, 2000.

[44] S. Sarawagi and A. Bhamidipaty, "Interactive deduplication using active learning," in *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 269–278, New York, NY, USA, July 2002.

[45] P. Ravikumar and W. Cohen, "A hierarchical graphical model for record linkage," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, Banff, Canada, August 2004.