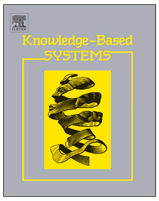
Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Prompt learning for metonymy resolution: Enhancing performance with internal prior knowledge of pre-trained language models

Biao Zhao [a,1], Weiqiang Jin [a,1], Yu Zhang [a], Subin Huang [c], Guang Yang [b,*]

[a] *School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China*
[b] *Bioengineering Department and Imperial-X, Imperial College London, London, W12 7SL, UK*
[c] *School of Computer and Information, Anhui Polytechnic University, Wuhu, Anhui, 241000, China*

ARTICLE INFO

Article history:
Received 25 April 2023
Received in revised form 7 August 2023
Accepted 21 August 2023
Available online 23 August 2023

Dataset link: https://github.com/albert-jin/PromptTuning2MetonymyResolution

Keywords:
Metonymy resolution
Prompt learning
Pre-trained language model
Prompting template engineering
Answer engineering

ABSTRACT

Linguistic metonymy is a common type of figurative language in natural language processing (NLP), where a concept is represented by a closely associated word or phrase, for example "business executives suits". As a result, metonymy resolution has become an important NLP task aimed at correctly identifying metonymic expressions within sentences. Previous approaches to this task have typically relied on pre-trained language models (PLMs) using a fine-tuning process. However, this can be time-consuming and resource-intensive, and may lead to a loss of factual prior knowledge. The emergence of a novel learning paradigm termed "prompt learning" or "prompt-tuning" has recently sparked widespread interest and captured considerable attention, as it has proven to yield remarkable results and surpass previous benchmarks. This approach uses a "pre-train→prompt→predict" paradigm and has been shown to better utilize the internal prior knowledge of a PLM, especially in situations with limited supervised resources. Inspired by this success, we investigated how prompt learning could improve metonymy resolution. We have developed a series of prompt learning approaches, called *PromptMR*, for metonymy resolution, and applied them to several widely-used metonymy resolution datasets. We also designed additional prompt-tuning augmentation strategies to further enhance the potential of prompt learning. Our experiments demonstrated that our method achieved state-of-the-art performance over multiple competitive baselines in both data-sufficient and data-scarce scenarios. The code implementations for *PromptMR* are accessible on GitHub via the URL: https://github.com/albert-jin/PromptTuning2MetonymyResolution.

1. Introduction

Metonymy is a common linguistic figure of speech in our daily conversations where the original meaning of a concept in a sentence is replaced with a closely associated attribute [1,2]. A simple example of metonymy is referring to a business executive as a "suit". However, there are more complex examples, such as "*Malaysia* has competed in the 7th Asian Youth Netball Championship in India in 2010", where the entity "*Malaysia*" refers to the "*Malaysia National Sports Delegation*", and is a substitute for its primary meaning, which is "country". Over recent decades, the metonymic linguistic phenomenon has had a profound impact on natural language processing (NLP), with methods for accurately detecting metonymic entities being particularly and directly beneficial to NLP applications such as geographical parsing [3,4] and named entity recognition [5,6].

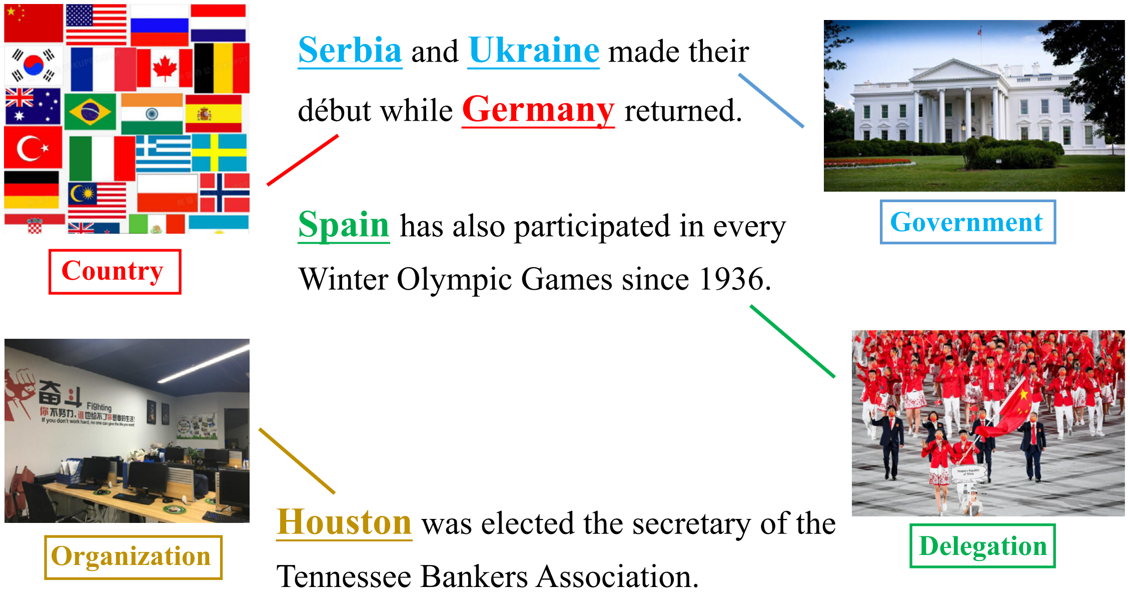
However, in NLP, there are no explicit provisions for exactly how to handle metonymy in a sentence, nor is there any consistent-agreed linguistic constraints without considering the linguistic factors. For instance, taking named entity recognition as an example,[2] Fig. 1 illustrates that based on the contextual semantics: (1) the entities "*Serbia*" and "*Ukraine*" in the first sentence should be classified as "governments" not "countries"; (2) the entity "*Spain*" in the second sentence represents the "Spainish national sports delegation" and not "countries"; (3) and "*Houston*" in the third sentence essentially denotes "governments". As can be seen, inaccurately treating these metonymic entities as countries could result in unexpected issues for downstream linguistic services.

[*] Corresponding author.
E-mail addresses: biaozhao@xjtu.edu.cn (B. Zhao), weiqiangjin@stu.xjtu.edu.cn (W. Jin), zy_yadx@163.com (Y. Zhang), subinhuang@ahpu.edu.cn (S. Huang), g.yang@imperial.ac.uk (G. Yang).

[1] Both authors contributed equally to this work.

[2] These examples are taken from the *ReLocaR* dataset of English Wikipedia.

https://doi.org/10.1016/j.knosys.2023.110928
0950-7051/



Serbia and Ukraine made their début while Germany returned.

Spain has also participated in every Winter Olympic Games since 1936.

Houston was elected the secretary of the Tennessee Bankers Association.

Fig. 1. Several typical metonymic phenomena in a named entity recognition task.

Early traditional machine learning methods [7,8] rely heavily on feature engineering, which involves defining and extracting grammar and syntactic patterns to provide models with the appropriate inductive bias. However, this process is time-consuming and requires both the knowledge of domain experts and a good deal of manual efforts. By contrast, the recent approaches based on neural networks, have delivered significant performance improvements due to their ability to automatically recognize semantics. Moreover, the neural architectures incorporating PLMs have performed particularly well. These approaches typically follow the fine-tuning paradigm, where the PLMs adapt themselves to task-specific objectives by optimizing their significant parameters [9]. For instance, Li et al. [2] proposed an end-to-end word-level classification approach that uses a pre-trained BERT model to generate a context representation by masking the target word. Similarly, Du et al. [1] introduced a novel method for metonymy resolution called "Entity BERT with Attention-guided Graph Convolutional Network" *(EBAGCN)* that integrates the entity constraints from a pre-trained BERT model and soft constraints from syntactic dependency trees, which achieves state-of-the-art results.

However, a potential drawback of these PLM-based approaches is their over-reliance on the encoder portion of the Transformers. Many disregard the decoder component of the framework, instead, use an externally-adopted classifier to complete downstream tasks. As a result, these fine-tuning-based methods of metonymy resolution may fail to take full advantage of the factual knowledge embedded within the PLMs. Further, their ultimate performance is highly dependent on the quality and quantity of the training data. As such, conventional BERT-based fine-tuning methods [1,2,10,11] may easily over-fit the models and fail to deliver satisfactory results in scenarios with limited data resources, such as few-shot settings.

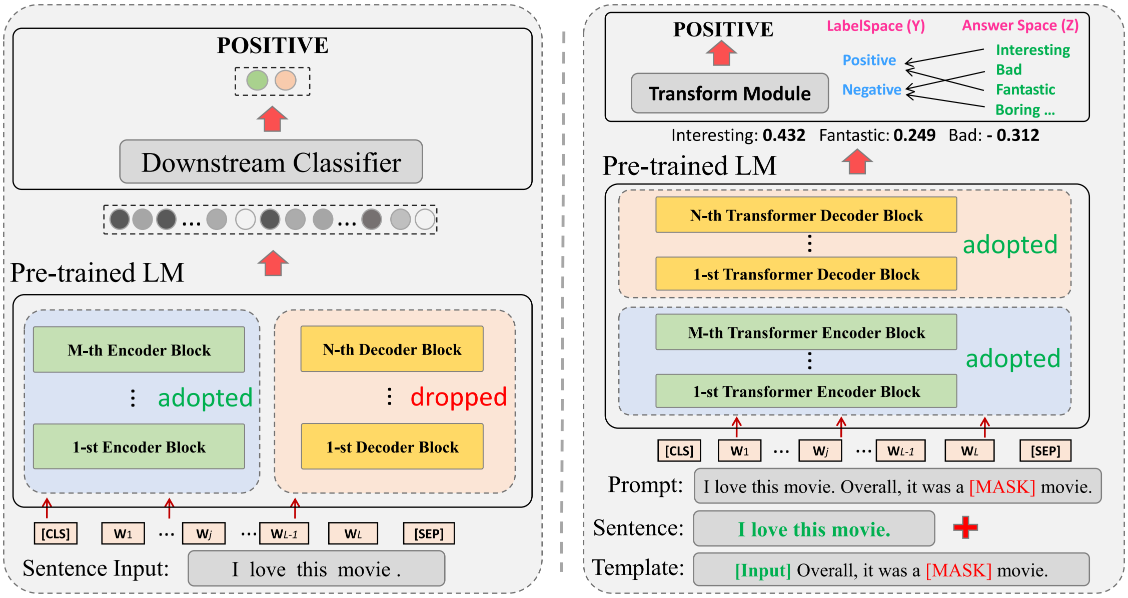
To address this limitation, *prompt learning*, also known as *prompt-tuning*, has emerged as a promising paradigm in the NLP community [9]. Prompt learning uses a text prompt to reformulate the fine-tuning approach to training, integrating it with a mask word prediction process that resembles the pre-training phase of a language model [12,13]. As shown in Fig. 2, this approach provides a more efficient way to harness the expansive prior knowledge contained in the PLMs, which ultimately leads to improved performance in low-resource scenarios. Having been greatly impressed by the achievements made with this new-generation paradigm, we are particularly curious about the potential value of adapting prompt learning to metonymy resolution. Specifically, we asked ourselves: "How much of an increase in performance would we see if we adapted prompt learning to metonymy resolution?" Additionally, we wondered, "Can prompt-based solutions for metonymy resolution perform well in scenarios with limited data resources?"

With these hypothetical questions in mind, in this paper, we conducted a series of investigations into the adaptation of prompt learning for metonymy resolution. Our research comprises two major components: (1) an in-depth study of the mutual relationship between the prompt learning paradigm and metonymy resolution tasks, and (2) a systematic analysis of various specific experimental factors, such as model's architecture and data preparation, along with corresponding strategies and model implementation techniques. Additionally, we explored how to fully leverage the potential of prompt learning in metonymy resolution [9] by employing several superior *prompt-tuning* strategies for both prompting template engineering and answer engineering. Each strategy is based on the latest techniques available [14–16].

To provide a comprehensive evaluation of how effective prompt learning might be, we selected several representative state-of-the-art methods as baselines for comparison (i.e., traditional full-supervised fine-tuning approaches) [1–3] for experimental comparisons. Further, to assess prompt learning in low-resource scenarios, we followed a random instance sampling strategy for each benchmark and conducted a series of comparative experiments in various different few-shot settings. We anticipate that this exploratory work will serve as a significant milestone for future studies on prompt learning for metonymy resolution.

The primary contributions of this paper are therefore summarized as follows.

- Motivated by the success of prompt learning [9], this work focuses on a series of prompt-based approaches for metonymy resolution tasks. Our method, referred to collectively as "Prompt Learning for Metonymy Resolution" *(PromptMR)*, is the first systematic attempt to adapt the prompt-tuning paradigm to metonymy resolution research in the field of NLP.



(a) Traditional PLMs Fine-Tuning Paradigm

(b) PLMs Prompt-Tuning Paradigm

Fig. 2. A comparative analysis of the (a) fine-tuning and (b) prompt-tuning paradigms with respect to aspect-based sentiment analysis (ABSA) at the concept level. In the case of prompt-tuning, the lower region signifies prompting template engineering, while the upper region signifies answer engineering. The employed pre-trained language model sits in the central position.

- In prompting learning for prompting template engineering, we explored the potential of the "Automated Continuous Template Construction" strategy [16] as well as the "Discrete Template Construction" strategy, where templates are manually-crafted.
- In prompting learning for answer engineering, we further explored the "Automatic Continuous Answer Search" strategy [14] as well as the "Discrete Answer Search", where we manually-crafted a mapping strategy that uses an answer word set to predict the final labels.
- Extensive comparative experiments demonstrate that the prompt learning methods are more than competitive when compared to the current state-of-the-art methods of metonymy resolution. PromptMR delivered strong classification performance in both data-sufficient scenarios (i.e., supervised learning with full data) and data-scarce scenarios (i.e., few-shot learning), surpassing the current baseline methods.
- To facilitate further research on prompt learning for metonymy resolution, we have released the code for our approach on GitHub.[3]

The remainder of this paper is structured as follows: Section 2 provides a brief introduction to the related literature on metonymy resolution tasks and the emerging prompt learning technique. In Section 3, we describe several experimental preliminary concepts before experiments. Section 4 then introduces the architectural details of PromptMR and its variants. Section 5 presents the extensive experiments and corresponding analysis. Section 6 and Section 7 further provide the ablated experimental studies and detailed methodology discussions, respectively. Finally, in Section 8, we conclude the paper with our primary contributions and discuss promising research directions for the future.

2. Related works

The current paper's introduction and discussion sections presume a level of familiarity with the relevant literature on both prompt learning [9,13,17], and metonymy resolution [3,7]. Our work is closely connected to Metonymy Resolution (MR) and the recently proposed Prompt Learning technique. Therefore, we will provide a concise overview of the related research, encompassing previous approaches to metonymy resolution as well as the advancements in prompt learning based on a wide range of the most recent literature.

2.1. Metonymy resolution

Metonymic expressions are common phenomena of linguistic syntactico-semantic violation [2,3]. As such, accurately identifying metonymy is a crucial prerequisite in a range of NLP tasks, including intelligent question answering [18] and information extraction [5,6,19]. This identification process, known as metonymy resolution, is a significant and challenging task that involves determining whether a target entity in a given sentence is being used metonymically or not.

Early studies on metonymy [2,3] treated metonymy resolution as a syntactico-semantic process of selecting restriction violations. These methods leveraged an algorithm called "*Semantic Formulas*" to identify the most suitable metonymic preferences. Markert et al. [20] were the first to reformulate metonymy recognition as a classification task based on the typicality of most metonymic readings. This reformulation aimed to make general disambiguation methods applicable to metonymy resolution. Metonymy resolution is approached as a word sense disambiguation task that aims to differentiate metonymies across different semantic classes, as opposed to individual words. Building on prior work [20], Markert et al. [21] present a supervised classification approach that incorporates the additional feature of syntactic head-modifier relations into metonymy resolution. Specifically, they leveraged the similarities between instances of conventional metonymy to identify the syntactic head-modifier relations, enabling complex inferences from the training to carry over to the

[3] GitHub: https://github.com/albert-jin/PromptTuning2MetonymyResolution Our project is built on OpenPrompt (An Open-Source Framework for Prompt-learning, developed by Tsinghua University).

test instances. Markert et al. [7] further contributed to metonymy resolution through their work on the SemEval-2007 shared task. Here, they provided a formal description, released a dataset, and introduced evaluation measures. Around the same time, Farkas et al. [8] proposed GYDER, which achieved the best score of 85.2% accuracy with SemEval-2007. GYDER effectively used various features of potentially metonymic words in context combined with a maximum entropy classifier. Zarcone et al. [22] proposed a novel "words-as-cues" framework that uses generalized event knowledge to develop hypotheses about logical metonymy interpretation. It reconceptualizes the lexicon as a dynamic system for incremental logical metonymies. Gritta et al. [3] proposed a minimalist method called *Predicate Window* (PreWin), which uses a SpaCy dependency parser [23] to extract features from the dependency labels and the entity head dependency. PreWin then employs a long short-term memory (LSTM) network to identify the predicate based on a single local head dependency relationship.

Traditional machine learning has also been used to resolve metonymy in the above-mentioned studies. However, the recent emergence of pre-trained language models has introduced some significant advantages for many NLP applications, including metonymy resolution [24]. For example, Li et al. [2] have proposed an end-to-end word-level classification approach that leverages a pre-trained BERT model, which generates a context representation by masking the target word. Du et al. [1] proposed a novel deep neural network method for metonymy resolution called *Entity BERT with Attention-Guided GCN* (EBAGCN), which combines a pre-trained BERT model's entity constraints with soft constraints from syntactic dependency trees for metonymy resolution.

Compared to the aforementioned metonymy resolution approaches, our proposed PromptMR is the first attempt to apply prompt learning technique to this task. Unlike traditional machine learning or deep learning-based solutions, PromptMR utilizes prompt engineering and answer engineering strategies to tackle metonymy resolution. The uniqueness of this approach lies in framing the task as an adaptive template engineering and answer engineering problem, achieved through prompt tuning to achieve superior performance. PromptMR offers distinct advantages in metonymy resolution by introducing meticulous template engineering and answer engineering strategies. It outperforms traditional machine learning and deep learning methods, delivering impressive results in both data-sufficient and data-scarce scenarios. PromptMR demonstrates strong classification performance even with limited samples, indicating superior generalization and adaptability. These comparisons highlight the potential of PromptMR in advancing metonymy resolution research.

2.2. Prompt learning

Previous PLM-based approaches to metonymy resolution have primarily focused on the fine-tuning mechanism that adapts the PLMs to downstream tasks [25]. By contrast, prompt tuning uses a textual template-based prompt to reformulate the task as a pre-training procedure for a PLM instead of relying solely on fine-tuning [9].

Prompt learning relies on creating appropriate templates to extract relevant knowledge from PLMs in a process known as template engineering. Within a framework called *LAMA*, Fabio et al. [26] manually crafted cloze templates to better elicit the underlying knowledge from PLMs. Taylor et al. [12] devised *Autoprompt*, which leverages a gradient-guided search algorithm to automatically generate prompts for multiple downstream tasks. Zhong et al. [16] proposed *Optiprompt*, an automatic optimization strategy that finds real-valued vectors for prompts in the continuous embedding space instead of searching for better discrete tokens. Lester and Gu et al. [27,28] developed *Soft Template*, a learnable continuous prompt for the PLM of Transfer Text-to-Text Transformer (T5) [29]. *Soft Template* uses an additional set of k tokens added to the input text that can be optimized during training via back-propagation. *P-Tuning* [30] is a technique that introduces a trainable continuous prompt embedding to automatically search for better prompts, which enhances the representation ability of the template. *P-Tuning v2* [31] is a prompt-tuning approach that builds on the idea of deep prompt tuning [13]. It adds trainable prompt embeddings before the PLM in a parallel manner, while freezing the PLM's parameters. This approach yields competitive performance both across universal scales and with different tasks.

Answer engineering is the final and crucial step of prompt learning. It focuses on designing appropriate mappings to search for effective prompt-based predictions within the original output from the answer space. Hambardzumyan et al. [14] proposed *Word-level Adversarial ReProgramming (WARP)*, which appends adversarial perturbations to the input text and follows a novel technique for verbalized token interpretation to instruct the PLMs for inference. Hu et al. [15] proposed *Knowledgeable Prompt-Tuning (KPT)*, which incorporates one or more external knowledge from knowledge bases into a verbalizer to stabilize. KPT works by refining and expanding the label word space of both the verbalizer and the PLMs.

Ding et al. [17] developed an open-source toolkit called *OpenPrompt*.[4] *OpenPrompt* integrates various prompt-based learning methods, making it easy for researchers to develop and deploy their prompt-tuning systems quickly. Our various implementations of PromptMR and its model variants are based on this *OpenPrompt* framework to ensure the code is readable and the functionality is extensible.

3. Overview

This section presents a preliminary overview of our experiments, including the motivations that inspired this work (see Section 3.1 Motivation), and some preliminary knowledge for readers, including the basic paradigm of prompt learning (see Section 3.2.2, Paradigm Definition of Prompt Learning), and a formal conceptual description of the basic metonymy resolution task (see Section 3.2.1, Task Definition of Metonymy Resolution).

3.1. Motivation

Accurate metonymy resolution is crucial to a range of NLP applications. However, previous methods for handling this issue have relied on fine-tuning PLMs. Such approaches fall under the paradigm of full-supervised learning, which requires a substantial number of training instances. Additionally, fine-tuning a traditional PLM is highly dependent on the quantity and quality of training data available. Yet most current metonymy resolution datasets are sparsely populated. Consequently, there is a substantial gap between perfect accuracy and the current accuracy of today's metonymy resolution methods.

In looking for a way to further increase the accuracy of the current methods, we took notice of prompt learning, which has recently emerged as a novel and disruptive training paradigm for models in the NLP arena. Prompt learning appears to be offering several advantages over traditional fine-tuning methods for tasks such as knowledge-based question answering [18,32]

[4] OpenPrompt, which can be accessed at https://github.com/thunlp/OpenPrompt/.

and named entity recognition [5,6]. Further, prompt learning has proven to be particularly effective in scenarios where training data is scarce [9,33]. However, to the best of our knowledge, this paradigm has never been applied to metonymy resolution. In light of this, we decided to investigate whether prompt-tuning would be of benefit to this important NLP task. Through a series of experimental explorations, we discovered that prompt learning is indeed a better and more effective method of metonymy resolution than the traditional fine-tuning approaches. The task definition and specific adaptations made to apply prompt-tuning to metonymy resolution follow.

3.2. Preliminaries

Here, we provide necessary background information on utilizing prompt learning technique for Metonymy Resolution in this work, ensuring that readers can comprehend the context and significance of our research work.

3.2.1. Task definition of metonymy resolution

Here, we formally introduce some preliminary knowledge regarding the formulations of a standard metonymy resolution task. Consistent with previous works [1,2,22], metonymy resolution is defined as a label classification task that involves identifying whether a target word is a metonymyic [3,7]. It is important to note that the target word, referred to as a potential metonymy, may consist of multiple tokens or lengthy phrases, such as the "*United Kingdom* of *Great Britain* and *Northern Ireland*".

Formally, given a pair of inputs, including an N-length sentence $S = \{w_i\}_{i=1}^{N}$, a target word $T = \{w_j\}_{j=k}^{k+L}$, where T denotes a word span of length L starting at position k and a pre-defined binary label set of $U = \{0, 1\}$. The metonymy resolution system then aims to determine which label in U the target word T belongs to. Here, the labels 0 and 1 correspond to the two categories, "literal" and "metonymic", respectively.

3.2.2. Paradigm definition of prompt learning

In NLP, prompt learning has garnered significant attention as a template-based learning paradigm. This approach utilizes pre-defined text prompts to guide the model's learning process. By incorporating prompts, traditional fine-tuning methods are combined with the capabilities of pre-trained models to enhance performance on specific tasks.

In prompt learning, the training process of the model resembles the pre-training phase of language models, where the model is trained to predict masked words or generate accurate text given a prompt. This training methodology enables the model to acquire extensive language knowledge and semantic reasoning abilities, guided by carefully designed prompts that serve as learning objectives. The fundamental theory of prompt learning encompasses two essential elements: template engineering and answer engineering. Template engineering involves the design and construction of effective prompts to ensure that the model can accurately understand and execute the given task. Templates often consist of specific syntactic structures, keywords, and contextual information that guide the model in generating appropriate answers or performing relevant reasoning. Answer engineering focuses on generating suitable responses based on the requirements of the task, including selecting appropriate generation methods, controlling the length of answers, and determining the output format. By employing well-designed prompts, it is possible to integrate the extensive knowledge of large-scale pre-trained models with the specific requirements of particular tasks, leading to improved performance and efficiency.

In summary, prompt learning is a template-based learning paradigm in NLP that utilizes well-designed prompts to guide the model's learning process, enabling it to adapt better to specific task requirements. This theoretical framework provides new perspectives and approaches for research and applications in the field of NLP, driving its development and advancement.

4. Model architectures

In this section, we present the specific structures of our metonymy resolution methods, collectively known as *PromptMR*. This set of solutions comprises on the following models: (1) PromptMR-base — a primitive and basic prompt-tuning method for metonymy resolution (see Section 4.1); (2) PromptMR-CTC — which relies on a continuous template construction strategy (see Section 4.2.1); (3) PromptMR-CAS — which relies on a continuous answer search strategy (see Section 4.2.2); and (4) CTC-PromptMR-CAS — which incorporates both the continuous template construction strategy and the answer search infused strategy (see Section 4.2.3).

Each of these PromptMR variants are built on the generic prompt-tuning paradigm described in Liu et al.'s survey [9]. The basic prompting framework, as shown on the left side of Fig. 3, can be divided into three fundamental steps.

- **Template Engineering** [16,27,28] Template engineering is a process that aims to generate a filled template (a prompt) by constructing a blank template (either a discrete template or a continuous template) for a series of natural language inputs. The prompt is then fed into the PLMs to predict intermediate (candidate) answers at the [*MASK*] position.
- **PLM Prediction** [25,29,34] The adopted PLMs are fed generated prompts, and then searches for a series of the highest-scoring tokens that maximize the PLM's score as intermediate answers.
- **Answer Search Engineering** [9,14,15] Answer search engineering involves reasoning the final labels from the highest-scoring intermediate answers. For example, in sentiment analysis, the model may predict that the final answer is "positive" not "negative" because the output probability distribution of sentiment-bearing words such as "happy", "satisfactory", and "wonderful" are higher than words like "sad", "disgrace", and "embarrassed".

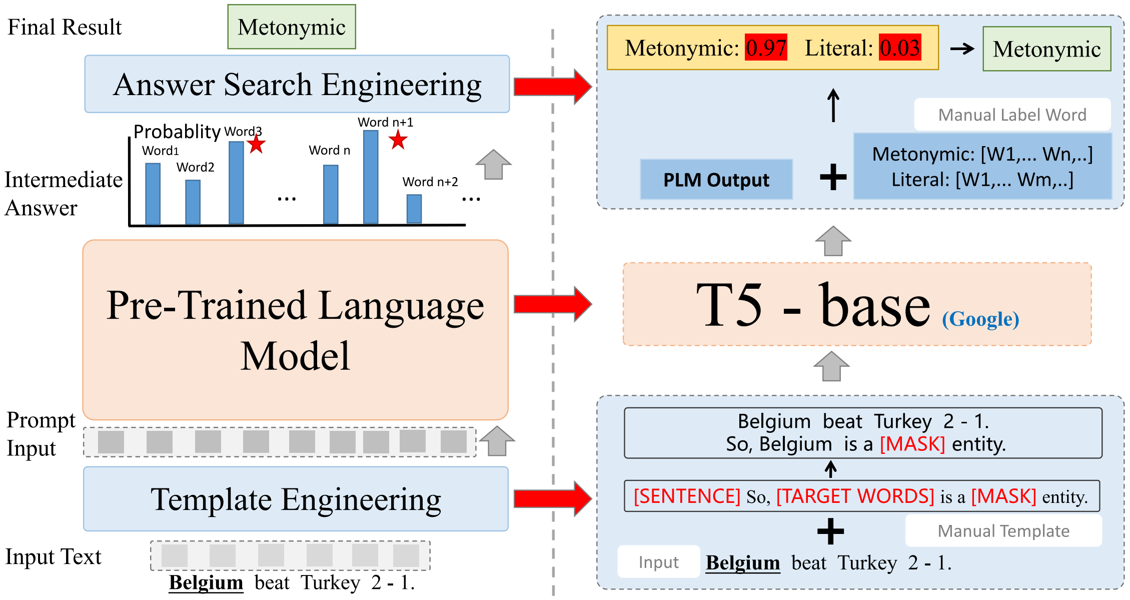
4.1. Basic prompt learning for metonymy resolution

In this section, we present a basic implementation of prompt learning for metonymy resolution, i.e., PromptMR-base, where only the PLM's parameters are learnable. Notably, both the template engineering and the answer engineering are manually designed and cannot be optimized during training.

This means the implementation of PromptMR-base involves using a manually-designed discrete template for template engineering, the Transfer Text-to-Text Transformer (T5) as the PLMs [29], and the generic answer mapping strategy [9] for the answer engineering scheme. The architecture is illustrated on Fig. 3.(b).

The model's architecture is detailed as follows:

1. **Manual Discrete Template** The bottom of Fig. 3.(b). depicts the process of manually engineering the discrete template. In this case, the sentence S = "Belgium beat Turkey 2-1" and the entity E = *Belgium* are given. The first step is to construct a template T: "[*SENTENCE*] So, [*TARGET WORDS*] is a [*MASK*] entity"., which is a text string that has three slots: two input slots for the inputs [*SENTENCE*], and [*TARGET WORD*], along with an answer slot [*MASK*] for the generated candidate answers, which holds the intermediate results of the final prediction. After S and E are respectively fed into the input slot, the template T becomes


Final Result
Metonymic
Answer Search Engineering
Probablity
Word1
Word2
Word3
Word n
Word n+1
Word n+2
...
Intermediate Answer
Pre-Trained Language Model
Prompt Input
Template Engineering
Input Text
Belgium beat Turkey 2 - 1.
Metonymic: 0.97 Literal: 0.03
Metonymic
Manual Label Word
PLM Output
Metonymic: [W1,... Wn,..]
Literal: [W1,... Wm,..]
T5 - base (Google)
Belgium beat Turkey 2 - 1.
So, Belgium is a [MASK] entity.
[SENTENCE] So, [TARGET WORDS] is a [MASK] entity.
Manual Template
Input
Belgium beat Turkey 2 - 1.

(a) Generic Architecture of **Prompt Learning**

(b) The first Exploration **PromptMR-base**

Fig. 3. The left panel illustrates the standard architecture for prompt learning, while the right panel shows the specific implementation of our primary model, PromptMR-base. The sentence is a typical example of metonymy resolution: "Belgium beat Turkey 2 - 1".

"Belgium beat Turkey 2-1. So, Belgium is a [*MASK*] entity". We call this a "*Prompt*" and the sequence is fed directly into the sequence to the adopted PLMs. The discrete template construction procedure is then formulated as follows:

$$Prompt := f_{prompt}(T, S, W) \qquad (1)$$

where f(·) represents the mapping function of the *Prompt* construction; T refers to the constructed templates; and S and W denote the input sentence and the target words, respectively.

2. **Pre-trained Language Model** As shown in the center of Fig. 3.(b), the PLM we used for PromptMR-base is the Transfer Text-to-Text Transformer (T5), a seq2seq language generation model that has been pre-trained on the task of filling in missing spans. The PLM T5, a typical Auto-Regressive Language Model, refers to its ability to generate text in a sequential manner, word by word. Given a prompt or an initial sequence of words, the model predicts the next word and then uses that prediction as input to generate the subsequent word, continuing this process until the desired length of the text is achieved. This auto-regressive approach enables T5 to be effective in various natural language processing tasks such as language translation, summarization, and text completion. With T5, we can re-frame all NLP tasks into a unified text-to-text-format where the input and output are always text strings, in contrast to BERT-style models that can only output either a class label or a span of the input. Thus, T5 is more suitable for performing prompt learning experiments than BERT due to this specific pre-training regime. However, computational limitations prevented us from using T5-large so, we used basic configuration of T5 (i.e. the basic version, T5-base). Specifically, the adopted pre-trained language model Transfer Text-to-Text Transformer (T5)[5] can be accessible at from the open advanced NLP community, Hugging Face. Noting that previous experiments have shown that T5-base's performance is sufficiently competitive to *T5-large* to support our findings [17,29].
The entire reasoning process of the PLMs can be represented as follows:

$$H_L^0 = \mathrm{Layer}_{\mathrm{E}}(Prompt) \qquad (2)$$

$$h_{1:N}^{d_{\mathrm{H}}} = H_L^N := \mathrm{Encoder}^N(Prompt) = \{TransBlock_k\}_{k=1}^N(H_L^0) \qquad (3)$$

$$p(\hat{w}_j) = \mathrm{SOFTMAX}\left(\mathbf{h}_j^{\mathcal{V}}\mathbf{W}_{d_{\mathrm{H}}}^{\mathcal{V}} + \mathbf{b}^{\mathcal{V}}\right) \quad j \in [1, \ldots, N] \qquad (4)$$

where $\mathrm{Layer}_{\mathrm{E}}(\cdot)$ denotes embedding initialization layer, L is the input length, and H_L^0 is the initialized hidden representation before it is fed to the first Transformer block. Encoder represents the T5-base encoder, which comprises N Transformer block layers $\{TransBlock_k\}_{k=1}^N$, where N equals 12. d_{H} is the hidden dimension of the Transformer encoder, which equals 768 with T5-base, while $\mathcal{V}$ denotes the vocabulary length, $\mathbf{W}_{d_{\mathrm{H}}}^{\mathcal{V}}$, and $\mathbf{b}^{\mathcal{V}}$ denotes the trainable weight parameters of the linear layer. Through the softmax normalization layer SOFTMAX(·), $p(\hat{w}_j)$ represents the probability score of the estimated word $\hat{w}_j$ in the jth position.
The probability score of the estimated vocabulary words in the position of [*MASK*] are obtained through the Transformer forward propagation process. These vocabulary words then act as the intermediate answers, which are fed into the answer search engineering.

3. **Manual Discrete Answer** Answer search engineering is one of the most essential steps in metonymy resolution as it directly affects the quality of the model's final performance. In this step, the model searches for the correct label from a pre-defined set of label words based on the probability distributions of the [*MASK*] position provided by the PLM T5 [29] – in this case the T5 PLM. As shown in the top of Fig. 3.(b), we used a manual discrete answer mapping strategy here. More specifically, we manually selected several label words to form a label word set for each of the two categories (i.e. metonymic and literal). Note that this search

[5] The Transformer-based PLM T5 from Hugging Face: https://huggingface.co/t5-base.

function could either be an argmax search that searches for the highest-scoring outputs of the average log logits based on the top K probabilities, or a sampling strategy that randomly generates outputs following the probability distribution of the PLM. During training, the adopted PLM is optimized to maximize the probability scores of the label words, corresponding to the correct category using a cross-entropy objective function. For reference, the averages of the label words in both the metonymic and literal categories are aggregated and calculated, and the highest average-scoring category is regarded as the final prediction. This process can be expressed as follows.

$$\hat{\boldsymbol{l}} := \underset{\boldsymbol{l} \in Labels}{\mathrm{Search}}(P_{LM}(w_{[MASK]}|Prompt, \theta)) \tag{5}$$

$$= \underset{\hat{w}_{[MASK]}}{Argmax} \sum_{i=1}^{K} \frac{1}{K} \log(p(\hat{w}_{[MASK]})). \tag{6}$$

where $\hat{\boldsymbol{l}}$ is the final estimated label from the label categories $Labels = \{$"*metonymic*", "*literal*"$\}$. This label is searched using the answer mapping strategy $Search(\cdot)$, which is based on the outputs from the pre-trained language model, $P_{LM}(w_j|Prompt, \theta)$. Here, θ denotes the trainable parameters of the adopted pre-trained language model.

As demonstrated earlier [11,35], the entire pre-trained language model (PLM) is considered to be replaceable in prompt learning and, therefore, we have purposefully omitted the details of this model. Instead, our attention is focused on template engineering and answer search engineering. After exhaustive testing, we selected the pre-defined text "[*SENTENCE*] So, [*TARGET WORDS*] is a [*MASK*] entity". as our manual template, along with the labels {"metonymic"} and {"literal"} as our two categories. Moreover, as for the strategies of template selection and label word selection in prompt learning, we have organized the investigation of the effects of specific selected templates and label word pairs on the prompt-tuning model performance as an independent section of ablation study (see Section 6), including two specific sub-experiments: the comparison experiments of prompting template selection (see Section 6.1) and the strategy comparisons of label words selection (see Section 6.2).

4.2. Enhanced prompt learning for metonymy resolution

Here, it is of interest to investigate whether the prompts used to awaken the PLM's internal prior knowledge can be further enhanced, and whether the answer search strategy can be further modified to improve overall performance.

To this end, we devised several different variants of PromptMR, each involving enhancments to the template engineering and/or answer engineering steps. Principally, we drew inspiration from previous enhancement strategies for prompt learning and explored the impact of adopting a dynamic template strategy (PromptMR-CTC) and a dynamic answer mapping strategy (PromptMR-CAS) on the final model's performance. Additionally, we combined both strategies to create CTC-PromptMR-CAS, and investigated what this variant might add to metonymy resolution. The architectures of these enhanced PromptMR's variants are depicted in Fig. 4, with details of the experimental results and analysis given in Section 5.5.3. More details of the implementations of each follow in the next few subsections.

4.2.1. CTC-based enhanced PromptMR

As we all know, manually engineering a template requires human involvement and extensive selection work, which is not intelligent enough. It has also been proven that different handcrafted templates can result in substantial performance gaps in the final model [36]. Further, we also know that a manual template that performs well in one dataset may not yield good results with another dataset. Thus, researchers are constantly devising new templates to adapt to new data scenarios. Motivated by previous works that apply continuous templates to prompt-tuning to achieve better results [12,27,28], we conjectured that an automatic method of learning a template might also benefit PromptMR.

Thus, inspired by previous works on soft templates [27,28], we experimented with an enhanced version of PromptMR, called PromptMR-CTC, that uses a continuous soft prompts template strategy to automatically learn templates instead of relying on hand-crafted templates. The soft prompts strategy, which is trainable through back-propagation, was first introduced by Lester et al. [27] and subsequently enhanced by Gu et al. [28]. The enhancing strategy, called "continuous template construction", is a simplified version of prefix tuning [13].

As shown in the bottom right of Fig. 4, several additional tunable tokens are inserted into the input text, which are optimizable during the training procedure. These continuous templates are a series of hybrid-format token sequences that contain both fixed language tokens and trainable embedded tokens (a set of tunable embeddings).

However, unlike a hand-crafted discrete template, which is essentially a natural language sequence, these continuous templates are optimized for specific tasks and can be adapted to new data scenarios. It is an approach that allows for more flexibility and adaptability in different data scenarios. The continuous templates are learned during training through back-propagation, and the output of the PLM is used as a continuous representation to construct the templates. Continuous template construction is then performed iteratively to refine the templates, which ultimately leads to better model performance.

Compared to traditional fine-tuning methods for PLMs, the manually defined template based on prompt learning strategy, while having advantages in eliciting prior knowledge from within PLM, is fundamentally a manually constructed discrete character template that cannot be optimized during the model training process. This limitation becomes one of the primary constraints on the performance of the PLM model, especially when dealing with the supervised data-scarce conditions (few-shot learning), leading to biased fitting results. Therefore, introducing an optimizable continuous template is extremely necessary. With this approach, the optimization strategy for continuous templates transforms the original straightforward parameter updating process for PLMs into a more comprehensive joint updating process involving both PLM and template slot parameters. Consequently, the overall training becomes more sophisticated and refined. Similar to the recently proposed prefix tuning [13], the newly introduced soft prompts engineering not only enhances optimization robustness but also leads to improved final task performance, thereby enabling efficient "prompt ensembling".

As depicted in Fig. 5, the mixed soft template construction strategy inserts the trainable soft tokens into several fixed sequence positions of the concrete word sentence, unlike the manual discrete template shown in the left of the diagram. Moreover, during training, the internal parameters of the pre-trained language model are simultaneously optimized within the parameters of the soft template.

It is important to emphasize that the parameters of the continuous template component are trained following the original soft prompts strategy. The main difference is that: compared to the original work that freezes the entire pre-trained language model, the adopted T5 is simultaneously optimized. It should also be

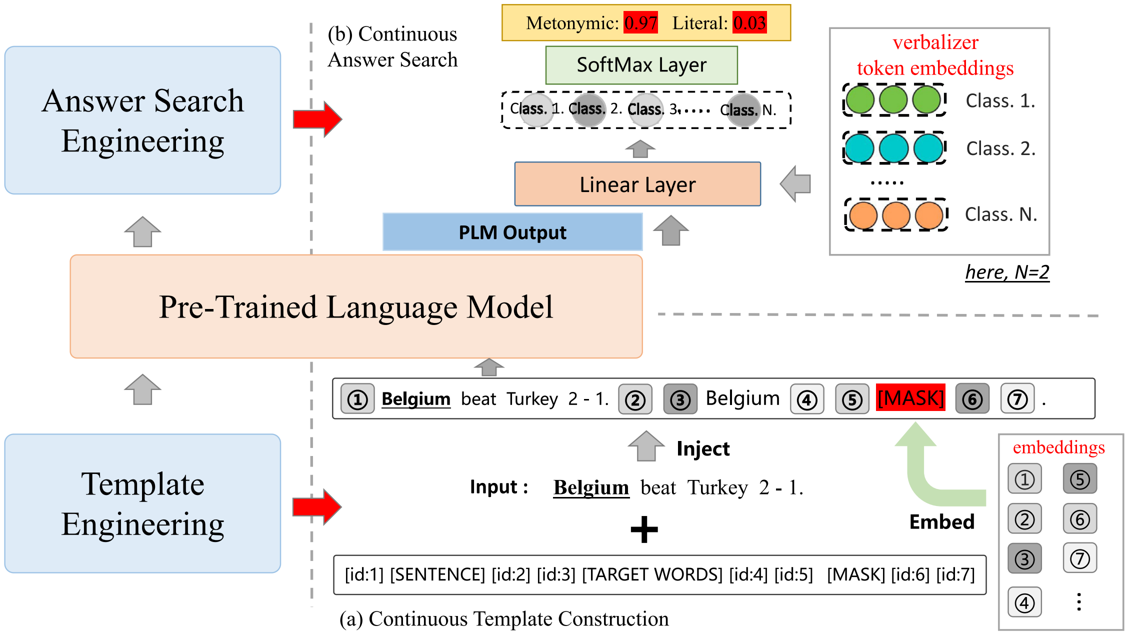
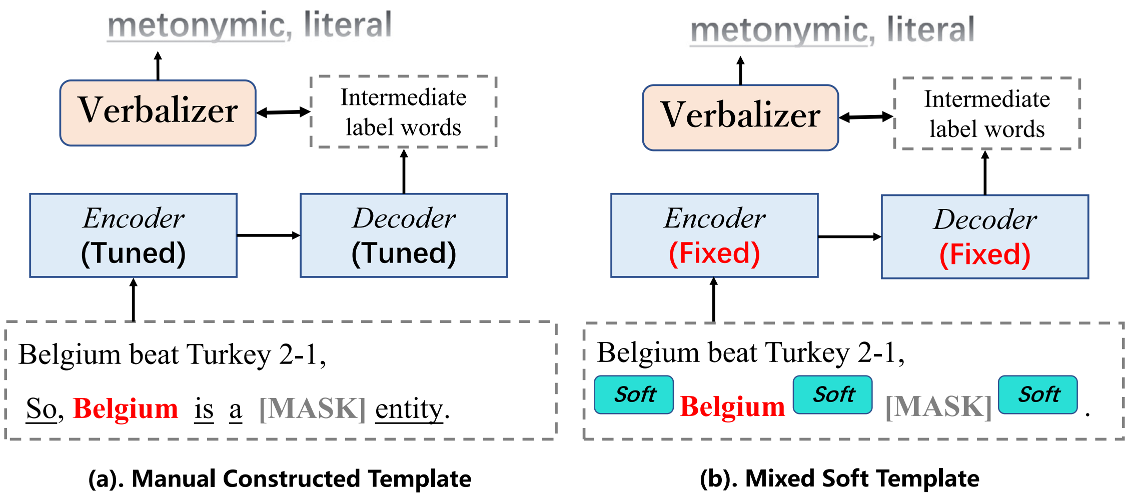

Answer Search Engineering
(b) Continuous Answer Search
Metonymic: 0.97 Literal: 0.03
SoftMax Layer
Class. 1. Class. 2. Class. 3. ..... Class. N.
Linear Layer
verbalizer token embeddings
Class. 1.
Class. 2.
.....
Class. N.
here, N=2
PLM Output
Pre-Trained Language Model
① Belgium beat Turkey 2 - 1. ② ③ Belgium ④ ⑤ [MASK] ⑥ ⑦ .
Inject
Input : Belgium beat Turkey 2 - 1.
+
Embed
embeddings
① ⑤ ② ⑥ ③ ⑦ ④ ⋮
[id:1] [SENTENCE] [id:2] [id:3] [TARGET WORDS] [id:4] [id:5] [MASK] [id:6] [id:7]
Template Engineering
(a) Continuous Template Construction

Fig. 4. The architecture illustration of the enhanced PromptMR. (a) represents the enhanced template engineering (continuous template construction), while (b) represents the enhanced answer engineering (continuous answer search), respectively. In (b), the output dimension is set to 2, which, in this case, equals the number of classification categories.



(a). Manual Constructed Template

(b). Mixed Soft Template

Fig. 5. Training comparisons between an (a) manual constructed template and (b) mixed soft template. "[Soft]" refers to the inserted trainable dynamic template tokens.

noted that, different template initialization methods could have different impacts on the robustness of the training process, and in turn, the final model's performance. Hence, in our experiments, we used two different strategies for initializing the continuous templates. One was to use a random initialization for each soft token (random initialization), and the other was to use the adopted PLM's primitive word representations to initialize the soft tokens (PLM embedding initialization).

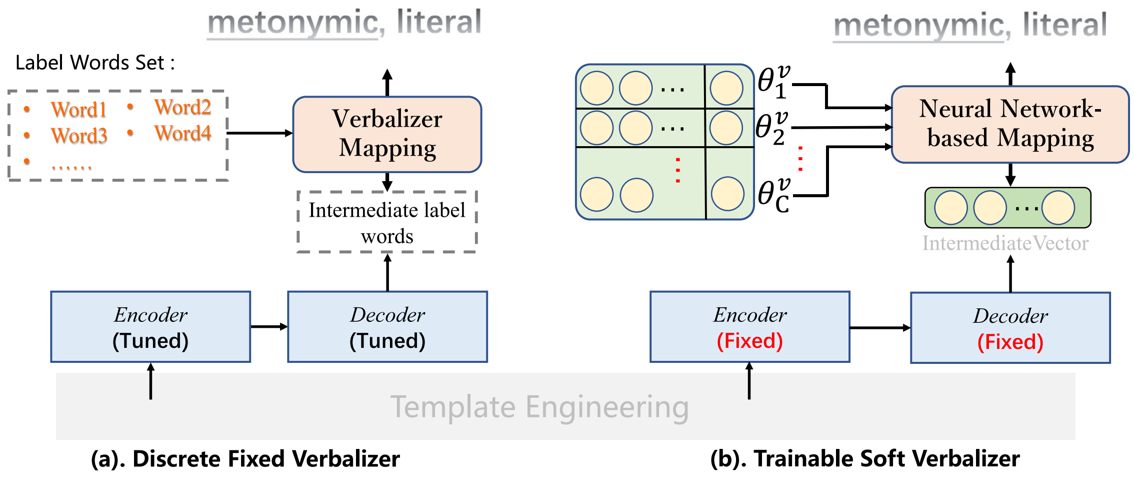
Intuitively, with the second PLM embedding initialization strategy, we converted the discrete template used in PromptMR-base — i.e., "[*SENTENCE*] So, [*TARGET WORDS*] is a [*MASK*] entity". (Section 4.1) — into a new continuous template by embedding the prompting words: {'So', ',', 'is', 'a', 'entity', '.'} into several soft tokens. Note that, in our experiments, we dropped the first initialization during our experiments and only used the second strategy with PromptMR-CTC to get better parameter initialization.

Owing to these adopted soft prompts, PromptMR-CTC appears to deliver better transfer learning capability with a range of different domain tasks without needing to rely on hand-crafted templates. Further, initializing training with the PLM embeddings increases both learning performance and the efficiency of the training times above and beyond what seems to be achievable with human-made discrete templates.

Extensive comparative experiments prove that this enhanced PromptMR-CTC equipped with the continuous template construction strategy improves performance over PromptMR-base. The specific performance impacts of these optimizable templates and the corresponding ablated analysis have been further discussed in Section 6.

4.2.2. CAS-based enhanced PromptMR

Similarly, the discrete answer engineering scheme used in PromptMR-base (Section 4.1) relies on manually-defined candidate answers in the metonymic and literal categories with which to map to the final labels. This approach requires selecting new, well-crafted label words when transferring them to a new



(a). Discrete Fixed Verbalizer

(b). Trainable Soft Verbalizer

Fig. 6. Answer engineering comparisons between the (a) discrete fixed verbalizer strategy and (b) continuous trainable soft verbalizer strategy.

domain, making the whole process labor-intensive and not particularly transferable. Intuitively, to improve answer engineering with prompt learning, we devised a enhanced PromptMR, continuous answer search-based PromptMR (PromptMR-CAS), which replaces the manual discrete answer mapping scheme with a "soft verbalizer" [14].

Such a continuous answer search strategy, akin to the optimization of continuous templates, revolutionizes the cumbersome process of searching for optimal discrete tokens as intermediate answers. It imbues a level of intelligence into the answer shaping strategy, transforming it into a more sophisticated self-optimizing approach. Notably, the inspiration for this strategy stems from Elsayed et al.'s groundbreaking Adversarial Reprogramming method proposed in 2019 [37]. This method involves reprogramming pretrained ImageNet classifiers by incorporating input-level adversarial perturbations to enhance performance on MNIST and CIFAR-10 image classification tasks. This approach has already demonstrated success in the computer vision (CV) domain, prompting us to naturally explore its adaptability to specific tasks in NLP. In particular, we are intrigued by its potential efficacy in the metonymy resolution task. More specifically, within our prompt learning-based metonymy resolution task, the adaptive parameter optimization facilitates automatic gradient-based optimization in the space of word embeddings. In conclusion, this approach optimizes answer engineering by allowing for a continuous answer search, resulting in increased performance and transferability. As a result, the PromptMR model gains increased flexibility and exhibits improved performance in downstream classifications.

As shown in the top right of Fig. 4, PromptMR-CAS uses this "soft verbalizer" [14,15], which is a tunable verbalizer proposed by Karen et al. to replace the manual discrete answer mapping used in PromptMR-base.

Fig. 6 illustrates the differences between (a) using a discrete fixed verbalizer and (b) using the learnable soft verbalizer introduced in continuous answer search-based PromptMR (PromptMR-CAS). Typically, a vector-based verbalizer, $[\theta^{\mathcal{V}_1}, \theta^{\mathcal{V}_1}, \ldots, \theta^{\mathcal{V}_C}]$, denotes the weight matrix of an extra linear layer which are applied as adversarial reprogramming [14]. Formally, given the intermediate vector representation generated from the PLM decoder, in PromptMR-CAS, the successive class-oriented probability calculation procedure is formulated as:

$$P_{\Theta}(l_i \mid x) = \frac{\exp \Theta^{V}_{l_i} P_{LM}(w_{[MASK]} | Prompt, \theta)}{\sum_{j \in C} \exp \Theta^{V}_{l_j} P_{LM}(w_{[MASK]} | Prompt, \theta)} \quad (7)$$

Here, $\Theta^{V}_{l_i}$ denotes the trainable embeddings of the verbalizer for the ith category, and $P_{LM}(w_{[MASK]} | Prompt, \theta)$ denotes the logit outputs of the pre-trained language model. Furthermore, because metonymy resolution is a binary classification task, the number of categories C is set to 2, which equals to the label set $Labels = \{$"*metonymic*", "*literal*"$\}$.

$$\hat{l} = \underset{l_i}{Argmax}\, P_{\Theta}(l_i \mid x) \quad (8)$$

Finally, as shown in Eq. (8), the final predicted label is produced through an argmax search that selects the highest-scoring probability outputs.

Regarding our experimental settings, we generated unique 1-dimensional embedded tokens [T_1], [T_2], ..., [T_C] for all the classification categories 1...C accordingly. We then constructed a tunable verbalizer using a categorical soft token vector for each of the two categories. The soft token vectors were initialized with the word embeddings of the '[MASK]', similar to the vectors from the PLM's word embedding layer. Note that, following the intuition from Hambardzumyan et al. [14], unlike most adversarial attacks, we refrain from updating the embeddings of the original input tokens, except for the tunable verbalizers. Next, we constructed a feed-forward network (comprising fully-connected linear layers) on top of the PLM's outputs. Given the vocabulary probability distributions of the *[MASK]* position (Dim: 1 × Len_v) and the pre-defined soft verbalizer (Dim: Len_v × C), the linear layer would act as a decoder, calculating the class-related logits between two features. With these class-related logits in hand, a softmax function is calculated over the verbalizer tokens to determine the probabilities for each of the two categories (i.e. metonymic and literal). The category with the highest probability is then chosen as the final label for the input text.

4.2.3. CAS and CTC integrated PromptMR

Intuitively, we also devised an enhanced PromptMR model that combines both the continuous template construction and continuous answer search strategies [14,27], called CTC-PromptMR-CAS. This model uses self-adaptive continuous templates and a tunable verbalizer to improve transfer learning performance and reduce training times. Compared to the other PromptMR variants, CTC-PromptMR-CAS is more flexible because it allows the analyst to tune the parameters in all three steps of prompt learning. The downside is that tuning these parameters to ensure robustness in training decreases the learning rate of the PLM (T5) from 1e-4 to 5e-5. Detailed experimental results and a corresponding analysis are presented in the next section, Section 5.5.3.

5. Experiments

In this section, we outline the experiments conducted to evaluate the effectiveness of the proposed PromptMR for metonymy resolution. The information provided includes: (1) the statistics of the adopted datasets we used (see Section 5.1); (2) the adopted comparative baselines (see Section 5.2); (3) the adopted evaluation metrics we used (see Section 5.3); (4) detailed experimental settings for the two learning scenarios evaluated, i.e. supervised training with full data and several few-shot learning scenarios, i.e. low-resource supervised training (see Section 5.4); and (5) comprehensive results of all the experiments with corresponding analyses (see Section 5.5).

5.1. Datasets

We experimented with three established benchmarks that are widely used in this field: SemEval [7], ReLocaR, and CoNLL [3]. All the datasets used are available from Github.[6] Additionally, we have provided a detailed discussion and introduction on the used metonymy resolution datasets in our GitHub page.[7] The specific details of each benchmark are provided below:

- **SemEval**: SemEval [7] is a benchmark dataset presented in SemEval-2007 Task 08 that comprises 3800 sentences from the British National Corpus. It consists of 925 training instances (737 literal and 188 metonymic) and 908 test instances (721 literal and 187 metonymic). The class distribution is roughly 20% metonymic, and 80% literal. SemEval is composed of three annotation categories, namely Locations, Organizations, and Class-independent categories.
- **ReLocaR**: ReLocaR [3] is a benchmark collected from Wikipedia that contains over 2008 sentences, all of which focus on the Locations type. It includes 1026 training instances (509 literal and 517 metonymic) and 982 test instances (486 literal and 496 metonymic). In comparison to SemEval, ReLocaR has a more balanced class distribution of roughly 50% literal and 50% metonymic samples, as well as a higher annotation quality.
- **CoNLL**: CoNLL [3] is a benchmark released alongside ReLocaR that was annotated by a single annotator from CoNLL-2003 NER Shared Task. It focuses on locations and contains approximately 7000 sentences, including 4972 training instances (3283 literal and 1,689 metonymic) and 1243 test instances (806 literal and 437 metonymic). The class ratio of metonymic versus literal is approximately 2:1.

5.2. Baselines

We considered several state-of-the-art baseline methods in comparison to PromptMR, including two feature engineering methods: GYDER and PreWin [8,38], a pre-trained embedding-based method built on an RNN called BiLSTM (GLoVE) [39], and four methods based on PLMs: BiLSTM (ELMo), TWM-BERT, EBGCN, and EBAGCN [1,2,39]. More details on each of these methods follow.

- **SVM+Wikipedia**: Support Vector Machine within Wikipedia (SVM+Wikipedia) [38] is an early probability statistical approach that employs support vector machine (SVM) in conjunction with the rich network of Wikipedia categories and its articles. It aims to automatically discover new relations and metonymies in text.
- **GYDER**: GYDER (the acronym was formed from the initials of the author' first names) [8] is a conventional machine learning method that uses feature engineering to achieve high accuracy with the SemEval-2007 metonymy resolution task. It uses the most relevant features, including grammatical annotations, syntactic and semantic characteristics of potential metonymies.
- **PreWin**: Predicate Window (PreWin) [3] is a minimalist statistical method proposed alongside the benchmark ReLocaR, which uses a predicate window to eliminate noise over long textual distances.
- **BiLSTM (GLoVE)** [39]: The Global Vectors-based Bidirectional LSTM, BiLSTM (GLoVE), is an special RNN that uses Bidirectional Long Short-Term Memory (Bi-LSTM) and incorporates pre-trained word embeddings with the *Global Vectors for Word Representation* (GLoVE) approach [40].
- **BiLSTM (GLoVE)+NER+POS** [39]: This model is built on BiLSTM (GLoVE) and integrates both named entity recognition (NER) and part-of-speech (POS) features to further enhance the model's capabilities.
- **BiLSTM (ELMo)** [39]: In contrast to BiLSTM (GLoVE), *ELMo-based BiLSTM*, BiLSTM (ELMo) employs the deep contextualized word representation model, *Embeddings from Language Models* (ELMo) [41] to replace the fixed embeddings generated by GLoVE. ELMo has been shown to capture contextual semantic features more effectively than traditional word embedding.
- **BiLSTM (ELMo)+NER+POS** [39]: Built on top of BiLSTM (ELMo), this model incorporates both named entity recognition (NER) and part-of-speech (POS) features to further enhance its power.
- **TWM-BERT** [2]: Target Word Masking BERT (Entity BERT) is a word-level classification approach based on a BERT model that uses the average output vectors of all sub-tokens associated with the target word to generate metonymyic semantic representations. These representations are then fed into a linear classifier to generate classification results, labeled as $U = \{0, 1\}$.
- **EBGCN** [1]: Entity BERT with Graph Convolutional Networks (EBGCN) incorporates external syntactic knowledge in the form of hard constraints using the graph convolutional network (GCN) based on *Entity BERT*.
- **EBAGCN** [1]: Entity BERT with Attention-Guided GCN (EBAGCN), proposed alongside EBGCN, uses an attention-guided GCN to convert hard syntactic constraints into soft constraints, which increases the use of external knowledge in NLP tasks.

In order to comprehensively understand the architecture-level characteristics of these baselines, we conducted a series of investigations from metaphor-related literature [1,2,8,38,39] and recorded the comparative results in Table 1. We focused on the following several aspects: Firstly, we examined the degree of data quality dependency, which refers to the sensitivity of different models to data quality. Secondly, we investigated the model training efficiency, including the convergence speed during the training process and the computational resources consumed. We also paid attention to the models' capability to handle long-term sequence dependencies, which is crucial for understanding the contextual nuances of metaphors. Additionally, we considered a range of model characteristics, such as interpretability and robustness in the presence of noise.

Through these comprehensive comparative investigations, we can intuitively assess the strengths and weaknesses of different model architectures and select the most suitable one for the task of metaphor detection. Our research findings provide strong guidance for subsequent model selection and valuable insights for further research and application.

[6] Metonymy resolution dataset sources are from: https://github.com/milangritta/Minimalist-Location-Metonymy-Resolution/tree/master/data.

[7] The datasets introduction: https://github.com/albert-jin/PromptTuning2MetonymyResolution/tree/main/dataset_MR.

Table 1
The comprehensive comparative statistics of the advantages (pros) and disadvantages (cons) of these mainstream baselines used for the metonymy resolution task.

Learning categories	Models	data quality dependency	training efficiency	long-term dependencies	parameter magnitude	Noise robustness	model interpretability	performance ceiling
Statistical Machine Learning	*SVM+Wikipedia*	medium	fast	severe	fewer	poor	yes	inferior
	GYDER	medium	fast	severe	fewer	poor	yes	inferior
	PreWin	medium	fast	severe	fewer	poor	yes	inferior
RNN-based	*BiLSTM (GLoVE)*	high	relatively fast	moderate	medium	relatively robust	no	mediocre
	BiLSTM (ELMo)	high	relatively fast	moderate	large	relatively robust	no	relatively excellent
	BiLSTM (ELMo) +NER+POS	high	relatively fast	moderate	large	relatively robust	no	relatively excellent
PLM-based	*TWM-BERT*	high	slow	mild	extra-large	robust	no	excellent
	EBGCN & EBAGCN	high	slow	mild	extra-large	robust	no	excellent

5.3. Evaluation metrics

Accuracy and F1-scores are the most widely adopted evaluation metrics for label classification tasks in NLP. Accuracy refers to the proportion of correctly predicted samples among all predicted samples. We calculated accuracy as $Accuracy = Count_T/Count_N$, where $Count_T$ represents the correctly predicted samples and $Count_N$ represents the total number of samples evaluated.

Presented in Eq. (9), F1-scores are a more comprehensive metric that takes both precision ($Prec$) and recall (Rec) into account. We calculated them using the equation:

$$F1 = \frac{2 * Prec * Rec}{Prec + Rec} \quad where \quad Prec = \frac{TP}{TP + FP} \quad \& \quad Rec = \frac{TP}{TP + FN} \tag{9}$$

where the $Prec$ and Rec denote the precision and recall rate of the classification results. TP (true positives) represent the number of samples whose actual and predicted class are both yes; FP (false positives) represent the number of samples whose actual class is no but are predicted as yes; and FN (false negatives) represent the number of samples whose actual class is yes but are predicted as no.

Note that we have reported the F1-scores for literal and metonymic categories separately, which are referred as F1-L and F1-M, respectively.

5.4. Experimental settings

This section details the hyperparameter settings for main scenarios tested, i.e., both the full-data supervised learning and the few-shot learning experiments.

5.4.1. Full-data supervised settings

We conducted the experiments in a hardware environment running the *Windows 10 Pro* with one GTX 1080Ti GPU. For the software environment, the code was implemented in Python 3.7 and PyTorch 1.9.0 with Hugging Face Transformers 4.23.1.[8] The maximum length of all input sentences was constrained to 256.

For simplicity, Table 2 lists the optimal hyperparameters for training each of PromptMR variants. In this table, LRt, LRv, and LR_p represent the learning rates of the template's learnable parameters, the verbalizer's learnable parameters, and the pre-trained language models (PLMs), respectively. *MaxLen* restricts the maximum sequence length of the prompt input. The number of epochs *Epoch* indicated within the parentheses apply to the few-shot learning settings, while the number of epochs *Epoch* outside the parentheses represent the epochs required for the routine setting of full-data supervised learning. The batch size for the supervised learning scenario with full data was set to 16, and to 4 in the few-shot learning scenarios.

Please note that there is an imbalance in the sample distribution within the SemEval and CoNLL benchmarks. Specifically, the training samples for the metonymy category in SemEval and CoNLL account for only 20% to 30% of the training samples for the literal category. This poses a challenge to the availability and quality of this dataset, making it one of the limiting bottlenecks for improving model performance. To address the issue of imbalanced distribution of category samples, we employ a mainstream and effective model training strategy called *Resampling*. This strategy involves using oversampling or under-sampling techniques to balance the number of samples in each class. Oversampling methods replicate minority class samples or generate synthetic samples, while under-sampling methods remove majority class samples or reduce their quantity. This approach helps achieve a more balanced distribution of samples across different classes, facilitating better learning of the minority class by the model. It alleviates the problem of under-fitting the features of the minority class due to its smaller size, while also mitigating the problem of over-fitting the features of the majority class. Specifically, when dealing with imbalanced data, we first calculate the sample counts for the minority and majority classes in the original training set, obtaining an approximate ratio of samples across different classes. Then, we use oversampling to resample the minority class by replicating or generating synthetic samples to match the quantity of the majority class samples. Next, we employ under-sampling to randomly delete samples from the majority class, reducing its quantity to match that of the minority class. Finally, the resampled dataset was used for subsequent model training.

Further, it is widely acknowledged that single experiments in metonymy resolution can exhibit random fluctuations, which, if trusted, would lead to a solution that is neither stable nor robust. Therefore, to mitigate such experimental perturbations, we performed five experiments using the same settings for each variant of PromptMR. The average performances on the test set in terms of accuracy and F1-scores have been reported to provide a more reliable account of PromptMR's performance.

5.4.2. Few-shot settings

As one final note, we anticipated that the prompt learning paradigm would offer several advantages in low-data scenarios. Hence, our main focus was on conducting experiments in the

[8] Hugging Face Transformers toolkit: https://github.com/huggingface/transformers.

Table 2
The optimal hyperparameters for training time of the different PromptMR variants: PromptMR-base — basic prompt-tuning model for metonymy resolution; PromptMR-CTC — enhanced prompt-tuning model with continuous template construction; and PromptMR-CAS — enhanced prompt-tuning model with continuous answer search.

Model	PLM	Epoch	MaxLen	LR$_t$	LR$_p$	LR$_v$	Batch size	Optimizer
PromptMR-base	T5-base [29]	10 (20)	128	–	1e−4	–	16 (4)	AdamW
PromptMR-CTC	T5-base [29]	10 (20)	256	1e−4	1e−4	–	16 (4)	AdamW
PromptMR-CAS	T5-base [29]	20 (20)	256	–	1e−4	3e−5	16 (4)	AdamW

Table 3
Experimental comparisons of PromptMR-base against other state-of-the-art methods, with all numbers measured in percentage (%) and the results averaged over five fully-trained runs. The accuracy metric Acc (Std) is denoted as '*Acc*' alongside the standard deviation '*Std*'. F1-L and F1-M denote the F1-scores for the literal and metonymic categories, respectively. Results in **bold** represent the best result, while the second-best results are underlined.

Model	ReLocaR			SemEval			CoNLL		
	Acc(Std)	F1-M	F1-L	Acc(Std)	F1-M	F1-L	Acc(Std)	F1-M	F1-L
SVM+Wikipedia	–	–	–	86.2(N/A)	59.1	91.6	–	–	-
GYDER	85.2(0.48)	55.0	91.17	–	–	–	82.7(0.54)	64.7	87.3
PreWin	83.6(0.71)	84.8	84.4	83.1 (0.64)	57.3	90.6	–	–	-
BiLSTM(GLoVE)	82.9(0.85)	83.0	82.9	75.4(1.72)	37.4	83.2	80.48(0.54)	–	-
BiLSTM(GLoVE) +NER+POS	84.2(0.69)	84.2	84.2	82.0(1.36)	37.7	88.8	85.23(0.51)	–	-
BiLSTM(ELMo)	90.0(0.40)	90.1	90.0	86.3(0.45)	54.7	91.9	88.4(0.47)	–	-
BiLSTM(ELMo) +NER+POS	90.1(0.36)	90.1	90.1	86.1(0.47)	55.6	91.6	89.1(0.32)	–	-
TWM-BERT	94.4(0.31)	–	–	88.2(0.61)	–	–	91.2(0.40)	–	-
EBGCN	95.5(0.46)	95.5	95.5	89.1(0.60)	67.5	93.5	92.9(0.54)	89.7	94.2
EBAGCN	95.7(0.34)	95.7	95.7	89.8(0.85)	68.3	**94.0**	93.3(0.29)	90.6	94.1
Prompt-base (Our Work)	**96.1(0.32)**	**96.2**	**96.1**	**90.5(0.67)**	**69.2**	93.9	**94.1(0.27)**	**91.4**	**95.3**

few-shot settings. However, as there is no consensus on few-shot learning methods, we designed our few-shot experiments to follow the practical experiences of previous works [42,43].

Specifically, we conduct the few-shot experiments to evaluate each variant within PromptMR under 5-shot, 10-shot, 20-shot, and 50-shot settings. Noting that each '*N*-shot' denotes the number of instances for each label in the training set rather than the sum of the training instances. To ensure the count of metonymic instances equaled the count of literal instances, we randomly selected the appropriate number of samples from the training set. It is important to emphasize that each category has an equal number of samples in a few-shot learning environment, so the problem of imbalanced data distribution can be disregarded. Therefore, there is no need to employ other sample balancing strategies such as resampling, class weight adjustment, data augmentation, or ensemble learning.

5.5. Experimental results

We compare PromptMR with several baselines on both resource-rich settings and few-shot settings. Section 5.5.1 of this section provides the performance statistics for PromptMR-base in comparison to the baselines in the full-data learning scenario. The results between PromptMR and baselines for the few-shot learning scenario are given in Section 5.5.2; and the overall performance comparisons between each of PromptMR variants are provided in Section 5.5.3.

5.5.1. Full data learning

Using the optimal hyperparameters outlined in Section 5.4, we performed a series of experiments to assess the efficacy of PromptMR-base in a scenario with sufficient data resources to build a high-quality model.

Table 3 provides the full-data training performance comparisons with representative state-of-the-art baselines. Most of the results in this table were taken from the original papers, including GYDER, PreWin, TWM-BERT and EBAGCN on *ReLocaR* and *SemEval*. However, the results reported for the *CoNLL* benchmark were evaluated using reproductions of the original experimental code.[9]

Based on these experimental results, we clearly see that even our basic prompt learning approach, PromptMR-base, outperforms the state-of-the-art baselines in terms of the comparisons of all three metrics. This firmly demonstrates that prompt learning is a superior solution to metonymy resolution over the techniques that rely on fine-tuning PLMs. More specifically, PromptMR-base makes a significant improvement over the traditional machine learning methods, *SVM+Wikipedia* and *GYDER*. There was an absolute increase in accuracy of approximately 4.3% over *SVM+Wikipedia* and a 10.1% improvement in Metonymic F1-score on the SemEval benchmark. Against *GYDER*, there was a approximately 10.9% increase in accuracy and a 41.2% improvement in Metonymic F1-score, again in the metonymic category, with the ReLocaR benchmark.

Moreover, as shown in the third column of Table 3, the model comparisons on the SemEval benchmark, we discovered a fascinating phenomenon that sheds new light on the characteristics of datasets. The overall performances of all baselines tested on the SemEval benchmark are not good, which we hypothesize is due to the imbalance label distributions of the annotation schemes. Especially, the statistical comparisons between PromptMR-bases and EBAGCN show that it is difficult to achieve further improvement on this dataset, as evidenced by the last two rows of

[9] Note that we were not always able to reproduce the experimental results as per the original work, so there may be a little bit slight performance discrepancies.

Table 3. This finding has provided us the implications towards our conjecture that SemEval is more one-dimensional than the other datasets, making it hard to generalize, even with the additional training data.

Disregard the issues illustrated above, PromptMR-base significantly outperforms deep learning methods of almost all the metrics, including accuracy, Literal F1-score (F1-L) and Metonymic F1-score (F1-M). Examples of such methods include the pre-trained embedding-based RNN method, BiLSTM (GLoVE) [39], and the PLM-based fine-tuning methods: BiLSTM(ELMo), TWM-BERT, and EABGCN [1,2,39].

From the results, we observed several other examples of where PromptMR-base outshone its competitors:

1. Compared to BiLSTM(GLoVE)+NER+POS, which is a pre-trained approach based on embeddings from an RNN, PromptMR-base delivered an average increase in accuracy of 6.1% on ReLocaR and 8.8% on CoNLL. Moreover, it also outperformed this comparator in terms of F1-scores in both the literal and metonymic categories;
2. Compared to the PLM-based fine-tuning methods, PromptMR-base showed a significant performance increase across all three benchmarks with respective gains of 5.0%, 4.4%, and 4.0% in absolute accuracy over BiLSTM(ELMo)+ NER+POS on the three benchmarks.

From the last three rows, it is indeed impressive to see that PromptMR-base outperformed the state-of-the-art method EBAGCN proposed by Du et al. [1] with all three datasets. Note that EBAGCN's major contribution is that it integrates the PLM-based fine-tuning method with external linguistic knowledge and syntactic dependency trees.[10] This undoubtedly confirms that the prompt learning paradigm is very effective solution to metonymy resolution that can provide competitive performance even without incorporating external knowledge like syntactic dependency trees. Notably, PromptMR-base is trained solely on the given training set, and does not introduce any external knowledge as an auxiliary signal. We argue that this is a significant advantage, as external knowledge sources can be costly to obtain and may not be available in all languages or domains.

In summary, the prompt learning paradigm, which makes the most of the internal knowledge inside in the PLMs, is the most crucial factor in producing such remarkable results with a metonymy resolution task. However, it is worth noting that incorporating external knowledge might still be beneficial in some scenarios, especially when dealing with domain-specific or low-resource languages.

In order to gain a deeper understanding of the specific performance details of our PromptMR-base method on the dataset, we additionally calculated the precision and recall corresponding to the literal F1-score and metonymic F1-score.[11] It is important to note that, in addition to the baseline performance comparison experiments mentioned above, we conducted ten of evaluation experiments on the PromptMR-base model using three datasets, and the statistical results are recorded in Table 4.

In Table 4, we observed that the average value of literal F1-score (F1-L) is higher than the average value of metonymic F1-score (F1-M) across all datasets, indicating that handling literal meaning is easier than handling metonymic meaning using this method. On the ReLocaR dataset, the average metonymic precision (Prec-M) and average metonymic recall (Rec-M) are very close, and their impact on F1-M is also similar. This suggests a relatively balanced influence of precision and recall on F1-M on this dataset. On the SemEval dataset, the average Prec-M is relatively low, while the average Rec-M is even lower. This results in a lower F1-M. In contrast, the average literal precision (Prec-L) and average literal recall (Rec-L) for F1-L are both high, leading to a higher F1-L score. This indicates that precision has a greater impact on metonymic F1-score, while recall has a greater impact on literal F1-score on the SemEval dataset. On the CoNLL dataset, the average Prec-M and average Rec-M are close, and their impact on F1-M is also similar. Similarly, the average Prec-L and average Rec-L for average F1-L are also close and have a similar impact on the final model performance. This suggests a relatively balanced influence of precision and recall on both metonymic F1-score and literal F1-score on the CoNLL dataset.

It is of significance to observe that the performance of the PromptMR-base method varies in terms of literal and metonymic meanings across different datasets. In some datasets, precision has a greater impact on metonymic F1-score, while in other datasets, precision has a greater impact on literal F1-score. These differences may be attributed to the characteristics and domains of the datasets, and further research and exploration are needed to understand the underlying reasons for these influences.

In conclusion, due to the generic nature of PLM-based fine-tuning paradigms, existing fine-tuning NLP solutions discard the PLM's decoder and adopt a newly-introduced fine-tunable adaption module with external parameters. This leads to significant parameter weight discrepancies between the pre-training and fine-tuning stages. Additionally, the general fine-tuning paradigm's dependency on sufficient and high-quality training data resources becomes a performance bottleneck for the metonymy resolution linguistic task. Compared to these mainstream PLM fine-tuning methods, Prompt Learning (Prompt-Tuning) is a viable approach that has been widely applied to NLP sub-field tasks, and there are some essential differences between them. (1). The prompt learning abandon the downstream classifier to predict the conditional token probabilities to obtain final labels. (2). The prompt-based methods leverage specific vocabulary conditional generative distributions to recover the downstream labels. (3) Prompt-tuning try to keep the whole Transformer architecture of PLM to ensure that the prior knowledge in Transformer is fully utilized. Through both theoretical validation and practical application, the prompt learning paradigm has demonstrated notable superiority over the mainstream fine-tuning paradigm for PLMs.

5.5.2. Few-shot learning

In our next set of experiments, we aimed to evaluate PromptMR-base's ability to handle scenarios with limited data. Hence, we evaluated its performance in a range of few-shot settings, i.e., {5, 10, 20, and 50-shot} settings, and compared its performance to our selected state-of-the-art methods. The statistical results are given in Table 5.

Given our limited experimental capacity, we opted not to consider the methods proven to be inferior in the previous experiments. This included SVM+Wikipedia, GYDER, and BiLSTM(ELMo) [8,38,39]. Rather, we selected two representative state-of-the-art methods as the baselines to experiment with in these few-shot comparisons: TWM-BERT [2] and EBAGCN [1]. This trade-off between performance demonstration and practical constraints allowed us to conduct a more thorough evaluation. TWM-BERT can be thought of as a PLM-based fine-tuning method without external knowledge, while EBAGCN is a PLM-based fine-tuning

[10] Dependency parsing is conducted by the Stanford CoreNLP — the natural language processing toolkit proposed by Christopher et al. (Stanford CoreNLP) https://stanfordnlp.github.io/CoreNLP/.

[11] We have released the corresponding experimental training logs at our public GitHub: https://github.com/albert-jin/PromptTuning2MetonymyResolution/tree/main/results_MR/prompt_base

Table 4
The statistical results of the average performance conducted on PromptMR-base from 10 independent evaluation experiments, including the averaged precision (*Prec-M, Prec-L*), recall (*Rec-M, Rec-L*), and F1-score with standard deviation (*F1-M (Std), F1-L (Std)*) for literal and metonymic aspects, across the three adopted datasets.

Dataset	Prec-M (*avg*)	Rec-M (*avg*)	F1-M (±*Std*)	Prec-L (*avg*)	Rec-L (*avg*)	F1-L (±*Std*)
ReLocaR	96.24	97.29	96.72 (±0.14)	97.12	96.18	96.69 (±0.22)
SemEval	72.57	64.73	68.43 (±0.47)	92.07	94.35	93.26 (±0.34)
CoNLL	92.13	91.08	91.66 (±0.32)	95.19	95.78	95.51 (±0.19)

Table 5
Few-shot comparisons of PromptMR-base with two other state-of-the-art methods, averaged over 5 runs. Acc (Std) denotes the accuracy metric with its standard deviation. F1-L and F1-M respectively denote the literal and metonymic F1-scores. All numbers are presented in percentages (%). Values in **bold** represent the best result for each dataset, while underlined numbers represent the second-best result.

Few-shot setting	Model	ReLocaR			SemEval			CoNLL		
		Acc(Std)	F1-M	F1-L	Acc(Std)	F1-M	F1-L	Acc(Std)	F1-M	F1-L
5-shot	TWM-BERT	59.9(12.2)	66.2	54.6	53.2(10.6)	52.6	56.0	54.8(10.1)	54.3	56.8
	EBAGCN	61.2(9.3)	69.8	56.4	53.7(9.4)	54.1	55.8	53.4(8.1)	55.7	54.8
	PromptMR-base	**68.4(9.8)**	**72.9**	**63.6**	**57.2(10.7)**	**56.2**	**62.5**	**60.3(9.6)**	**59.7**	**57.6**
10-shot	TWM-BERT	72.4(10.1)	71.3	57.6	58.6(9.1)	56.1	72.8	61.5(9.1)	59.1	66.6
	EBAGCN	73.1(11.5)	72.7	60.3	59.9(7.2)	54.2	75.8	64.2(8.2)	61.8	69.2
	PromptMR-base	**77.2(9.5)**	**79.8**	**76.8**	**62.8(8.2)**	**57.9**	**81.8**	**68.3(8.2)**	**63.3**	**72.1**
20-shot	TWM-BERT	79.3(7.8)	77.5	80.9	64.7(7.4)	59.3	74.3	69.3(6.24)	65.9	73.4
	EBAGCN	80.2(8.1)	77.2	84.0	65.3(8.5)	56.5	77.8	71.7(4.2)	67.2	77.4
	PromptMR-base	**85.9(5.7)**	**85.7**	**87.4**	**68.9(5.3)**	**62.3**	**80.8**	**76.3(3.6)**	**68.7**	**81.5**
50-shot	TWM-BERT	83.4(6.2)	82.6	84.1	72.7(6.2)	61.7	77.9	74.4(3.5)	69.2	78.1
	EBAGCN	84.1(5.6)	82.9	85.3	75.4(5.9)	59.1	82.0	76.9(4.7)	70.7	81.3
	PromptMR-base	**90.4(3.9)**	**89.5**	**90.9**	**82.9(5.6)**	**67.7**	**87.2**	**82.8(3.4)**	**78.5**	**86.4**

method enhanced by external knowledge, such as syntactic dependency tree. To our best knowledge, EBAGCN is currently the most accurate method of metonymy resolution.

It is important to emphasize that, in few-shot learning settings, the number of training samples involved in each epoch of model training optimization is significantly smaller than the volume of effective training samples in full supervised data. For instance, in a 20-shot training scenario, after 20 training epochs with a learning rate binding of 10^{-4}, only 400 samples are optimized. From a quantitative analysis perspective, such a quantity is far from being equivalent to a full training epoch. Compared to regular training, the number of samples involved is very limited. For example, the SemEval and ReLocaR datasets consist of approximately 1000 instances, while the CoNLL dataset is even larger with over 7000 instances. To avoid PromptMR from being under-optimized due to excessively low learning rates or insufficient training iterations, we deliberately increased the number of training epochs in few-shot learning and set the batch size to 4 (corresponding to 16 in full data supervised learning). It is important to emphasize that the fine-tuning of the entire few-shot parameter configuration was conducted by keeping other parameters fixed and modifying specific hyperparameters through grid-based optimization. The experiment underwent multiple rounds of parameter tuning to achieve the best performance of the model. Experimental results have shown that this approach provides a more appropriate and efficient optimization strategy for the model. Both theoretical considerations and empirical evidence indicate that, without significant occurrences of over-fitting, it allows for the thorough acquisition of semantic knowledge within these few limited training samples.

From the results, we can find that:

1. Across all three benchmarks and all few-shot settings, PromptMR-base consistently outperformed the two state-of-the-art baselines by a significant margin. In the 5-shot setting, PromptMR-base yielded an approximately 8.5%, 5.0%, and 5.5% improvements in accuracy over TWM-BERT on ReLocaR, SemEval, and CoNLL, respectively. In the 20-shot setting, PromptMR-base showed an average improvement of 6.6%, 4.2%, and 7.0% in accuracy over EBAGCN on ReLocaR, SemEval, and CoNLL, respectively.
2. PromptMR-base only exhibits a fewer minor deviations in most of the experiments, which indicates that it is highly stable during training despite being subject to a range of different mini few-shot training samples. We speculate that this is partly due to the fact that the internal knowledge within PLMs is effectively utilized, and thus it better withstands a certain degree of noise and volatility in the experiments.
3. In the 50-shot setting, PromptMR-base shows clear advantages, achieving an average absolute accuracy of 91.4% with the *ReLocaR* benchmark — (91.5% F1-score for the metonymic category and 90.9% F1-score for the literal category). Compared to TWM-BERT and EBAGCN, PromptMR-base shows a significant margin in performance gains with an average of 5% to 10% in absolute gains across all three metrics on all benchmarks under 50-shot setting experiments. Further, looking at the last row in Table 3, these results come very close to results of experiments with the full training data. Nominally, there is less than a 10% performance gap over the three metrics across the three benchmarks.

It is worth noting that, in the 50-shot setting, only 50 samples of the two categories (i.e. metonymic and literal) are provided to train the metonymy resolution model. This practical sample count is very much smaller than that of the full-data training setting, where the model might be trained on several thousand training samples.

We have identified key factors contributing to the superior performance of PromptMR over other baselines in both data-sufficient and data-scarce scenarios. The superior performance of PromptMR in metonymy resolution is attributed to its diversified pre-training, optimized prompt engineering, targeted guidance, and effective utilization of PLM's prior knowledge, and these aspects are listed as follows:

1. Diversified pre-training and optimized prompt engineering: PromptMR utilizes diverse pre-training strategies and optimized prompt templates. This approach improves the model's understanding of various domains and guides it in generating accurate responses.
2. Adjustable targeted guidance: PromptMR provides predetermined questions or instructions to guide the model's output. This targeted guidance improves the accuracy and relevance of the generated answers by focusing on specific domains or tasks.
3. Effective utilization of PLM's prior knowledge: PromptMR effectively leverages the prior knowledge embedded within PLMs, surpassing the limitations of fine-tuning and achieving impressive performance even with limited training samples.

PromptMR produced state-of-the-art results with *ReLocaR*, *SemEval*, and *CoNLL* given a full-data training paradigm. Additionally, through exhaustive comparative experiments, PromptMR was shown to outperform the other state-of-the-art baselines in all the few-shot settings by a significant margin. Thus, these overall results and our analysis firmly demonstrate that prompt learning is far superior to the current PLM-based fine-tuning methods in both full-data and data-scarce scenarios.

In summary, experiments with PromptMR, a prompt learning-based method for metonymy resolution, show its effectiveness in both full-data and few-shot learning scenarios. Base on our analysis, we think that the crucial factor is that prompt learning involves using fixed prompts to specify a pre-trained language model's behavior, restructuring downstream tasks to resemble original language model training, enabling better knowledge transfer and leveraging limited sample data effectively. This approach also helps focus the model's attention on task-relevant information and addresses issues of neglecting crucial information in low-resource scenarios. Prompts play a vital role in guiding the model with limited training data. Additionally, prompt template or answer search engineering further enhances the model's efficiency, particularly in few-shot data-scarce scenarios, potentially leading to significant improvements.

In our next experiments, we aimed to further optimize the template engineering and answer engineering schemes to better exploit the advantages of prompt learning.

5.5.3. PromptMR comparisons

So far, we have reported the performance statistics of the comparisons between the three enhanced variants of PromptMR together with PromptMR-base in Table 6. Note though that, due to space limitations, only the accuracy of the results are reported; the deviation '*Std*' was omitted. (F1-L %) and (F1-M %) denote the results of literal and metonymic F1-scores, respectively. Again, the numbers in **boldface** indicate the best results, while values in underlined indicate second-best results for each dataset, respectively.

From Table 6, we made several observations.

1. Competitive performances over current approaches Compared with the performance results of the baselines shown in Table 3, each PromptMR variant in the full-data learning delivered performance superior to that of the comparative baselines. In other words, all of the variants tested consistently achieved state-of-the-art results over the other PLM-based fine-tuning approaches. Thus, PromptMR represents a new state-of-the-art benchmark for handling metonymy resolution tasks.

2. Excellent performances in few-shot learning As shown in the few-shot learning experiments, two PromptMR variants (i.e., PromptMR-CAS and CTC-PromptMR-CAS) consistently delivered better results than PromptMR-base, especially in the 5-shot and 10-shot scenarios. However, the continuous learnable template used in PromptMR-CTC did not seem to bring significant improvements in these scenarios. This is possibly because a hand-crafted template does not sufficiently capture the semantics of the task. These observations suggest that models based on prompt learning, especially PromptMR-CAS and CTC-PromptMR-CAS, are more effective at handling metonymy resolution in few-shot scenarios. Further, we can see that PromptMR-CAS, which comes equipped with learnable soft answers, shows very promising performance improvements. This indicates that the two strategies for handling low-resource metonymy resolution tasks might have some complementarity. Based on the aforementioned observations, we can conclude that while CTC-PromptMR-CAS may not outperform PromptMR-base and PromptMR-CAS in full-data training settings, this is the optimal enhanced PromptMR variant for performing metonymy resolution in few-shot learning settings. Overall, we find these results highlight the effectiveness and flexibility of prompt learning-based approaches for handling metonymy resolution tasks in both full-data and few-shot settings.

Additionally, based on the experimental analysis above, we gain some valuable insights into the contributions of these newly introduced strategies. Firstly, in the continuous template construction strategy, continuous templates are trained through backpropagation, utilizing the continuous representation of the pretrained language model's output to construct templates. Through iterative optimization of the template construction process, we achieve improved model performance. It is presumed that this continuous template construction strategy allows for better adaptation to different tasks and domains, thereby enhancing metonymy resolution performance. Next, the continuous answer search strategy replaces the manual discrete answer mapping used in PromptMR-base with a learnable soft verbalizer. In the traditional discrete answer engineering, PromptMR-base heavily relies on manually defining candidate answers, which are subsequently mapped to final labels. This manual process not only consumes a significant amount of time but also limits the transferability of the approach. It is hypothesized that the continuous answer search approach optimizes answer engineering by enabling continuous searching for answers, resulting in enhanced performance and transferability.

In conclusion, by incorporating the continuous template construction and continuous answer search strategies, our models exhibit improved adaptability to different application scenarios, surpassing the limitations of discrete templates and manual answer mapping. The continuous template construction allows the model to automatically learn more adaptive templates, enhancing flexibility and adaptability. Simultaneously, continuous answer search improves answer engineering by utilizing a learnable soft verbalizer, making the process more intelligent and efficient, thus enhancing both model performance and transferability. The combination of continuous optimizable templates and

Table 6
Performance statistics for each of the PromptMR variants in the few-shot and full-data supervised learning settings. All results were averaged over 5 runs to ensure experimental reliability and authenticity.

Setting	Model	ReLocaR			SemEval			CoNLL		
		Acc	F1-M	F1-L	Acc	F1-M	F1-L	Acc	F1-M	F1-L
5 - shot	PromptMR-base	68.4	72.9	63.6	57.2	**56.2**	62.5	60.3	**59.7**	57.6
	PromptMR-CTC	70.4	73.5	61.7	**66.4**	31.8	**78.9**	52.4	54.8	50.7
	PromptMR-CAS	**75.7**	**78.3**	**72.4**	65.4	40.0	75.7	**67.4**	56.3	**73.8**
	CTC-PromptMR-CAS	72.3	74.8	66.7	66.1	30.5	77.6	67.1	58.4	70.3
10 - shot	PromptMR-base	77.2	79.8	76.8	62.8	**57.9**	81.8	68.3	63.3	72.1
	PromptMR-CTC	78.7	79.1	77.9	53.1	27.1	65.3	70.1	58.2	77.9
	PromptMR-CAS	**82.9**	**82.1**	**83.6**	74.2	44.6	83.2	68.9	54.1	78.8
	CTC-PromptMR-CAS	78.9	80.3	77.9	**77.7**	43.1	**86.1**	**74.4**	**65.2**	**79.7**
20 - shot	PromptMR-base	85.9	85.7	**87.4**	68.9	**62.3**	80.8	76.3	68.7	81.5
	PromptMR-CTC	87.1	85.9	87.2	72.3	45.8	81.4	75.6	65.5	83.2
	PromptMR-CAS	**87.4**	**87.9**	86.8	**77.4**	42.1	**85.9**	**78.9**	**70.9**	**84.3**
	CTC-PromptMR-CAS	84.2	84.3	84.1	69.1	39.4	83.2	76.1	68.3	76.3
50 - shot	PromptMR-base	**90.4**	**89.5**	**90.9**	82.9	67.7	87.2	82.8	78.5	86.4
	PromptMR-CTC	90.1	89.3	88.9	73.8	52.7	81.9	**85.6**	**80.3**	88.1
	PromptMR-CAS	88.2	87.9	88.5	**87.9**	**69.3**	85.4	85.4	79.9	**88.4**
	CTC-PromptMR-CAS	89.1	88.9	89.2	85.7	52.1	**91.6**	84.1	74.2	87.9
Full - data Training	PromptMR-base	96.1	96.2	96.1	**90.5**	69.2	**93.9**	94.1	91.4	95.3
	PromptMR-CTC	96.3	96.4	96.2	89.7	68.3	93.4	**94.3**	**91.6**	**95.8**
	PromptMR-CAS	**96.5**	**96.6**	**96.4**	90.1	69.7	92.6	94.1	91.2	95.4
	CTC-PromptMR-CAS	95.4	95.3	95.9	89.6	**70.4**	93.2	93.8	90.2	94.8

soft answer search in PromptMR has demonstrated its effectiveness in metonymy resolution. The continued answer research in refining these techniques, exploring different search strategies, addressing transfer learning challenges, and incorporating multimodal approaches holds great promise for further advancements in metonymy resolution.

6. Ablation study

Continuing from the concise introduction of the basic implementation of PromptMR in Section 4.1, we further delve into the analysis of different templates composed of selected prompt words and label words to explore the impact of these crucial factors on the final recognition performance. Subsequently, we conducted research on the impact of different pre-trained language models on the capability to determine metaphoricity based on prompt learning. Thus, in this section, extensive ablation experiments were conducted to optimize PromptMR, and corresponding experimental results were discussed to further investigate the significant factors when applying prompt learning techniques to metonymy resolution.

6.1. Ablation analysis of template selection

Note that the quality of manually selected templates plays an important role in determining their performance for particular prompting tasks, including metonymy resolution, as validated by previous research [9,12,36]. Therefore, it is crucial to search for the optimal template for our PromptMR, particularly for PromptMR-base. In this regard, we investigated the influence of manual templates using the ReLocaR benchmark, which has balanced distributions of literal and metonymic entities. Notably, these ablation studies were conducted under full-data supervised scenarios to ensure training robustness.

Concretely, we manually designed several practical discrete templates, which are displayed in Table 7. Among these templates, the **E5** template, which expresses a simple conceptual subordination relationship, was adopted as the baseline. The narratives and semantics of the remaining four templates (**A1** to **D4**) are more coherent and understandable, conforming to everyday natural linguistic expressions.

In the manual template column of Table 7, [SENT] represents the input slot for the original sentence for metonymy classification, while [TAR-WORDS] denotes the input slot for a concept that exists in the sentence. Additionally, [MASK] is a specific word generation position reserved for the PLMs.

As shown in the table, the overall ablation results suggest that the choice of templates significantly affects the performance of PromptMR-base in metonymy resolution. Further analysis of the experiments revealed that templates such as Template **A1** and **B2** consistently performed better than others (i.e., **C3** and **D4**), while Template **E5** showed the lowest performance on the ReLocR dataset, with a significant decrease in accuracy (−0.9% and −0.5%) compared to PromptMR-base, which adopts **A1** and **B2**. These findings highlight the importance of careful prompt template selection and customization in achieving optimal performance in prompt learning models. Further research could explore the factors contributing to the differential performance of prompt templates in the metonymy resolution task and identify ways to improve their effectiveness.

6.2. Ablation analysis of label words selection

Previous prompt-tuning research has noted that the quality of selected label words may influence the model performance in specific application conditions [9,36]. To understand the actual effects of label word selection when applying prompt learning to metonymy resolution, we investigated the influence of four different label words on the performance of the PromptMR-base

Table 7
Performances of PromptMR-base that adopts different manual template validated on the development set of ReLocaR.

ID	Manual Templates	% Acc.(Gain)
A1	[SENT] So, [TAR-WORDS] is a [MASK] entity.	**96.1 (+0.9)**
B2	[SENT],in which [TAR-WORDS] is a [MASK].	95.7 (+0.5)
C3	[SENT], [TAR-WORDS] is represented as [MASK].	95.4 (+0.2)
D4	[TAR-WORDS] is a [MASK] entity in [SENT].	95.4 (+0.2)
E5(baseline)	[SENT], [TAR-WORDS], and [MASK].	95.2 (+0.0)

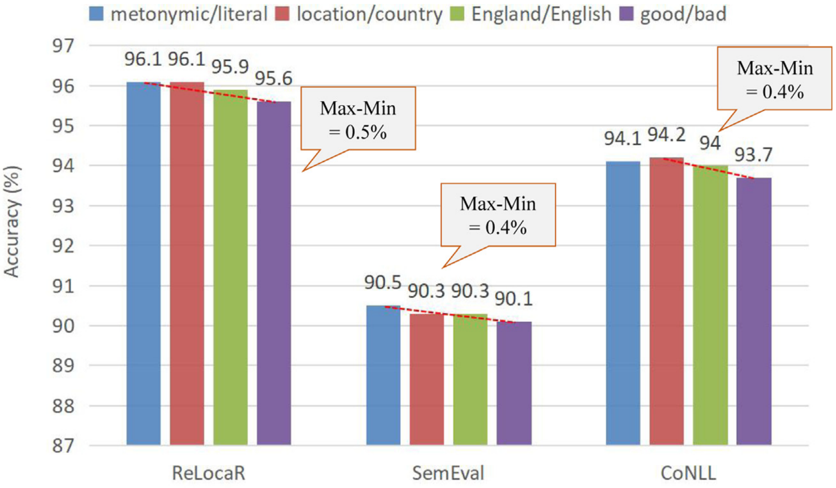


Fig. 7. Ablated evaluations (using % accuracy) of PromptMR-base which adopts different manual label words validated on three benchmarks.

model. We conducted extensive experiments on three benchmark datasets (i.e., ReLocaR, SemEval, and CoNLL).

Based on the experience and knowledge of linguistic experts,[12] we predefined four different categories of label words: {*metonymic/literal, location/country, England/English, good/bad*}. Specifically, we filtered the label words from three primary dimensions provided from their reviews: (1) linguistic concreteness and abstractness; (2) semantic cross-domain comparability; and (3) linguistic synonyms and antonyms.

Specifically, among them, metonymic and literal are words that directly describe the nature of the prompt, used for direct semantic judgment of metonymy. On the other hand, location and country represent a semi-abstract level of semantic representation, abstracted from metonymy and literal translation, by substituting the prompt words with different meanings. The selection of "England/English" as a metonymy prompt word pair is based on the recommendation of linguistic experts, distinguishing it from other prompt word choices such as "American" and "America", or "China" and "Chinese". The primary reason for choosing "England/English" is to highlight whether a more specific semantic representation can enhance metonymy prompt learning when compared to other prompt word choices. As for the word pair "good/bad", our intention is to explore whether it remains effective for PromptMR learning on the aspect of semantic differentiation, without considering the influence of metonymy. Extensive experimentation has demonstrated that, under thorough fine-tuning of the pre-trained language model, the choice of semantic words has a relatively stable impact on model performance. However, in the case of zero-shot or few-shot fine-tuning, the selection of prompt words has a more significant effect on the final model performance.

It is worth noting that, during the experiments, to minimize the impact of experimental random factors and ensure a fair comparison, all the results were reported using the average of five independent runs. Fig. 7 reported the qualitative evaluation results for PromptMR-base which adopted different label words as intermediate answers.

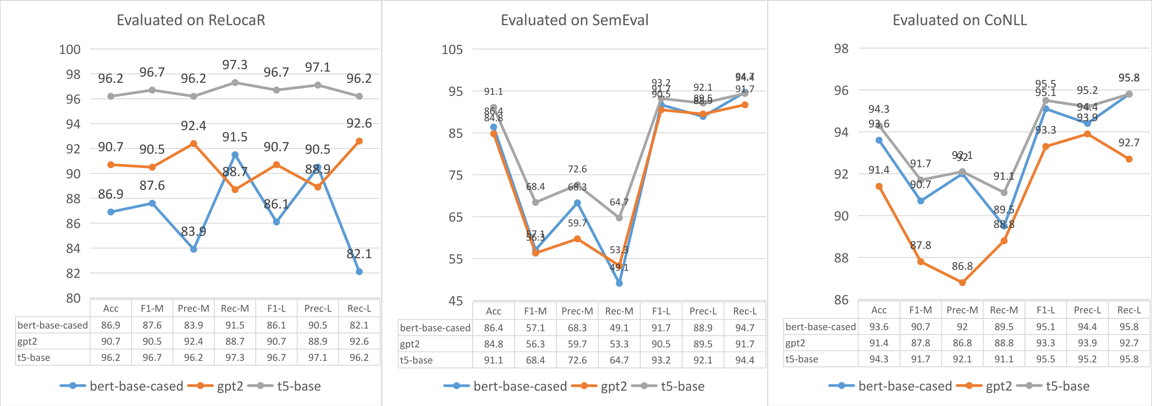
Based on the experimental results, we can see that there is only a slight overall performance gap between the selected label words. For instance, PromptMR-base, which adopted the label word combination "*metonymic/literal*", outperformed all other three implementations of PromptMR-base by no more than 0.5% in accuracy. Based on the corresponding statistics, the average performance gap in accuracy between any two different selections is generally within the range of 0.2% to 0.3%.

Above all, our findings show that there is only a negligible difference between the four model variants. In conclusion, this suggests that the learning of label words has a minimal effect on the model's performance in metonymy resolution. Therefore, this ablated experiments indicate that the choice strategy of label words is not a critical factor when designing prompt-based models for metonymy resolution tasks. Compared to the label word selections, when applying the prompt learning technique to metonymy resolution, the template selection strategies could be a more crucial step to better utilize the rich factual prior knowledge hidden in the PLMs.

6.3. Ablation analysis of the PLM selection

Within the framework of prompt learning, different pre-trained language models vary in terms of the quantity and quality of knowledge they inherently possess. Furthermore, their architectural differences contribute to the fact that the choice of pre-trained language model used becomes one of the key factors influencing the metonymy resolution performance of the

[12] We thank Prof. Yang from Imperial College London for the professional suggestions.



Evaluated on ReLocaR	Acc	F1-M	Prec-M	Rec-M	F1-L	Prec-L	Rec-L
bert-base-cased	86.9	87.6	83.9	91.5	86.1	90.5	82.1
gpt2	90.7	90.5	92.4	88.7	90.7	88.9	92.6
t5-base	96.2	96.7	96.2	97.3	96.7	97.1	96.2

Evaluated on SemEval	Acc	F1-M	Prec-M	Rec-M	F1-L	Prec-L	Rec-L
bert-base-cased	86.4	57.1	68.3	49.1	91.7	88.9	94.7
gpt2	84.8	56.3	59.7	53.3	90.5	89.5	91.7
t5-base	91.1	68.4	72.6	64.7	93.2	92.1	94.4

Evaluated on CoNLL	Acc	F1-M	Prec-M	Rec-M	F1-L	Prec-L	Rec-L
bert-base-cased	93.6	90.7	92	89.5	95.1	94.4	95.8
gpt2	91.4	87.8	86.8	88.8	93.3	93.9	92.7
t5-base	94.3	91.7	92.1	91.1	95.5	95.2	95.8

Fig. 8. Ablated experimental results (in percentage %) of PromptMR-base which adopts different PLMs (i.e. BERT, GPT2 and T5) that validated on three benchmarks. Sub-chart.1 (left), sub-chart.2 (center), and sub-chart.3 (right) display the evaluation results of five important metrics (i.e., average accuracy, metonymic/literal F1-scores, and metonymic/literal precision and recall) on the three benchmarks: ReLocaR, SemEval, and CoNLL.

prompt learning model. Hence, in this section, we further introduce two classic and high-performing pre-trained language models, Bidirectional Encoder Representations from Transformers (BERT) [25] and Generative Pre-trained Transformer version.2 (GPT-2) [30,44]. Through systematic evaluation based on the original PromptMR-base, we aim to demonstrate the differences and fluctuations in the performance of PromptMRs when adopting different pre-trained language models, thereby facilitating a more intuitive understanding of the model's strengths and weaknesses for the subsequent readers.

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model introduced by Google in 2018 [25]. Built on the Transformer architecture, BERT employs a bidirectional approach to contextual understanding, enabling it to grasp the semantic information in text effectively. Through unsupervised training on vast amounts of unlabeled data, BERT learns rich representations that can be fine-tuned for various downstream natural language processing tasks. Its contextual understanding makes it highly versatile for tasks such as sentiment analysis, named entity recognition, and question answering. GPT-2 (Generative Pre-trained Transformer version.2), developed by OpenAI [44], is an advanced pre-trained language model that builds upon the success of its predecessor, GPT [45]. It also utilizes the Transformer architecture. GPT-2 focuses on generating coherent and contextually relevant text by leveraging a large-scale unsupervised training process. With its impressive size and capacity, GPT-2 has demonstrated exceptional performance in natural language generation tasks, including language translation, text completion, and story generation. Initially, access to the full GPT-2 model was limited due to concerns about potential misuse.

In this ablation experiment, all the pre-trained language models we selected are from the pre-training versions available in the Transformers community of Hugging Face.[13] Specifically, we utilized the version "bert-base-cased" of BERT.[14] and the official standard version "gpt2" of GPT2,[15] both sourced from the Hugging Face model repository. When selecting the model versions, we took into account a comprehensive assessment of performance, training costs, and empirical reproducibility difficulty. Hugging Face's Transformers ecosystem is a comprehensive ecosystem that provides a wide range of pre-trained language models and tools for NLP tasks. The Transformers library by Hugging Face offers a collection of state-of-the-art pre-trained models, including BERT, GPT, RoBERTa, and many others. Hugging Face empowers developers and researchers with a rich set of tools and resources, making it easier to leverage pre-trained models and advance the state of the art in NLP.

Throughout this whole experiment, we took deliberate measures to ensure that the only influencing factor on the PromptMR-base's performance was the variation in language models. To achieve this, we carefully maintained consistency in other training hyperparameters and variables, including the learning rate and batch size. Also, the selection strategy of templates and label words followed the guidelines outlined in Section 4.1. The ablated experimental results were obtained by conducting five independent of full-data supervised training on three benchmarks. We conducted a systematic evaluation on three benchmarks, ReLocaR, SemEval, and CoNLL, and organized the results into three subline charts in Fig. 8. The models utilizing the PLMs T5, BERT, and GPT2 as their backbones were labeled as PromptMR$_{(t5)}$, PromptMR$_{(bert)}$, and PromptMR$_{(gpt2)}$, respectively. We computed the average performance for each PLM used, including various metrics (e.g. average accuracy, metonymic/literal F1-scores, and metonymic/literal precision and recall).

In the first subline chart, we observed that the PromptMR equipped with BERT performed the poorest, followed by PromptMR$_{(gpt2)}$, while PromptMR$_{(t5)}$ outperformed the other two models in all metrics significantly. On average, PromptMR$_{(t5)}$ surpassed PromptMR$_{(gpt2)}$ by approximately 5 percentage points and outperformed PromptMR$_{(bert)}$ by around 7 to 8 percentage points. Additionally, PromptMR$_{(t5)}$ exhibited greater stability in terms of fluctuation across all metrics compared to the other two models. Similarly, in the second and third subline charts, we visually observed that the PromptMR model utilizing T5 consistently outperformed the other models across all metrics. It is worth mentioning that both the SemEval and CoNLL datasets exhibited imbalanced distributions between metonymic and literal examples (metonymic accounting for only 20% to 30%). Therefore, we paid particular attention to the performance on these metonymic cases. On the SemEval and CoNLL, PromptMR$_{(t5)}$ significantly outperformed PromptMR$_{(bert)}$ and PromptMR$_{(gpt2)}$ in terms of metonymic F1-score, metonymic precision, and metonymic recall metrics. For instance, the F1-M metric showed an improvement of over 10% on the SemEval dataset and approximately 4%

[13] Hugging Face Transformers: https://huggingface.co/docs/transformers/v4.29.1/en/index.

[14] The basic version of BERT from Hugging Face: https://huggingface.co/bert-base-cased.

[15] The basic version of GPT2 from Hugging Face: https://huggingface.co/gpt2.

on the CoNLL dataset. Moreover, the metonymic precision and metonymic recall exhibited similar levels of improvement.

In conclusion, based on the above experimental analysis, it is evident that adopting the PLM T5 is a superior choice in terms of both metaphor recognition effectiveness and prediction stability. This experiment comprehensively evaluated the performance of the PromptMRs by utilizing different pre-trained language models as backbones. The results demonstrated that PromptMR$_{(t5)}$ which adopts the T5 model exhibited excellent performance across all metrics, particularly in metonymic cases. These findings provide valuable insights and guidance for further improving metaphor recognition.

7. Discussion

In this section, we analyze in detail the characteristics and advantages of PromptMR compared with the existing fine-tuning methods from a comprehensive and in-depth theoretical perspective. We do this by the following points:

- Theoretical analysis, which includes a comparison with existing deep-learning-based metonymy resolution methods;
- Research inspiration and practical implications for successive research of metonymy resolution.

7.1. Methodology comparisons

This subsection gives a methodology comparison between PromptMR and mainstream deep-learning based metonymy resolution solutions from a quantitative bird's-eye view.

1. To enhance the application of pre-training models in metonymy resolution, the fine-tuning paradigm necessitates a significant amount of monitoring data to fine-tuning the model parameters. Specifically, during pre-training, the model is trained using auto-regression and auto-encoding techniques, which differ significantly from the form of metonymy resolution and may limit the pre-training model's full potential. The characteristic of the fine-tuning paradigm that requires a significant amount of data to adapt to new task forms inevitably results in poor few-shot learning ability and susceptibility to over-fitting for these fine-tuning methods.
2. Compared to traditional fine-tuning methods that place more emphasis on the complex and sophisticated design of downstream classifiers for metonymy resolution, prompt-tuning techniques free researchers from the burden of designing intricate downstream classification networks. Instead, the fine-tuning paradigm concentrates on designing effective templates to better activate the prior knowledge in the PLM.
3. Since prompt-learning involves many details such as templating strategy, initializing strategy, and verbalizing strategy, etc. need to be considered. Compared to current fine-tuning methods, prompt learning has already been recognized as a complicated process, composed of hierarchical and complex steps. Thus, how to select ensemble-worthy prompts and appropriate label words to distill more knowledge from PLMs is also under-explored.
4. In addition, the learning rate of the PLM itself is generally kept consistent with the customized downstream classification network. However, most studies have found that adapting the prompt-tuning model using continuous trainable templates and verbalizers to different learning rate ratios has an impact on the optimal performance of metonymy resolution [12,16,17]. Therefore, for this class of prompt-tuning models, further exploration is needed to address the issue of better coordinating the optimization weights between the PLM's parameters and the trainable parameters of the downstream or upstream networks.
5. When considering transfer learning from metonymy resolution to different NLP tasks, mainstream fine-tuning methods inevitably involve modifications to the model architecture, such as transitioning from the binary classification problem of metonymy resolution to sentiment analysis with three classes. In contrast, prompt-tuning only requires customizing templates for the new domain task and defining corresponding label words for different categories. Therefore, compared to the fine-tuning approach, prompt-tuning has a more robust cross-domain transferability.

Given the nascent stage of this research field, we still lack a systematic understanding of the tradeoffs between these different training paradigms. The metonymy resolution task could benefit from systematic explorations, such as those performed in the pre-train and fine-tune paradigm, regarding the trade-offs between these different strategies. We leave this task for future work.

7.2. Research inspiration and practical implications

The innovations of our proposed PromptMR are illuminating and have broad applicability. To help readers understand our contributions better, we provide a detailed discussion of the inspiration and practical implications of our research.

1. The series of experiments in this study further confirms that pre-training models have a wealth of knowledge, which endows them with significant few-shot learning ability. The prompt-tuning technique leverages appropriate templates to trigger the activation of the knowledge acquired by PLMs during the pre-training phase. Particularly in the current scenario where labeled data resources are severely limited, this feature becomes especially critical.
2. Metonymy resolution needs to be considered in the context of the success of the prompting paradigm, which was previously built on top of pre-trained models like BERT that were developed for the pre-train paradigm [1,2]. It is unclear whether the pre-training methods that are effective for the latter can be directly applied to the former, or whether we need to entirely rethink pre-training methods to improve the accuracy or applicability of metonymy resolution in prompting-based learning. This important research question has not been extensively explored in the literature.
3. Particularly, the current trend towards increasingly large pre-training model parameters has led to significant waste of deployment resources when fine-tuning a model for a specific task and deploying it for online business. In contrast, the prompt-tuning approach can enable the PLM to work on its original parameters without modification, resulting in significant savings of deployment resources.
4. Regarding the generalizability of PromptMR to other related NLP tasks, the key factor is the transferability of Prompt Learning. PromptMR, being an advanced prompt-based learning approach, exhibits domain-transferability due to its user-friendly prompt engineering. The intuitive construction of prompt templates significantly impacts the model's performance, enabling better model-level and task-level transferability compared to traditional PLM-based fine-tuning methods. By fine-tuning prompt templates and designing novel answer mapping strategies, PromptMR can be adapted to various NLP tasks, including

word sense disambiguation, entity recognition, and relation extraction.

5. We acknowledge the potential challenges and extensions beyond metonymy resolution, especially concerning task complexity. Extending PromptMR to complex NLP tasks may present unique challenges related to data representation, prompt design, and fine-tuning strategies. Additionally, data size and domain adaptation become important factors when dealing with limited data availability in certain tasks. To overcome these limitations, we are committed to addressing these challenges and enhancing the applicability of PromptMR across a broader range of NLP tasks in our future work. Specifically, we plan to explore domain adaptation and transfer learning techniques. Furthermore, we aim to investigate the applicability of PromptMR to tasks involving compositionality and structure, such as natural language inference or question-answering tasks.

In additional to the prompt-tuning solutions for metonymy resolution proposed in this paper, the application of prompt-learning can be especially effective in some specific NLP fields [9], such as named entity recognition [5,6,43], and knowledge-based question answering [18,19]. Overall, this approach has universal characteristics that can help solve many NLP application problems, and from a long-term perspective, the inspiration of our research presented in this paper can benefit almost all areas of textual understanding related to NLP.

8. Conclusion and future perspective

This work was motivated by the notion of using prompt learning to improve the performance of metonymy resolution tasks. To this end, we adapted prompt-tuning to metonymy resolution, devising a series of different approaches to compare for their efficacy under various experimental settings. To the best of our knowledge, PromptMR represents the first systematic exploration of the prompt-tuning paradigm for the metonymy resolution research. Our comprehensive experiments demonstrate that PromptMR yields state-of-the-art performance in terms of accuracy, plus the metonymic-F1, and the literal-F1 metrics for metonymy resolution, confirming the superiority of prompt learning over the currently popular PLM-based fine-tuning techniques. Additionally, to further investigate the potential of prompt learning for handling metonymy resolution, we evaluated two different prompt-tuning enhancing strategies, namely continuous template construction and continuous answer search. Again, a set of comprehensive experiments prove the effectiveness of both these PromptMR variants. Notably, PromptMR performs extremely well in low-resource scenarios, with extensive few-shot learning experiments demonstrating this framework's overwhelming superiority compared to the current PLM-based fine-tuning paradigm. The source code for all our implementations is available at GitHub:https://github.com/albert-jin/PromptTuning2MetonymyResolution.

In the future, we plan to conduct the following research: (1) We will explore the capabilities of PromptMR in handling various minority languages by conducting more experiments on multilingual corpora. Specifically, we plan to leverage the significant translation capabilities of ChatGPT for translating the metonymy resolution datasets into several minority languages. This will also be facilitated by the diverse language-specific PLM versions available in Hugging Face's Transformers community. By doing this, we can easily deploy PromptMR for a wide range of linguistic contexts, including minority languages. (2) We will also be testing different model and architecture innovations to explore the further potential of prompt learning. (3) Inspired by EBAGCN, we plan to infuse external knowledge into the PromptMR framework, such as descriptive knowledge and structured knowledge graphs, to help the PLMs better leverage the advantages of their pre-training and improve metonymy resolution performance. By integrating external knowledge, such as incorporating textual explanations of topic nouns to the subsequent enhanced PLM prompts, we anticipate an increased performance of the PromptMR framework. The fusion of external knowledge as an additional supervision signal has the potential to enhance PromptMR's semantic understanding and answer generation capabilities.

CRediT authorship contribution statement

Biao Zhao: Writing – review & editing, Writing – original draft, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Weiqiang Jin:** Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Methodology, Investigation, Data curation, Conceptualization, Formal analysis. **Yu Zhang:** Validation, Writing – review & editing. **Subin Huang:** Funding acquisition, Supervision, Visualization. **Guang Yang:** Funding acquisition, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: The corresponding author Guang Yang was supported in part by the ERC IMI (101005122), the H2020 (952172), the MRC (MC/PC/21013), the Royal Society (IEC\NSFC\211235), and the UKRI Future Leaders Fellowship (MR/V023799/1). This research was also funded by the University Natural Science Research Projects of Anhui Province under Grant No. 2022AH050972 from the professor Subin Huang.

Data availability

Full code to replicate the experiments and all the datasets used in the experiments can be found at Github: https://github.com/albert-jin/PromptTuning2MetonymyResolution.

Acknowledgments

This work was conducted by Biao Zhao and Weiqiang Jin during their research at Xi'an Jiaotong University. Both the first two authors, Biao Zhao and Weiqiang Jin, contributed equally to this work. The corresponding author of this paper is Guang Yang. Guang Yang was supported in part by the ERC IMI (101005122), the H2020 (952172), the UK Medical Research Council (MRC) (MC/PC/21013), the Royal Society, United Kingdom (IEC\NSFC\211235), and the UKRI Future Leaders Fellowship (MR/V023799/1). This research was also funded by the University Natural Science Research Projects of Anhui Province, China under Grant No. 2022AH050972. The authors would like to express their appreciation for the valuable discussions and perceptive inspirations provided by professor Guizhong Liu from Xi'an Jiaotong University. The authors would also like to thank the anonymous reviewers for their helpful comments, corrections, and recommendations, which significantly improved the quality of the paper.

References

[1] S. Du, H. Wang, Addressing syntax-based semantic complementation: Incorporating entity and soft dependency constraints into metonymy resolution, Future Internet 14 (3) (2022) http://dx.doi.org/10.3390/fi14030085, URL https://www.mdpi.com/1999-5903/14/3/85.

[2] H. Li, M. Vasardani, M. Tomko, T. Baldwin, Target word masking for location metonymy resolution, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 3696–3707, http://dx.doi.org/10.18653/v1/2020.coling-main.330, URL https://aclanthology.org/2020.coling-main.330.

[3] M. Gritta, M.T. Pilehvar, N. Limsopatham, N. Collier, Vancouver welcomes you! minimalist location metonymy resolution, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1248–1259, http://dx.doi.org/10.18653/v1/P17-1115, URL https://aclanthology.org/P17-1115.

[4] Y. Xiao, Q. Du, Statistical age-of-information optimization for status update over multi-state fading channels, 2023, arXiv:2303.11153.

[5] Y. Shen, X. Ma, Z. Tan, S. Zhang, W. Wang, W. Lu, Locate and label: A two-stage identifier for nested named entity recognition, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, Association for Computational Linguistics, 2021, pp. 2782–2794, http://dx.doi.org/10.18653/v1/2021.acl-long.216, URL https://aclanthology.org/2021.acl-long.216.

[6] Y. Shen, X. Wang, Z. Tan, G. Xu, P. Xie, F. Huang, W. Lu, Y. Zhuang, Parallel instance query network for named entity recognition, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 947–961, http://dx.doi.org/10.18653/v1/2022.acl-long.67, URL https://aclanthology.org/2022.acl-long.67.

[7] K. Markert, M. Nissim, SemEval-2007 task 08: Metonymy resolution at SemEval-2007, in: Proceedings of the Fourth International Workshop on Semantic Evaluations, SemEVal-2007, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 36–41, URL https://aclanthology.org/S07-1007.

[8] R. Farkas, E. Simon, G. Szarvas, D. Varga, GYDER: Maxent metonymy resolution, in: Proceedings of the Fourth International Workshop on Semantic Evaluations, SemEVal-2007, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 161–164, URL https://aclanthology.org/S07-1033.

[9] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, ACM Comput. Surv. (2022) http://dx.doi.org/10.1145/3560815, in press.

[10] J. Gao, H. Yu, S. Zhang, Joint event causality extraction using dual-channel enhanced neural network, Knowl.-Based Syst. 258 (2022) 109935.

[11] W. Jin, B. Zhao, L. Zhang, C. Liu, H. Yu, Back to common sense: Oxford dictionary descriptive knowledge augmentation for aspect-based sentiment analysis, Inf. Process. Manage. 60 (3) (2023) 103260, http://dx.doi.org/10.1016/j.ipm.2022.103260, URL https://www.sciencedirect.com/science/article/pii/S0306457322003612.

[12] T. Shin, Y. Razeghi, R.L. Logan IV, E. Wallace, S. Singh, AutoPrompt: Eliciting knowledge from language models with automatically generated prompts, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, Online, Association for Computational Linguistics, 2020, pp. 4222–4235, http://dx.doi.org/10.18653/v1/2020.emnlp-main.346, URL https://aclanthology.org/2020.emnlp-main.346.

[13] X.L. Li, P. Liang, Prefix-tuning: Optimizing continuous prompts for generation, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, Association for Computational Linguistics, 2021, pp. 4582–4597, http://dx.doi.org/10.18653/v1/2021.acl-long.353, URL https://aclanthology.org/2021.acl-long.353.

[14] K. Hambardzumyan, H. Khachatrian, J. May, WARP: Word-level adversarial reprogramming, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 4921–4933, http://dx.doi.org/10.18653/v1/2021.acl-long.381, URL https://aclanthology.org/2021.acl-long.381.

[15] S. Hu, N. Ding, H. Wang, Z. Liu, J. Wang, J. Li, W. Wu, M. Sun, Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2225–2240, http://dx.doi.org/10.18653/v1/2022.acl-long.158, URL https://aclanthology.org/2022.acl-long.158.

[16] Z. Zhong, D. Friedman, D. Chen, Factual probing is [MASK]: Learning vs. Learning to recall, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, Association for Computational Linguistics, 2021, pp. 5017–5033, http://dx.doi.org/10.18653/v1/2021.naacl-main.398, URL https://aclanthology.org/2021.naacl-main.398.

[17] N. Ding, S. Hu, W. Zhao, Y. Chen, Z. Liu, H. Zheng, M. Sun, OpenPrompt: An open-source framework for prompt-learning, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 105–113, http://dx.doi.org/10.18653/v1/2022.acl-demo.10, URL https://aclanthology.org/2022.acl-demo.10.

[18] W. Jin, B. Zhao, H. Yu, X. Tao, R. Yin, G. Liu, Improving embedded knowledge graph multi-hop question answering by introducing relational chain reasoning, Data Min. Knowl. Discov. (2022) http://dx.doi.org/10.1007/s10618-022-00891-8.

[19] W. Jin, B. Zhao, C. Liu, Fintech key-phrase: a new Chinese financial high-tech dataset accelerating expression-level information retrieval, in: The 28th International Conference on Database Systems for Advanced Applications, DASFAA 2023, in: Lecture Notes in Computer Science, Springer, Cham, Tianjin, China, 2023, pp. 425–440.

[20] K. Markert, M. Nissim, Metonymy resolution as a classification task, in: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, EMNLP 2002, Association for Computational Linguistics, Pontacana, Dominican, 2002, pp. 204–213, http://dx.doi.org/10.3115/1118693.1118720, URL https://aclanthology.org/W02-1027.

[21] M. Nissim, K. Markert, Syntactic features and word similarity for supervised metonymy resolution, in: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Sapporo, Japan, 2003, pp. 56–63, http://dx.doi.org/10.3115/1075096.1075104, URL https://aclanthology.org/P03-1008.

[22] A. Zarcone, S. Padó, A. Lenci, Logical metonymy resolution in a words-as-cues framework: Evidence from self-paced reading and probe recognition, Cogn. Sci. 38 (5) (2014) 973–996, http://dx.doi.org/10.1111/cogs.12108, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cogs.12108 URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.12108.

[23] M. Honnibal, M. Johnson, An improved non-monotonic transition system for dependency parsing, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 1373–1378, http://dx.doi.org/10.18653/v1/D15-1162, URL https://aclanthology.org/D15-1162.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS '17, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 6000–6010.

[25] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186, http://dx.doi.org/10.18653/v1/N19-1423, URL https://aclanthology.org/N19-1423.

[26] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language models as knowledge bases? in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 2463–2473, http://dx.doi.org/10.18653/v1/D19-1250, URL https://aclanthology.org/D19-1250.

[27] B. Lester, R. Al-Rfou, N. Constant, The power of scale for parameter-efficient prompt tuning, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 3045–3059, http://dx.doi.org/10.18653/v1/2021.emnlp-main.243, URL https://aclanthology.org/2021.emnlp-main.243.

[28] Y. Gu, X. Han, Z. Liu, M. Huang, PPT: Pre-trained prompt tuning for few-shot learning, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 8410–8423, http://dx.doi.org/10.18653/v1/2022.acl-long.576, URL https://aclanthology.org/2022.acl-long.576.

[29] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (1) (2022) https://dl.acm.org/doi/10.5555/3455716.3455856.

[30] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, J. Tang, GPT understands, too, 2021, http://dx.doi.org/10.48550/ARXIV.2103.10385, arXiv URL https://arxiv.org/abs/2103.10385.

[31] X. Liu, K. Ji, Y. Fu, W. Tam, Z. Du, Z. Yang, J. Tang, P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 61–68, http://dx.doi.org/10.18653/v1/2022.acl-short.8, URL https://aclanthology.org/2022.acl-short.8.

[32] B. Zhao, W. Jin, Z. Chen, Y. Guo, A semi-independent policies training method with shared representation for heterogeneous multi-agents reinforcement learning, Front. Neurosci. 17 (2023).

[33] N. Xia, H. Yu, Y. Wang, J. Xuan, X. Luo, DAFS: A domain aware few shot generative model for event detection, Mach. Learn. 112 (3) (2023) 1011–1031, http://dx.doi.org/10.1007/s10994-022-06198-5.

[34] L. Zhuang, L. Wayne, S. Ya, Z. Jun, A robustly optimized BERT pre-training approach with post-training, in: Proceedings of the 20th Chinese National Conference on Computational Linguistics, Chinese Information Processing Society of China, Huhhot, China, 2021, pp. 1218–1227, URL https://aclanthology.org/2021.ccl-1.108.

[35] B. Zhao, W. Jin, J.D. Ser, G. Yang, ChatAgri: Exploring potentials of ChatGPT on cross-linguistic agricultural text classification, 2023, arXiv:2305.15024.

[36] T. Gao, A. Fisch, D. Chen, Making pre-trained language models better few-shot learners, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, Association for Computational Linguistics, 2021, pp. 3816–3830, http://dx.doi.org/10.18653/v1/2021.acl-long.295, URL https://aclanthology.org/2021.acl-long.295.

[37] G.F. Elsayed, I. Goodfellow, J. Sohl-Dickstein, Adversarial reprogramming of neural networks, in: International Conference on Learning Representations, 2019, URL https://openreview.net/forum?id=Syx_Ss05tm.

[38] V. Nastase, M. Strube, Transforming wikipedia into a large scale multilingual concept network, Artificial Intelligence 194 (2013) 62–85, http://dx.doi.org/10.1016/j.artint.2012.06.008.

[39] S. Zhang, D. Zheng, X. Hu, M. Yang, Bidirectional long short-term memory networks for relation classification, in: Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, Shanghai, China, 2015, pp. 73–78, URL https://aclanthology.org/Y15-1009.

[40] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543, http://dx.doi.org/10.3115/v1/D14-1162, URL https://aclanthology.org/D14-1162.

[41] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 2227–2237, http://dx.doi.org/10.18653/v1/N18-1202, URL https://aclanthology.org/N18-1202.

[42] L. Cui, Y. Wu, J. Liu, S. Yang, Y. Zhang, Template-based named entity recognition using BART, in: Findings of the Association for Computational Linguistics, ACL-IJCNLP 2021, Online, Association for Computational Linguistics, 2021, pp. 1835–1845, http://dx.doi.org/10.18653/v1/2021.findings-acl.161, URL https://aclanthology.org/2021.findings-acl.161.

[43] R. Ma, X. Zhou, T. Gui, Y. Tan, L. Li, Q. Zhang, X. Huang, Template-free prompt tuning for few-shot NER, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 5721–5732, http://dx.doi.org/10.18653/v1/2022.naacl-main.420, URL https://aclanthology.org/2022.naacl-main.420.

[44] R. Alec, W. Jeffrey, C. Rewon, L. David, A. Dario, S. Ilya, Language models are unsupervised multitask learners, OpenAI Blog (2019).

[45] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training, OpenAI Blog (2018).