TKDP: Threefold Knowledge-Enriched Deep Prompt Tuning for Few-Shot Named Entity Recognition

Jiang Liu, Hao Fei, Fei Li, Jingye Li, Bobo Li, Liang Zhao, Chong Teng, and Donghong Ji

***Abstract*—Few-shot named entity recognition (NER) exploits limited annotated instances to identify named mentions. Effectively transferring the internal or external resources thus becomes the key to few-shot NER. While the existing prompt tuning methods have shown remarkable few-shot performances, they still fail to make full use of knowledge. In this work, we investigate the integration of rich knowledge to prompt tuning for stronger few-shot NER. We propose incorporating the deep prompt tuning framework with threefold knowledge (namely *TKDP*), including the internal 1) *context knowledge* and the external 2) *label knowledge* & 3) *sememe knowledge*. TKDP encodes the three feature sources and incorporates them into soft prompt embeddings, which are further injected into an existing pre-trained language model to facilitate predictions. On five benchmark datasets, the performance of our knowledge-enriched model was boosted by at most 11.53% F1 over the raw deep prompt method, and it significantly outperforms 9 strong-performing baseline systems in 5-/10-/20-shot settings, showing great potential in few-shot NER. Our TKDP framework can be broadly adapted to other few-shot tasks without much effort.**

***Index Terms*—Few-shot learning, HowNet, named entity recognition (NER), prompt tuning.**

I. Introduction

NAMED entity recognition aims to extract named mentions (e.g., people, organizations and locations) from text [1], [2]. As one of the fundamental tasks in natural language processing (NLP), NER often serves as an upstream component of more complex tasks such as information retrieval [3], relation extraction [4] and machine reading comprehension [5] etc. Within the last decade, NER has achieved remarkable success with the aid of deep learning techniques, relying on large-scale standard corpora [5], [6]. However, manually constructing annotations is time-consuming, labor-intensive and even infeasible, especially for certain fields, e.g., medicine [7]. This thus demands the research of few-shot NER, to learn a NER system with fewer labeled examples [8].

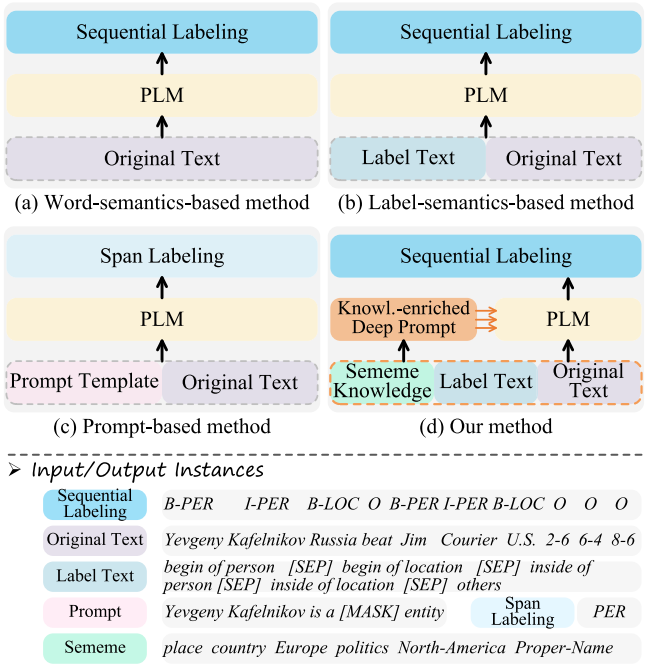


Fig. 1. Comparison between existing few-shot NER methods and our knowledge-enriched deep-prompt-based framework that makes use of threefold knowledge features: sememe, label and context knowledge.

Existing few-shot NER works have made considerable progress [9], [10], which can be technically grouped into three paradigms: **word-semantic-based method**, **label-semantic-based method** and **prompt-based method**. As shown in Fig. 1, the word-semantic-based method [8] depends solely on the input words and their context, while the label-semantic-based method [7] additionally makes use of the label knowledge. In contrast, the prompt-based methods [11] is built upon the current pre-trained language models (PLMs) [12], [13], [14], trying to guide the model to identify entities with pre-built natural language templates, i.e., prompt texts.

While prompt-based methods have achieved better few-shot performances [15], [16], the construction of hard discrete prompt templates can be much experience-oriented and cause unstable results. Also for the NER task, enumerating all possible spans to build prompts is low-efficient. The recent progress of deep

Manuscript received 3 May 2023; revised 29 March 2024; accepted 12 April 2024. Date of publication 16 April 2024; date of current version 27 September 2024. This work was supported by the National Natural Science Foundation of China under Grant 62176187. The work of Liang Zhao would like to thank the supported by Center for Artificial Intelligence (C4AI-USP), in part by the Sao Paulo Research Foundation (FAPESP under Grant #2019/07665-4), in part by IBM Corporation, and in part by the China Branch of BRICS Institute of Future Networks. Recommended for acceptance by F. Ozcan. *(Corresponding authors: Donghong Ji; Fei Li.)*

Jiang Liu, Fei Li, Jingye Li, Bobo Li, Chong Teng, and Donghong Ji are with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, Hubei 430072, China (e-mail: liujiang@whu.edu.cn; lifei_csnlp@whu.edu.cn; theodorelee@whu.edu.cn; boboli@whu.edu.cn; tengchong@whu.edu.cn; dhji@whu.edu.cn).

Hao Fei is with the School of Computing, National University of Singapore, Singapore 119077 (e-mail: haofei37@nus.edu.sg).

Liang Zhao is with the University of São Paulo, São Paulo 05508-220, Brazil (e-mail: zhao@usp.br).

Digital Object Identifier 10.1109/TKDE.2024.3389650

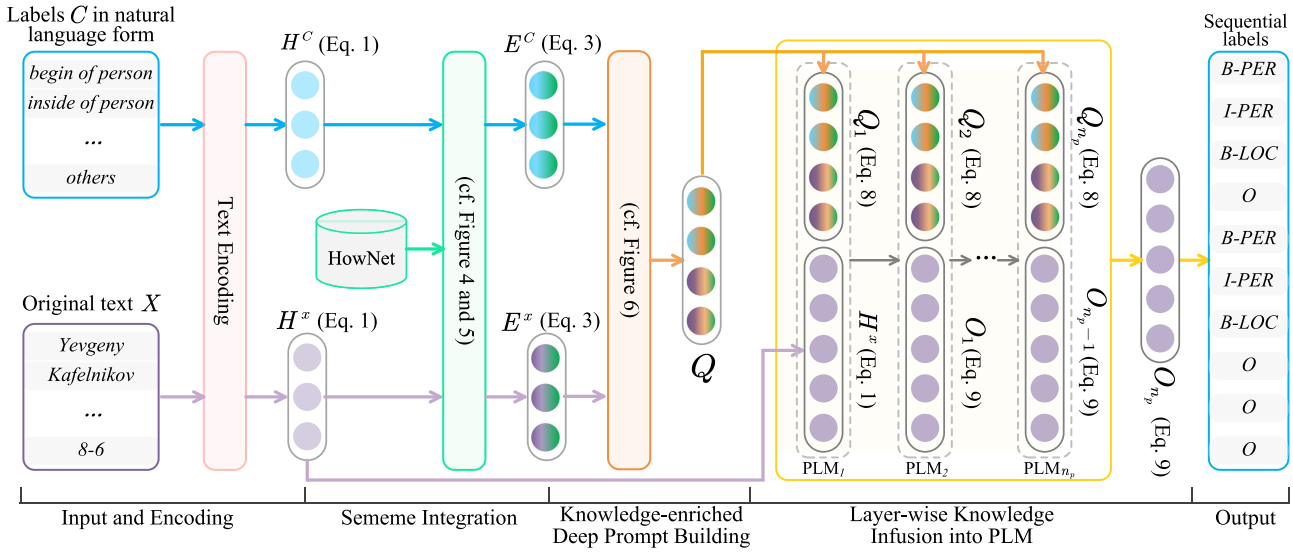


Fig. 2. Overall architecture of our **TKDP** model. $\boldsymbol{H}^x$ and $\boldsymbol{H}^C$ are text embeddings and label embeddings, respectively. $\boldsymbol{E}^x$ and $\boldsymbol{E}^C$ represent sememe-enhanced text embeddings and label embeddings. $\boldsymbol{Q}_i$ is the knowledge-enriched prompt embedding, which is inserted into the i-th layer of PLM.

prompt tuning [17] helps relieve these concerns by considering soft and continuous prompts, which helps few-shot NER achieve remarkable results [18]. Notwithstanding, we note that there are still sufficient rooms to improve. Prior studies extensively reveal that NER relies much on semantic understanding, where the knowledge (internal and external) plays the critical role, especially for the few-shot scenario [19], [20]. However, we observe that the current prompt-based work scarcely considers this. Therefore, this paper investigates the feasibility of incorporating rich knowledge for better prompt-based few-shot NER.

In fact, the deep prompt method is effective in learning semantics from internal **context features**, yet external information is overlooked, such as **label texts** and **sememe knowledge**. Essentially, the entity labels provide rich semantic clues to describe what an entity mention would be [21], while the sememe is a finite and semantically indivisible set of words that can compose the meaning of a word and extend the word semantics [22], [23]. In this work, we take the initiative to enhance the deep prompt method with all three types of internal and external knowledge, aka, threefold knowledge-enriched deep prompt tuning (*TKDP*). We depict the differences between our method and previous methods in Fig. 1.

As shown in Fig. 2, our framework is built based on an existing PLM, equipped with knowledge-enriched soft prompts. Specifically, TKDP first takes the label descriptions and sentential words as inputs. It then embeds the sememe representations from HowNet into the label and text contextual representations for semantics enhancement. Afterward, based on the above sememe-enhanced label and text representations, we generate two types of knowledge-enriched deep prompt embeddings, which are then fused into different layers of PLM for deep interaction. With the incorporation of sememe, label and text context knowledge, the framework is expected to output *BIO* label sequences more accurately.

Extensive experiments are conducted on total five public benchmark datasets across multiple domains. The results show that our method significantly outperforms 9 strong baseline systems in all three few-shot settings (5-/10-/20-shots), and boosts the results by at most 11.53% F1 over the raw soft prompt method, demonstrating the urgent necessity to incorporate internal and external knowledge. We further show that our system wins the current popular prompt learning systems by large margins, including prompt-tuning [24], p-tuning [25], prefix-tuning [26] and discrete template-based prompt methods [11], [27].

Our knowledge-enriched prompt tuning framework can be broadly applied to other few-shot NLP tasks without much effort. All the resources of this paper will be publicly available at https://github.com/solkx/tkdp.git to facilitate related research.

II. Related Work

A. Few-Shot NER

In this section, we briefly survey the existing few-shot NER methods under three categories.

Word-semantic-based Methods: Earlier studies learn the context semantics for few-shot NER from only the input words, i.e., inferring the entity words from their contexts. For example, Yang and Katiyar [8] proposed a model based on the nearest neighbor learning and structured reasoning, where each token is represented in the tokenized examples of the support set by a contextual representation in the sentence. Das et al. [28] used contrastive learning to infer the distribution distance of its Gaussian embedding, thus reducing the distance of token embedding of similar entities and increasing the distance of token embedding of different entities. One of the key issues for this method is feature deficiency, the NER model is hard to learn a robust inductive bias for recognizing unseen named entities with only word semantics.

Label-semantic-based Methods: For few-shot NER, the context features learnt from labeled data are limited, thus label information is considered to be applied in label-semantic-based methods. Intuitively, mention labels describe the attributes of specific entity prototypes in natural language, offering rich extended semantic features of mentions for few-shot NER [21]. Hou et al. [29] first leveraged the semantics of label names for few-shot NER. Huang et al. [7] and Ji et al. [9] created prototype representations from different entity types, and assigned labels through the nearest neighbor criterion according to label dependencies. Ma et al. [30] proposed a two-tower model to incorporate additional signals from label semantic information.

Prompt-based Methods: In recent years, the birth of large PLMs [31] has activated the generative paradigm. Based on PLMs, prompt tuning techniques have shown prominent few-shot performances for multiple NLP tasks [15], [16], yet most of the prompt templates are customized for sentence-level classification rather than NER. Later, Cui et al. [11] first applied prompt tuning to the sequence tagging task, who created prompt templates by enumerating all possible mention spans. Lee et al. [32] and Ma et al. [27] followed the same practice of the hard prompt tuning for few-shot NER. As mentioned earlier, building templates is experience-oriented, and also low-efficient to enumerate the spans. Instead of constructing templates for discrete prompts, Chen et al. [18] leveraged deep prompt tuning [17] for few-shot NER, where the prompts are soft and continuous representations. Although achieving remarkable results, current deep prompt methods still overlook the full utilization of semantic features, which thus motivates our work of knowledge-enriched deep prompt tuning.

B. Sememe

HowNet and Sememes: HowNet is one of the most famous knowledge bases, which defines more than 100,000 Chinese and English words with 2,187 terms [33]. It describes words or phrases through morphemes, which are the smallest units of semantic concepts [34]. Recent studies have shown that HowNets can provide rich well-recognized semantic representations that facilitate downstream NLP tasks, such as word similarity computation [35], word sense disambiguation [36], event detection [37], word representation learning [38], language modeling [39], lexicon expansion [40], relation extraction [41] and semantic rationality evaluation [42].

Jin et al. [43] incorporated Chinese character information into their sememe prediction model and achieved certain performance boost. Qi et al. [44] made the first attempt to use cross-lingual sememe prediction to build a sememe knowledge base for other languages. Qi et al. [45] incorporated sememe knowledge into the semantic composition model for the first time and verified its effectiveness. Qi et al. [23] built a unified sememe knowledge base for multiple languages based on BabelNet (a multilingual encyclopedia dictionary). Lyu et al. [46] proposed the glyph enhanced Chinese character representation, which optimizes the prediction of lexical sememes by focusing on finer-grained information. Qi et al. [47] used image information for the prediction of lexical sememes for the first time.

Sememes for Enriching Semantics: Taking Fig. 3 as example, the word *"U.S."* can be represented as the combination of sememes: *"place"*, *"politics"*, *"North-America"*, *"US"*, *"Proper-Name"* and *"country"*. That is to say, the sememe set completely describes the semantic space of a word and contains rich semantics. This information is beneficial to the extraction of entities. For example, *"U.S."* and *"Russia"* share the latent sememe concept *"place"*, which can strengthen their connection with the entity type *"LOC"*.

C. Knowledge-Enhanced Prompt Tuning

It is also common practice to leverage external or internal knowledge to enhance the downstream NLP tasks. Here we briefly survey the literature concerning knowledge-enhanced prompt learning methods. Shin et al. [48] used a gradient guided search algorithm to automatically construct discrete prompts for various task sets, consisting of raw input and trigger words. The entire process does not involve fusion of knowledge, so it is different from our automatic construction of knowledge-fused prompts. Hu et al. [49] incorporated the external knowledge into the verbalizer to form knowledgeable prompt tuning to improve and stabilize prompt tuning. Specifically, Hu et al. [49] used a knowledge base to extend the label word space of the verbalizer and used the PLM to refine the extended label words before using them for prediction. Chen et al. [50] injected the latent knowledge contained in relation labels into the construction of prompts with learnable virtual type words and answer words to solve the few-shot relation extraction problem. Li et al. [51] set the event detection task as condition generation. Then, knowledge-injected prompts are constructed using external knowledge bases, and a prompt tuning strategy is leveraged to optimize the prompts. Our work differs from the above works in multiple aspects: First, we consider the use of multiple knowledge from both internal and external sources; Besides, we consider the sememe knowledge from HowNet to enhance the text semantics, while the above works have not; Finally, we focus on few-shot NER tasks, while the above methods are more suitable for classification tasks rather than sequence labeling tasks.

III. Methodology

Fig. 2 presents our overall framework, which consists of five parts, including the input and encoding layer, the sememe integration layer, knowledge-enriched deep prompt construction module, layer-wise knowledge infusion module, and output layer.

A. Input and Encoding

Our system takes two input sources: 1) the original text $X = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^n$ of length n, where x_i represents the i-th word; and 2) the natural language description of L labels $C = \{C_1, C_2, \ldots, C_L\} \in \mathbb{R}^L$, where each label description $C_j = \{c_{j,1}, c_{j,2}, \ldots, c_{j,l}\} \in \mathbb{R}^l$ consists of l description

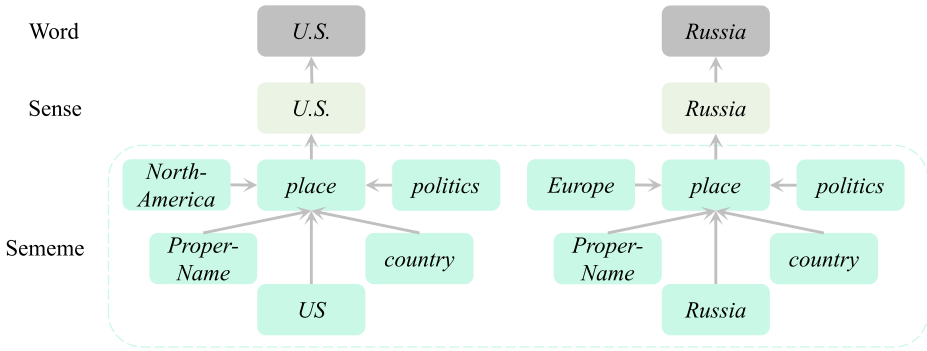


Fig. 3. Illustration of word, sense and sememe in HowNet.

TABLE I
NATURAL LANGUAGE FORM CORRESPONDING TO EACH ENTITY TYPE OF THE DATASETS

Datasets	Original Label Type	Natural Language Form
CoNLL-2003	ORG	organization
	MISC	miscellaneous
	PER	person
	LOC	location
WNUT-2017	location	location
	group	group
	corporation	corporation
	person	person
	creative_work	creative work
	product	product
MIT-Restaurant	Rating	rating
	Amenity	amenity
	Location	location
	Restaurant_Name	restaurant name
	Price	price
	Hours	hour
	Dish	dish
	Cuisine	cuisine
MIT-Movie	Actor	actor
	Plot	plot
	Opinion	opinion
	Award	award
	Year	year
	Genre	genre
	Origin	origin
	Director	director
	Soundtrack	soundtrack
	Relationship	relationship
	Character_Name	character name
	Quote	quote
JNLPBA	DNA	deoxyribonucleic acid
	protein	protein
	cell_type	cell type
	cell_line	cell line
	RNA	ribonucleic acid

words. For example, the description for the label PER can be unfolded as:

$$B\text{-}PER \rightarrow begin\ of\ person$$
$$I\text{-}PER \rightarrow inside\ of\ person$$
$$O \rightarrow others$$

The natural language forms corresponding to the original entity types of all datasets are detailed in Table I. We encode input text and label description using the embedding layer of BERT [12]. Then X and C are encoded as:

$$\boldsymbol{H}^x = \{\boldsymbol{h}_1^x, \boldsymbol{h}_2^x, \ldots, \boldsymbol{h}_n^x\} \in \mathbb{R}^{n \times d_h},$$
$$\boldsymbol{H}^C = \{\boldsymbol{H}_1^c, \boldsymbol{H}_2^c, \ldots, \boldsymbol{H}_L^c\} \in \mathbb{R}^{L \times l \times d_h},$$
$$where\ \boldsymbol{H}_j^c = \{\boldsymbol{h}_{j,1}^c, \boldsymbol{h}_{j,2}^c, \ldots, \boldsymbol{h}_{j,l}^c\} \in \mathbb{R}^{l \times d_h}, \quad (1)$$

where $\boldsymbol{h}_i^x, \boldsymbol{h}_{j,i}^c \in \mathbb{R}^{d_h}$ are the representations of the i-th word, and d_h represents the dimension of a word representation.

B. Sememe Integration

Sememe is a language-independent smallest semantic unit [22], which has been leveraged as an external knowledge base in many NLP tasks to enrich the semantics of inputs, e.g., word sense disambiguation [36], event detection [37] and relation extraction [41]. In HowNet, a word is defined with one or more senses, and each sense contains one or more sememes. Now we consider embedding the sememe information into our framework. We use two embedding methods, discriminating by whether to incorporate structured information between sememes or not.

Unstructured Sememe Integration: Fig. 4 shows an illustration of sememe knowledge integration without structured information. First, we obtain the sememe set $S = \{s_1, s_2, \ldots, s_m\} \in \mathbb{R}^m$ of the target word from HowNet, where m is the number of sememes. Then obtain the representation of the sememe set $\boldsymbol{H}^s = \{\boldsymbol{h}_1^s, \boldsymbol{h}_2^s, \ldots, \boldsymbol{h}_m^s\} \in \mathbb{R}^{m \times d_h}$ is obtained using text encoder, where $\boldsymbol{h}_i^s \in \mathbb{R}^{d_h}$ represents the i-th sememe representation. Note that the embeddings for sememe words are shared between the input sentence and labels.

After obtaining the sememe representations, we use an attention mechanism to retrieve those sememes that are more related to the query word. The attention score of the i-th sememe representation $\boldsymbol{h}_i^s$ and the word representation $\boldsymbol{h} \in \mathbb{R}^{d_h}$ is calculated by the euclidean distance:[1]

$$r_i = \frac{d_i}{\sum_{i=1}^{m} d_i}, \ d_i = \sqrt{\sum_{j=1}(h_j - h_{i,j}^s)^2}, \quad (2)$$

[1]Only in this subsection, we unify the previous word representations $\boldsymbol{h}_i^x$ and $\boldsymbol{h}_{j_i}^c$ to $\boldsymbol{h}$ for brevity.

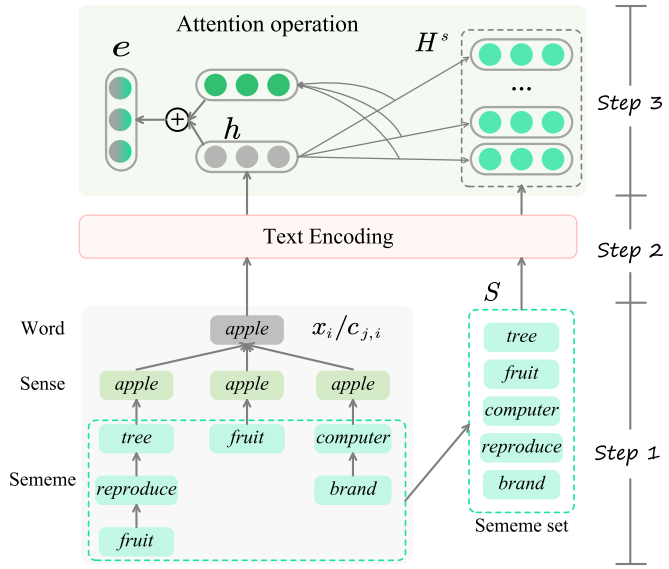


Fig. 4. Illustration of the sememe knowledge integration without structured information. **Step 1**: all the sememes (e.g., fruit) of the word (e.g., apple) are grouped into the sememe set S without considering order or hierarchy. **Step 2**: S and the target word are encoded as representations, $\boldsymbol{H}^s$ and $\boldsymbol{h}$. **Step 3**: $\boldsymbol{H}^s$ and $\boldsymbol{h}$ are fused via attention and residual to obtain $\boldsymbol{e}$.

where h_j and $h^s_{i,j}$ are the scalar of the j-th dimension of word representation and sememe representation respectively. Finally, the sememe-enhanced word representation $\boldsymbol{e} \in \mathbb{R}^{d_h}$ is obtained via $\boldsymbol{e} = \boldsymbol{h} + \sum_{i=1}^{m} r_i \cdot \boldsymbol{h^s_i}$.

Now we obtain the sememe-enriched text representation $\boldsymbol{E}^x \in \mathbb{R}^{n \times d_h}$ and the sememe-enriched label set representation $\boldsymbol{E}^C \in \mathbb{R}^{L \times l \times d_h}$:

$$\begin{aligned} \boldsymbol{E}^x &= \{\boldsymbol{e}^x_1, \boldsymbol{e}^x_2, \ldots, \boldsymbol{e}^x_n\}, \\ \boldsymbol{E}^C &= \{\boldsymbol{E}^c_1, \boldsymbol{E}^c_2, \ldots, \boldsymbol{E}^c_L\}, \\ \boldsymbol{E}^c_j &= \{\boldsymbol{e}^c_{j,1}, \boldsymbol{e}^c_{j,2}, \ldots, \boldsymbol{e}^c_{j,l}\} \in \mathbb{R}^{l \times d_h}, \end{aligned} \tag{3}$$

where $\boldsymbol{e}^x_i$, $\boldsymbol{e}^c_{j,i} \in \mathbb{R}^{d_h}$ are the representations of the i-th word integrated with sememe konwledge.

Structured Sememe Integration: Fig. 5 shows an illustration of sememe knowledge integration with structured information. First, we obtain the sense graph set G of target word from HowNet:

$$\begin{aligned} G &= \{g_1, g_2 \ldots, g_z\} \in \mathbb{R}^z, \\ g_i &= [V_i, A_i], \\ V_i &= \{v_{i,1}, v_{i,2}, \ldots, v_{i,z_v}\} \in \mathbb{R}^{z_v}, \end{aligned} \tag{4}$$

where g_i represents the i-th sense graph, z represents the number of sense graphs, V_i represents the node set of the i-th sense graph, z_v represents the number of nodes, and $A_i \in \mathbb{R}^{z_v \times z_v}$ represents the corresponding adjacency matrix. The embeddings for sememe words are shared between the input sentence and labels.

After encoding, the sense graph representation set $\boldsymbol{H}^g$ can be obtained:

$$\boldsymbol{H}^g = \{\boldsymbol{h}^g_1, \boldsymbol{h}^g_2, \ldots, \boldsymbol{h}^g_z\},$$

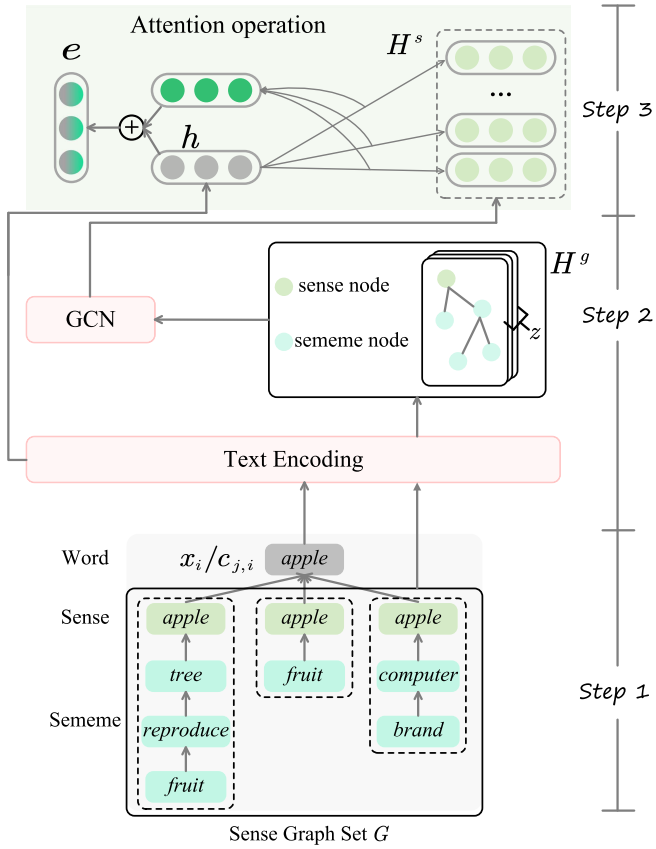


Fig. 5. Illustration of the sememe knowledge integration with structured information. **Step 1**: we build a sense graph G for the target word (e.g., apple). **Step 2**: sememe words and the target word are encoded as representations $\boldsymbol{H}^g$ and $\boldsymbol{h}$, and then $\boldsymbol{H}^g$ is integrated with structured information through GCN to generate $\boldsymbol{H}^s$. **Step 3**: $\boldsymbol{H}^s$ and $\boldsymbol{h}$ are fused via attention and residual.

$$\begin{aligned} \boldsymbol{h}^g_i &= [\boldsymbol{H}^V_i, A_i], \\ \boldsymbol{H}^V_i &= \{\boldsymbol{h}^v_{i,1}, \boldsymbol{h}^v_{i,2}, \ldots, \boldsymbol{h}^v_{i,z_v}\} \in \mathbb{R}^{z_v \times d_h}, \end{aligned} \tag{5}$$

where $\boldsymbol{h}^g_i$ represents the i-th sense graph representation, $\boldsymbol{H}^V_i$ denotes the representation of its node set and $\boldsymbol{h}^v_{i,j} \in \mathbb{R}^{d_h}$ indicates the j-th node representation.

Then we use the graph convolution network (GCN) [52] to process the sense graph to extract structured features, and finally output the representations of the sense nodes to construct the representation of the sense graph set $\boldsymbol{H}^s \in \mathbb{R}^{z \times d_h}$. Since its subsequent role is the same as the one used in unstructured sememe integration, we use the same symbol $\boldsymbol{H}^s$. Also the subsequent steps for calculating are the same as the ones for unstructured sememe integration.

C. Knowledge-Enriched Deep Prompt Building

Next, we create the knowledge-enriched deep prompt representations based on $\boldsymbol{E}^x$ and $\boldsymbol{E}^c$. As shown in Fig. 6, we first use two randomly-initialize prompt embeddings $\boldsymbol{P}^x$ and $\boldsymbol{P}^c \in \mathbb{R}^{l_p \times n_p \times d_h}$ to fused with $\boldsymbol{E}^x$ and $\boldsymbol{E}^C$, where l_p represents the length of the prompt and n_p represents the depth of the prompt (the number of layers embedded in the PLM). Technically, we map $\boldsymbol{P}^x$, $\boldsymbol{P}^c$, $\boldsymbol{E}^x$ and $\boldsymbol{E}^C$ to the same

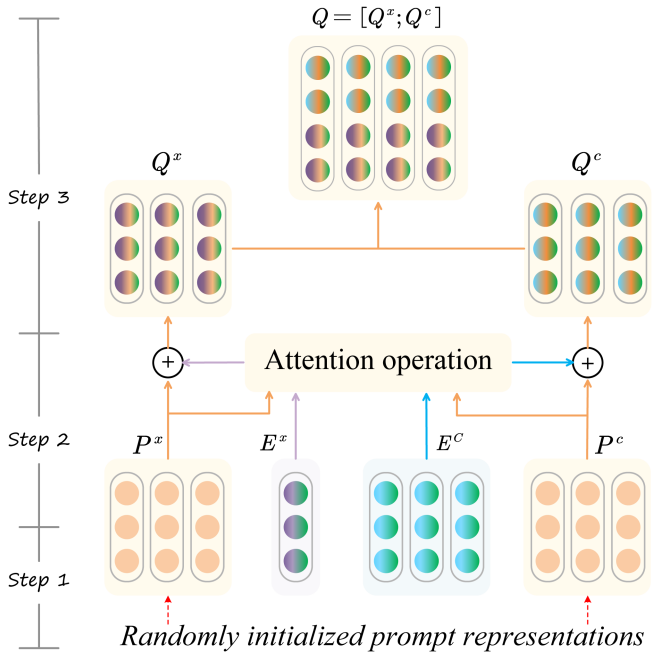


Fig. 6. Procedure of building knowledge-enriched deep prompt representations. **Step 1**: prompt embedding $\boldsymbol{P}^x$ and $\boldsymbol{P}^c$ are generated; **Step 2**: $\boldsymbol{P}^x$ and $\boldsymbol{P}^c$ are fused with the sememe-enhanced context representation $\boldsymbol{E}^x$ and label representation $\boldsymbol{E}^C$ via attention operation; **Step 3**: the knowledge-enriched context prompt $\boldsymbol{Q}^x$ and label prompt $\boldsymbol{Q}^c$ are concatenated into unified deep prompt representation $\boldsymbol{Q}$.

feature space:

$$\hat{\boldsymbol{P}}^x = \boldsymbol{W}^p \boldsymbol{P}^x + \boldsymbol{b}^p,\ \hat{\boldsymbol{P}}^c = \boldsymbol{W}^p \boldsymbol{P}^c + \boldsymbol{b}^p,$$
$$\hat{\boldsymbol{E}}^x = \boldsymbol{W}^x \boldsymbol{E}^x + \boldsymbol{b}^x,\ \hat{\boldsymbol{E}}^C = \boldsymbol{W}^c \boldsymbol{E}^C + \boldsymbol{b}^c, \tag{6}$$

where $\boldsymbol{W}^p$, $\boldsymbol{W}^x$, $\boldsymbol{W}^c \in \mathbb{R}^{d_h \times d_h}$ and $\boldsymbol{b}^p$, $\boldsymbol{b}^x$, $\boldsymbol{b}^c \in \mathbb{R}^{d_h}$ are trainable weights and biases.

Then $\hat{\boldsymbol{E}}^x$ is integrated into $\hat{\boldsymbol{P}}^x$ through the attention mechanism. We calculate the attention score between two representations in $\hat{\boldsymbol{P}}^x$ and $\hat{\boldsymbol{E}}^x$. For the j-th representation $\hat{\boldsymbol{p}}^x_{i,j} \in \mathbb{R}^{d_h}$ of the i-th layer in $\hat{\boldsymbol{P}}^x$ ($1 \leqslant i \leqslant n_p$, $1 \leqslant j \leqslant l_p$), and the k-th representation $\hat{\boldsymbol{e}}^x_k \in \mathbb{R}^{d_h}$ in $\hat{\boldsymbol{E}}^x$ ($1 \leqslant k \leqslant n$), we compute the dot product between $\hat{\boldsymbol{p}}^x_{i,j}$ and $\hat{\boldsymbol{e}}^x_k$ and obtain their attention score:

$$U^x_{i,j,k} = \frac{\exp(\hat{\boldsymbol{p}}^x_{i,j} \cdot \hat{\boldsymbol{e}}^x_k)}{\sum_{k=1}^{n} \exp(\hat{\boldsymbol{p}}^x_{i,j} \cdot \hat{\boldsymbol{e}}^x_k)}. \tag{7}$$

We use the same method to integrate $\hat{\boldsymbol{E}}^C$ into $\hat{\boldsymbol{P}}^c$. But before that, we first average $\hat{\boldsymbol{E}}^C$ using average pooling to get a sentence-level representation of the label: $\hat{\boldsymbol{E}}^c = \text{AvgPooling}(\hat{\boldsymbol{E}}^C) \in \mathbb{R}^{L \times d_h}$. Then the subsequent operation is the same as the operation in (7), so that the attention score $U^c_{i,j,k}$ can be obtained.

Afterwards, knowledge-enriched context prompt $\boldsymbol{Q}^x \in \mathbb{R}^{l_p \times n_p \times d_h}$ and label prompt $\boldsymbol{Q}^c \in \mathbb{R}^{l_p \times n_p \times d_h}$ are obtained:

$$\hat{\boldsymbol{q}}^x_{i,j} = \hat{\boldsymbol{p}}^x_{i,j} + \sum_{k=1}^{n} U^x_{i,j,k} \cdot \hat{\boldsymbol{e}}^x_k,$$
$$\boldsymbol{q}^x_{i,j} = \text{Tanh}\left(\hat{\boldsymbol{W}}^x \hat{\boldsymbol{q}}^x_{i,j} + \hat{\boldsymbol{b}}^x\right),$$

TABLE II
STATISTICS ON FIVE COMMON BENCHMARK DATASETS

	Domain	#Train	#Test	#Entity
CoNLL-2003	News	14,042	3,454	4
WNUT-2017	Social	3,395	1,288	6
MIT-Movie	Review	7,817	1,954	12
MIT-Restaurant	Review	7,661	1,522	8
JNLPBA	Biology	16,806	3,855	5

$$\hat{\boldsymbol{q}}^c_{i,j} = \hat{\boldsymbol{p}}^c_{i,j} + \sum_{k=1}^{L} U^c_{i,j,k} \cdot \hat{\boldsymbol{e}}^c_k,$$
$$\boldsymbol{q}^c_{i,j} = \text{Tanh}\left(\hat{\boldsymbol{W}}^c \hat{\boldsymbol{q}}^c_{i,j} + \hat{\boldsymbol{b}}^c\right), \tag{8}$$

where $\text{Tanh}(\cdot)$ is the hyperbolic tangent function, $\hat{\boldsymbol{W}}^x$, $\hat{\boldsymbol{W}}^c \in \mathbb{R}^{d_h \times d_h}$ and $\hat{\boldsymbol{b}}^x$, $\hat{\boldsymbol{b}}^c \in \mathbb{R}^{d_h}$ are trainable weights and biases, $\boldsymbol{q}^x_{i,j} \in \mathbb{R}^{d_h}$ and $\boldsymbol{q}^c_{i,j} \in \mathbb{R}^{d_h}$ are the j-th representations of the i-th layer in $\boldsymbol{Q}^x$ and $\boldsymbol{Q}^c$. Finally, $\boldsymbol{Q}^x$ and $\boldsymbol{Q}^c$ are concatenated to obtain the knowledge-enriched deep prompt $\boldsymbol{Q} = [\boldsymbol{Q}^x; \boldsymbol{Q}^c] \in \mathbb{R}^{2l_p \times n_p \times d_h}$.

D. Layer-Wise Knowledge Infusion Into PLM

We now insert the deep prompt $\boldsymbol{Q}$ into different layers of PLM [17]. We use $\boldsymbol{Q}_i \in \mathbb{R}^{2l_p \times d_h}$ to represent the i-th layer of $\boldsymbol{Q}$. Specifically, we concatenate $\boldsymbol{Q}_i$ with the output $\boldsymbol{O}_{i-1} \in \mathbb{R}^{n \times d_h}$ of the (i-1)-th layer of the PLM as the input of the i-th layer of the PLM. The input of the first layer of the PLM is the concatenation of $\boldsymbol{Q}_1$ and $\boldsymbol{H}^x$:

$$\begin{cases} \boldsymbol{O}_1 = \text{PLM}_1([\boldsymbol{Q}_1; \boldsymbol{H}^x]), i = 1, \\ \boldsymbol{O}_i = \text{PLM}_i([\boldsymbol{Q}_i; \boldsymbol{O}_{i-1}]), 1 < i \leqslant n_p. \end{cases} \tag{9}$$

E. Output and Learning

Finally, the output $\boldsymbol{O}_{n_p}$ of the last layer of PLM is used to predict the *BIO* label sequence via $\boldsymbol{y} = \text{Softmax}(\boldsymbol{O}_{n_p})$. For each X, the goal of our few-shot training is to minimize the negative log-likelihood loss between the predicted probability of the sequence $\boldsymbol{y}$ and the probability of the corresponding gold sequence $\boldsymbol{y}^g$, formalized as:

$$\mathcal{L} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{r \in C_o, r=1}^{L} \boldsymbol{y}^g_{i,r} \log \boldsymbol{y}_{i,r}, \tag{10}$$

where $\boldsymbol{y}^g_{i,r}$ and $\boldsymbol{y}_{i,r}$ represent the true and predicted probabilities that the i-th word should be assigned the label r. $C_o \in \mathbb{R}^L$ represents the label set and L denotes its size.

IV. Experimental Setups

A. Datasets

We evaluate our methods on five public benchmark NER datasets, namely CoNLL-2003 [53] on news domain, WNUT-2017 [54] on social domain, MIT-Movie [55] and MIT-Restaurant [56] on review domain, JNLPBA [57] on biology domain. Table II shows the statistics of these datasets.

B. Baselines

StructShot [8] utilizes a nearest neighbor classifier for few-shot predictions. **CONTaiNER** [28] leverages contrastive learning to infer the distributional distance of Gaussian embeddings of entities. **Prototype** [7] builds a prototypical network [58] and utilizes the nearest neighbor criterion to assign the entity category. **LabelBERT** [30] uses the semantics of label names as additional signals. **EntLM** [27] is a few-shot NER method that leverages an entity-oriented LM objective. **TemplateNER** [11] is a prompt-based approach, which enumerates all possible n-gram spans as templates and classifies each of them. **LightNER** [18] constructs deep prompts into self-attention layers via a bootstrap mechanism. **SEE-Few** [10] uses context clues and entity type information to recombine candidate spans into entities and then classify them. **FFF-NER** [59] introduces three new token types, *"is entity", "which type"* and *bracket*, and fine-tunes the pretraining model through masked token prediction or generation. Their method predicts entity span and entity type separately, unlike sequential labeling models that simultaneously predict span and type. **Llama-2** [60] is a collection of pre-trained and fine-tuned large language models, ranging in size from 7 billion to 70 billion parameters.

Although, there are also some other few-shot NER works, we do not make comparisons with them for some reasons:

a) Unfair Comparison: We do not compare with the methods of Chen et al. [21] and Ji et al. [9], because their methods are pre-trained on a large corpus, while our method does not require such pre-training.

b) Different Data: Although Huang et al. [61] reported the model performances without the pre-training stage, they only have the results of the CoNLL-2003 and MIT-Movie datasets, where the MIT-Movie dataset (with more sentences and entities) is different from ours. For the CoNLL-2003 dataset, our method performs better in all settings.

c) Inaccessible Code: In addition, Ji et al. [9] have not disclosed their source code. Thus we are unable to re-implement their results and make fair comparisons.

C. Implementation Details

We conduct experiments on 5-shot, 10-shot and 20-shot respectively. For the k-shot setting, we sample k instances of each entity type from the training set. We randomly sample five different training sets and one development set for each setting, each training sample tuned at different prompt lengths, then we report the mean and standard deviation of the five training samples.

We evaluate our model using precision (P), recall (R) and F1, where a predicted entity is considered correct when its boundary and type are exactly the same as these of a golden entity. In addition, the effects of all baseline models are obtained without using source domain training data. We also reproduce the results of datasets not reported in baseline articles.

Except that TemplateNER is based on BART-base [14], other models are based on BERT-base [12]. During training, only the knowledge-enriched deep prompt in Fig. 2 is trainable, and other parts are frozen. The hyperparameter settings of our model are shown in Table III. Our model is implemented using PyTorch and trained using NVIDIA RTX 3090 GPU.

TABLE III
HYPER-PARAMETER SETTINGS

Hyper-parameter	value
Epoch	200
l_p	12, 24, 36, 48, 60
Batch Size	16
Learning Rate	5e-5
n	128
l	10
d_h	768
n_p	12
Dropout	0.1

V. RESULTS AND ANALYSIS

A. Main Comparisons

Table IV shows the overall comparison results. As can be seen from the table, our method is able to achieve the best and second best performance compared to all the baseline models. From the average results, our method is slightly inferior to FFF-NER. One likely reason is that FFF-NER fine-tunes BERT, while our method does not. The reason that we maintain such setting is to follow the soft prompt approach [24]. We aim to quickly adapt our model to a specific few-shot task without fine-tuning the large pretrained model, which is a more efficient way for few-shot learning. We also demonstrate the advantage of our method by comparing the training and inference efficiency, as shown in Table VII. Our method is 52 times faster than FFF-NER in terms of training speed.

In addition, after introducing the structured information of sememes (TKDP+struct), the performance of our model has been improved in most cases. Averagely, the P, R and F1 of our model is improved by 0.50%, 0.86% and 0.18% respectively under 5-shot, 10-shot and 20-shot settings, which shows the advantage of using structured information.

In addition, we also test the zero-shot and five-shot performance of large language models such as Llama-2-13B in our task. In the zero-shot setting, Llama achieves 47.46%, 27.94%, 38.17%, 43.29% and 48.68% respectively under CoNLL-2003, WNUT-2017, MIT-Movie, MIT-Restaurant and JNLPBA datasets. In the five-shot setting, the performance is 58.54%, 28.31%, 39.12%, 45.50% and 50.11% respectively, which is improved compared to the results of zero-shot, but compared with our method, it is lower than our method on four datasets. From the average results, our method is still 6.37% (50.69% - 44.32%) higher than the result of Llama. This demonstrates that although large language models contain sufficient knowledge, they cannot be easily adapted to every field, thus showing the necessity of few-shot fine-tuning. Besides quantitative analysis, we also manually observed the errors generated by Llama. Two kinds of errors are frequently generated: First, the schema of

TABLE IV
COMPARISON OF OUR METHOD WITH BASELINE MODELS

		5-Shot	10-Shot	20-Shot	5-Shot	10-Shot	20-Shot
		CoNLL-2003 (News)			*WNUT-2017* (Social)		
• **Word-semantics-based**	StructShot	45.82(10.30)	62.37(10.96)	69.51(6.46)	20.99(6.83)	27.54(4.57)	33.13(2.68)
	CONTaiNER	51.70(9.97)	63.50(9.65)	71.00(6.44)	21.56(5.44)	28.48(4.06)	33.85(2.74)
• **Label-semantics-based**	Prototype	40.37(8.06)	52.83(3.22)	53.89(1.95)	17.42(3.67)	20.09(1.57)	22.18(0.87)
	LabelBERT	31.78(2.43)	37.81(5.64)	51.25(3.08)	11.47(3.01)	15.41(2.79)	23.11(1.62)
• **Prompt-based**	EntLM	49.59(8.30)	64.79(3.86)	69.52(4.48)	24.82(2.90)	31.28(1.71)	34.75(1.61)
	TemplateNER	43.04(6.15)	57.86(5.68)	66.38(6.09)	19.25(2.88)	25.53(2.21)	31.91(0.99)
	LightNER	28.62(4.60)	42.96(5.47)	65.25(5.95)	18.13(3.93)	24.97(1.55)	31.29(1.16)
• **Others**	SEE-Few	55.21(3.93)	61.99(1.73)	68.21(2.60)	22.49(4.53)	30.69(1.91)	34.63(2.59)
	FFF-NER	**67.90**(3.95)	**72.07**(1.76)	76.28(0.72)	**30.48**(3.69)	**34.42**(2.43)	**36.88**(2.05)
• LLMs	Llama-2-13B	58.54(1.86)	——	——	28.31(1.89)	——	——
• **Ours**	TKDP	56.72(3.54)	64.87(3.27)	73.43(0.69)	25.94(3.31)	32.48(1.07)	35.69(1.69)
	TKDP+struct	57.99(4.00)	66.05(1.08)	75.07(0.77)	26.93(3.53)	32.89(1.62)	35.30(1.75)
	TKDP+FFF	63.99(3.43)	72.02(1.70)	**77.09**(1.45)	28.42(1.09)	31.55(3.72)	34.76(2.07)
		MIT-Movie (Review)			*MIT-Restaurant* (Review)		
• **Word-semantics-based**	StructShot	41.60(8.97)	53.19(5.52)	60.42(2.98)	31.93(4.32)	34.30(2.56)	42.69(1.12)
	CONTaiNER	51.71(8.55)	57.10(5.38)	60.65(2.65)	45.50(8.99)	48.81(5.61)	51.25(3.14)
• **Label-semantics-based**	Prototype	31.47(2.39)	38.08(1.64)	43.64(1.19)	44.10(4.66)	45.94(3.78)	53.46(2.80)
	LabelBERT	44.64(2.89)	53.90(1.71)	59.17(1.44)	39.67(2.54)	48.01(2.23)	57.14(1.56)
• **Prompt-based**	EntLM	46.62(9.46)	57.31(3.72)	62.36(4.14)	42.60(4.26)	55.90(1.48)	63.00(1.22)
	TemplateNER	45.97(3.86)	49.30(3.35)	59.09(0.35)	39.22(3.26)	46.00(2.22)	57.10(0.98)
	LightNER	26.77(0.65)	40.58(0.48)	50.61(1.21)	30.74(4.38)	52.90(2.81)	59.97(7.03)
• **Others**	SEE-Few	36.81(2.25)	42.06(1.08)	50.67(1.21)	45.25(3.18)	51.20(1.48)	60.75(2.07)
	FFF-NER	**60.32**(1.12)	**65.07**(1.07)	**67.30**(0.73)	**51.99**(2.33)	**60.60**(1.25)	**65.86**(1.14)
• LLMs	Llama-2-13B	39.12(2.12)	——	——	45.50(3.03)	——	——
• **Ours**	TKDP	53.46(0.71)	59.56(0.84)	62.87(0.73)	47.44(3.24)	56.82(0.97)	61.76(1.67)
	TKDP+struct	53.59(1.13)	59.49(0.58)	62.38(0.84)	46.69(2.25)	55.98(0.97)	62.23(1.08)
	TKDP+FFF	58.91(0.91)	62.33(0.48)	65.20(0.68)	51.64(2.32)	59.56(1.03)	64.28(0.85)
		JNLPBA (Biolog)			*Avg. of All Dataset*		
• **Word-semantics-based**	StructShot	30.52(3.14)	35.54(3.02)	45.69(2.10)	34.17	42.59	50.29
	CONTaiNER	32.77(3.41)	38.70(5.61)	46.65(1.77)	40.65	47.32	52.68
• **Label-semantics-based**	Prototype	27.99(2.28)	36.17(3.87)	46.06(1.50)	32.27	38.62	43.85
	LabelBERT	26.18(3.66)	34.67(4.15)	46.78(2.16)	30.75	37.96	47.49
• **Prompt-based**	EntLM	35.45(2.08)	41.21(2.19)	48.03(1.89)	39.82	50.10	55.53
	TemplateNER	33.28(2.99)	40.44(2.01)	44.07(1.54)	36.15	43.83	51.71
	LightNER	24.89(1.98)	27.41(1.97)	42.43(1.20)	25.83	37.76	49.91
• **Others**	SEE-Few	29.08(5.63)	39.31(2.12)	46.42(3.28)	37.77	45.05	52.14
	FFF-NER	50.48(2.85)	53.53(2.53)	57.28(0.75)	**52.82**	**57.94**	**61.12**
• LLMs	Llama-2-13B	50.11(2.58)	——	——	44.32	——	——
• **Ours**	TKDP	33.57(1.36)	42.44(1.62)	49.31(1.45)	43.43	51.23	56.61
	TKDP+struct	34.45(0.93)	43.04(1.97)	48.98(1.40)	43.93	52.09	56.79
	TKDP+FFF	**50.49**(1.81)	**55.33**(1.44)	**58.51**(0.76)	50.69	56.16	59.97

Values in parentheses are standard deviations, green score is the highest result in each column of the dataset, blue score is the second highest result in each column of the dataset. TKDP+struct represents that TKDP are integrated with structured information of sememes as illustrated in fig. 5. TKDP+FFF [60] represents that the method proposed in FFF-NER is leveraged in TKDP.

Llama is inconsistent with the schema of the datasets. For example, Llama will classify the entities "world cup" and "Uzbek" as ORG and LOC based on common sense, but their gold labels in the CoNLL-2003 dataset are both MISC. Second, Llama extracts some general entities, such as "player" and "he", but they are not regarded as entities in the CONLL-2003 dataset.

B. Ablation Experiment

We now work on our method itself, investigating the impact of three kinds of knowledge on model performance. TKDP-SK means that the sememe integration steps in Section III-B are skipped and only label knowledge and context knowledge are integrated into soft prompts. TKDP-CK means that context

TABLE V
ABLATION STUDIES

	5-Shot	10-Shot	20-Shot	5-Shot	10-Shot	20-Shot	5-Shot	10-Shot	20-Shot
	CoNLL-2003			*WNUT-2017*			*MIT-Movie*		
TKDP	**56.72**(3.54)	64.87(3.27)	73.43(0.69)	**25.94**(3.31)	**32.48**(1.07)	**35.69**(1.69)	**53.46**(0.71)	**59.56**(0.84)	**62.87**(0.73)
TKDP− CK	55.73(4.17)	**66.15**(2.83)	**73.65**(1.57)	24.99(3.22)	31.01(0.96)	34.53(1.90)	51.98(0.51)	57.37(0.97)	61.85(0.67)
TKDP− LK	52.90(3.44)	62.28(2.48)	68.70(1.76)	24.41(3.74)	29.40(1.65)	34.10(1.41)	52.33(0.69)	58.14(0.67)	61.80(0.84)
TKDP− SK	55.87(3.64)	63.85(3.40)	71.64(1.20)	25.78(2.97)	32.21(0.86)	34.87(2.06)	53.04(1.43)	58.30(1.47)	62.03(0.66)
TKDP− SK,LK,CK	47.00(1.84)	55.22(4.53)	65.29(1.01)	14.41(4.16)	23.43(2.45)	30.20(0.78)	48.49(1.06)	53.94(0.56)	58.77(1.16)
	MIT-Restaurant			*JNLPBA*			*Avg. of All Dataset*		
TKDP	**47.44**(3.24)	**56.80**(0.97)	**61.76**(1.67)	**33.57**(1.36)	**42.44**(1.62)	**49.31**(1.45)	**43.43**	**51.23**	**56.61**
TKDP− CK	46.29(2.79)	54.54(0.87)	61.37(1.29)	31.10(2.04)	38.73(1.65)	46.80(1.91)	42.02	49.56	55.64
TKDP− LK	45.94(2.56)	55.14(1.02)	60.88(1.35)	30.55(2.86)	38.67(1.37)	46.70(2.03)	41.23	48.73	54.44
TKDP− SK	45.44(2.97)	54.86(2.93)	60.35(1.27)	32.54(2.13)	42.25(2.42)	48.62(1.54)	42.53	50.38	55.50
TKDP− SK,LK,CK	42.16(3.23)	51.65(1.01)	57.45(1.46)	26.36(2.04)	31.75(2.74)	38.35(1.55)	35.68	43.20	50.01

SK (Sememe Knowledge), LK (Label Knowledge) and CK (Context Knowledge). without all the knowledge, our TKDP is relegated to deep prompt tuning [18] (DPT).

knowledge mentioned in Section III-C is removed, which is implemented by only using P^x as Q^x mentioned in (8). Similarly, TKDP-LK means that label knowledge is removed by only utilizing P^c as Q^c mentioned in (8) while ignoring other parts.

The comparison results are shown in the Table V. From the averaged results, we see that our model performs the best in all settings when combining all three types of knowledge. In particular, with all this knowledge, the average result is 7.75% (43.43-35.68), 8.03% (51.23-43.20) and 6.60% (56.61-50.01) higher in 5-shot, 10-shot and 20-shot than the F1-value of deep prompt (DPT [17]) without any knowledge. Notably, our TKDP improves the DPT method on F1 by 11.53% (25.94-11.41) on WNUT-2017 in the 5-shot setting. In addition, under the 10-shot and 20-shot settings of the CoNLL-2003 dataset, the result of TKDP increase effect after removing context knowledge. This may be because we have integrated the entire context into the prompt without screening useful information, thus introducing some noise.

C. Comparison of Different Continuous Prompt Methods

First, we briefly introduce three continuous prompting methods of fusing knowledge.

- **Prompt-Tuning** is a method similar to Lester et al. [24], which inserts consecutive prompts in the input embedding sequence for tuning. In this experiment, we enrich the continuous prompt with three kinds of knowledge.
- **P-Tuning** is a method similar to Liu et al. [25], which still inserts continuous prompts in the input embedding sequence, but it uses BiLSTM [62] to associate prefixes and infixes. We first enrich this continuous prompt with three kinds of knowledge, and then associate the label-knowledge prompt and the context-knowledge prompt as prefixes and infixes.
- **Prefix-Tuning** is a deep prompt tuning method similar to Li and Liang [26]. It uses the same prefix and infix in each layer of GPT-2 [63] (transformed using MLP layers). We enrich the continuous prompt with three kinds of knowledge, and then insert this continuous prompt into each layer of the pretrained model.

In Table VI, the results of other continuous prompts compared with our method are reported, and we draw the following conclusions: 1) Our method performs the best under the three settings in all datasets, because we incorporate different knowledge in each layer of PLM. Specifically, in terms of average results, our method improves the results by 3.12%, 4.04%, and 4.42% under three settings, compared with the prompt-tuning method (Prefix-Tuning). 2) Prefix-Tuning is suboptimal in all cases, because it is also a deep prompt method. However, the knowledge contained in each layer is the same and limited. 3) The average performance of P-tuning is better than Prompt-tuning, indicating that the interaction of label knowledge and context knowledge can certain performance improvement. However, compared with the performance improvement by deep prompts, it is not significant.

D. Comparison of Continuous and Discrete Prompting Methods

We skip the construction of knowledge-enriched deep prompts in Fig. 2, and directly splice E^x and E^C into discrete deep prompts, which are integrated into different PLM layers for in-depth interaction. The experimental results are shown in Fig. 7. Without the prompt embedding layer, the performances of all three settings of the datasets drop significantly, because there are almost no trainable parameters in the whole model to fine-tune. The knowledge learned from the data is thus greatly reduced, leading to decreases in model performances.

E. Effect of Prompt Depth and Length on Model Performance

This experiment mainly explores the impact of two main parameters of prompts, namely the length of prompts and the depth of prompts, on the performance of the model. Different prompt lengths and prompt depths can bring more significant impact on the model.

TABLE VI
COMPARISON OF DIFFERENT CONTINUOUS PROMPTING METHODS

	5-Shot	10-Shot	20-Shot	5-Shot	10-Shot	20-Shot	5-Shot	10-Shot	20-Shot
	CoNLL-2003			*WNUT-2017*			*MIT-Movie*		
• **Ours**	**56.72**(3.54)	**64.87**(3.27)	**73.43**(0.69)	**25.94**(3.31)	**32.48**(1.07)	**35.69**(1.69)	**53.46**(0.71)	**59.56**(0.84)	**62.87**(0.73)
• **Prompt-Tuning**	40.80(5.37)	53.85(0.83)	59.95(1.65)	18.10(3.69)	22.10(2.49)	25.91(1.85)	45.23(2.06)	51.56(1.77)	57.28(1.06)
• **P-Tuning**	46.92(5.23)	56.16(2.32)	63.53(1.61)	17.01(4.22)	23.94(2.32)	25.46(2.79)	46.48(1.31)	51.41(1.59)	56.84(1.15)
• **Prefix-Tuning**	53.89(4.42))	60.23(3.07)	68.44(1.73)	23.82(3.53)	29.38(2.66)	33.35(1.79)	50.87(1.26)	55.96(0.70)	59.51(0.47)
	MIT-Restaurant			*JNLPBA*			*Avg. of All Dataset*		
• **Ours**	**47.44**(3.24)	**56.80**(0.97)	**61.76**(1.67)	**33.57**(1.36)	**42.44**(1.62)	**49.31**(1.45)	**43.43**	**51.23**	**56.61**
• **Prompt-Tuning**	40.27(3.02)	47.85(3.92)	52.73(1.71)	25.86(3.15)	28.90(2.33)	38.21(2.83)	34.05	40.85	46.82
• **P-Tuning**	42.57(1.19)	51.32(0.87)	57.47(1.12)	26.85(2.42)	30.75(2.69)	38.93(2.99)	35.96	42.69	48.45
• **Prefix-Tuning**	44.11(3.63)	52.84(1.82)	57.69(1.58)	28.84(2.45)	37.55(1.75)	41.98(1.84)	40.31	47.19	52.19

The prompt embeddings of all methods in this table are combined with sememes, labels and context knowledge for fair comparison.

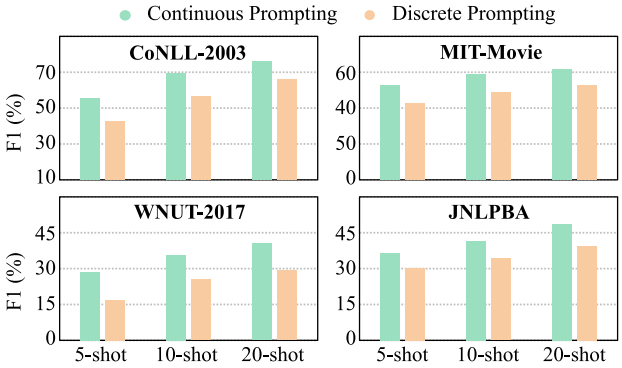


Fig. 7. Comparison of continuous and discrete prompting.

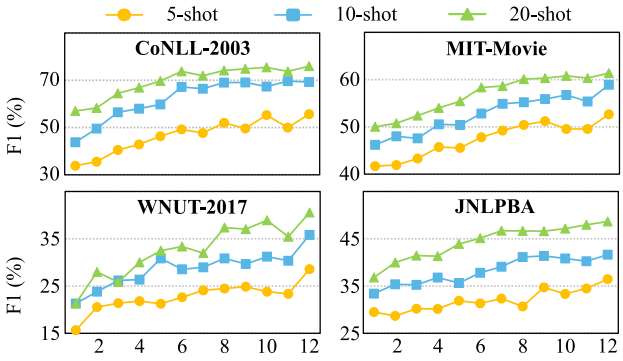


Fig. 8. Effect of prompt depth n_p on model performance.

1) Effect of Prompt Depth on Model Performance: In Fig. 8, we explore the effect of different depths (n_p) of prompt embeddings on four datasets and three settings. It is clear from the trends that the model performance tends to grow as the embedding depth increases. This is reasonable because more embedding layers bring more knowledge carrying capacity. Therefore, we generally set n_p=12 in our experiments.

2) Effect of Prompt Length on Model Performance: In each setting of other experiments, we tune parameters for five training samples separately, which makes different optimal prompt lengths for different samples. However, in this experiment, the prompt length will be fixed when tuning parameters for the five training samples separately. We still conduct experiments under three settings on four datasets, and the experimental results are shown in Fig. 9.

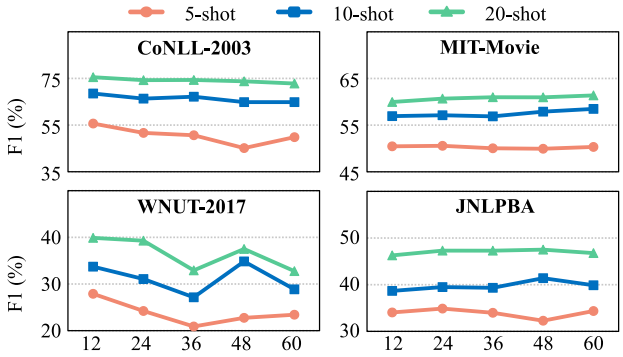


Fig. 9. Effect of prompt length l_p on model performance.

It can be seen from the figure that different datasets have different sensitivities to the prompt length under different settings: (1) For the CoNLL-2003 and MIT-Movie datasets, the model performance does not change significantly with the change of the prompt length. However, there is a slight upward trend on 20-shot and 10-shot, which may be more longer prompts can incorporate more knowledge. (2) For WNUT-2017 and JNLPBA, the performance of the model is greatly affected by the prompt length.

F. Case Study

To help directly understand how our method manages injected knowledge and captures useful features from knowledge augmentation prompts, we perform a case study through visualization. Fig. 10 shows the attention scores between the input and the prompts in the last layer of PLM. First, we can see that our knowledge-enriched prompts learn to attend to different input words, suggesting that the model can indeed capture useful information. Conversely, the prompts without knowledge do not provide any help for semantic reasoning. Second, we notice that the input sentence pays more attention to the second half of the prompt (context knowledge) rather than the first half (label knowledge). This also makes sense, since contextual features are more useful than label features in this case. Third, with the

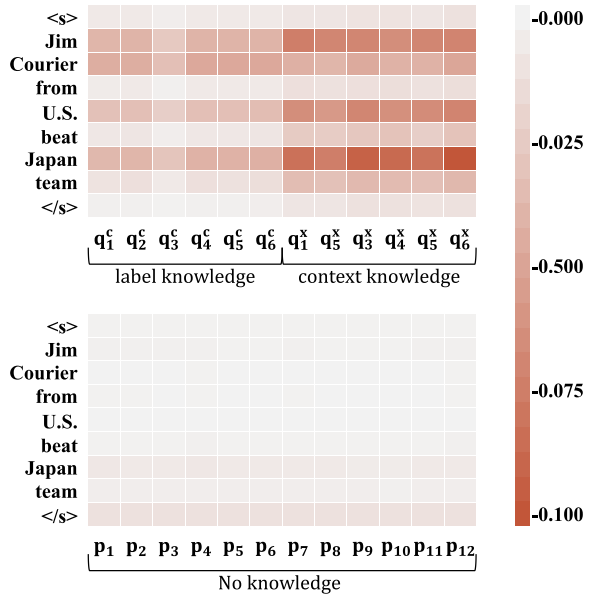


Fig. 10. Visualization of attention weights between the input and knowledge embeddings when using and abandoning knowledge.

TABLE VII
COMPARISONS OF TRAINING AND INFERENCE EFFICIENCIES OF OUR MODAL AND BASELINES

Model	Training sent/s	Inference sent/s
StructShot	70.29	231.65
Prototype	25.08	512.31
EntLM	3.60	503.85
TemplateNER	74.63	0.44
SEE-Few	6.80	54.99
FFF-NER	3.07	10.56
TKDP	160.75	365.94

"Sent/s" represents the number of sentences processed per second.

help of knowledge-enriched prompts, entity mentions such as "Jim Courier", "U.S" and "Japan" are highly weighted and thus help to give correct predictions.

G. Efficiency Analysis

We tested the training and inference efficiency of our model and baselines on the CoNLL-2003 dataset, and the results are shown in Table VII. During the training stage, our method is the fastest compared to other methods, which is reasonable because the purpose of the soft prompt method is to improve the training efficiency and meanwhile maintain the similar performance with fully fine-tuning methods. By contrast, FFF-NER is the slowest during training because it needs to train on a large number of data samples.

In terms of inference, the speed of our method is not optimal, which is slower than that of Prototype or EntLM. This is mainly because our method requires the calculation of additional soft prompts, while Prototype and EntLM do not. TemplateNER and FFF-NER are the slowest because they need to enumerate all substancial spans.

VI. CONCLUSION

We propose a threefold knowledge-enriched deep prompt tuning to achieve effective and fast few-shot NER. We incorporate sememe knowledge, label knowledge and context knowledge into the deep prompts. We conduct extensive experiments on five datasets. The results show that our method outperforms strong-performing baseline models on there few-shot settings. Ablation experiments show that it is effective to integrate three types of knowledge into deep prompts, and the performance can be boosted significantly compared with the deep prompt method without knowledge. In addition, we found that our method shows superior than existing prompt-based methods in terms of performance and efficiency. We hope this work can contribute a valuable reference for few-shot NER and other tasks.

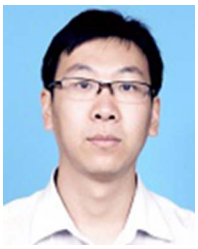
REFERENCES

[1] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.
[2] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 50–70, Jan. 2022.
[3] J. Guo, G. Xu, X. Cheng, and H. Li, "Named entity recognition in query," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2009, pp. 267–274.
[4] Y. S. Chan and D. Roth, "Exploiting syntactico-semantic structures for relation extraction," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics Hum. Lang. Technol.*, 2011, pp. 551–560.
[5] J. Yu, B. Bohnet, and M. Poesio, "Named entity recognition as dependency parsing," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6470–6476.
[6] L. Cui and Y. Zhang, "Hierarchically-refined label attention network for sequence labeling," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 4115–4128.
[7] J. Huang et al., "Few-shot named entity recognition: An empirical baseline study," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 10 408–10 423.
[8] Y. Yang and A. Katiyar, "Simple and effective few-shot named entity recognition with structured nearest neighbor learning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 6365–6375.
[9] B. Ji et al., "Few-shot named entity recognition with entity-level prototypical network enhanced by dispersedly distributed prototypes," in *In Proc. 29th Int. Conf. Comput. Linguistics*, 2022, pp. 1842–1854.
[10] Z. Yang, L. Zhang, and D. Zhou, "See-few: Seed, expand and entail for few-shot named entity recognition," in *Proc. 29th Int. Conf. Comput. Linguistics*, 2022, pp. 2540–2550.
[11] L. Cui, Y. Wu, J. Liu, S. Yang, and Y. Zhang, "Template-based named entity recognition using BART," in *Proc. Assoc. Comput. Linguistics Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 1835–1845.
[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol.*, 2019, pp. 4171–4186.
[13] Y. Liu et al., "RoBERTa: A robustly optimized bert pretraining approach," 2019, *arXiv: 1907.11692*.
[14] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.
[15] C. Wang, J. Wang, M. Qiu, J. Huang, and M. Gao, "TransPrompt: Towards an automatic transferable prompting framework for few-shot text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 2792–2802.
[16] Y. Sun, Y. Zheng, C. Hao, and H. Qiu, "NSP-BERT: A prompt-based few-shot learner through an original pre-training task — next sentence prediction," in *Proc. 29th Int. Conf. Comput. Linguistics*, 2022, pp. 3233–3250.

[17] X. Liu et al., "P-Tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 61–68.
[18] X. Chen et al., "LightNER: A lightweight tuning paradigm for low-resource NER via pluggable prompting," in *Proc. 29th Int. Conf. Comput. Linguistics*, 2022, pp. 2374–2387.
[19] D. Seyler, T. Dembelova, L. Del Corro, J. Hoffart, and G. Weikum, "A study of the importance of external knowledge in the named entity recognition task," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 241–246.
[20] D. Sui, Y. Chen, K. Liu, J. Zhao, and S. Liu, "Leverage lexical knowledge for Chinese named entity recognition via collaborative graph network," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 3830–3840.
[21] J. Chen, Q. Liu, H. Lin, X. Han, and L. Sun, "Few-shot named entity recognition with self-describing networks," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 5711–5722.
[22] L. Bloomfield, "A set of postulates for the science of language," *Language*, vol. 2, no. 3, pp. 153–164, 1926.
[23] F. Qi, L. Chang, M. Sun, S. Ouyang, and Z. Liu, "Towards building a multilingual sememe knowledge base: Predicting sememes for babelnet synsets," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 8624–8631.
[24] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 3045–3059.
[25] X. Liu et al., "GPT understands, too," *AI Open*, 2023.
[26] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 4582–4597.
[27] R. Ma et al., "Template-free prompt tuning for few-shot NER," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2022, pp. 5721–5732.
[28] S. S. S. Das, A. Katiyar, R. J. Passonneau, and R. Zhang, "CONTaiNER: Few-shot named entity recognition via contrastive learning," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 6338–6353.
[29] Y. Hou et al., "Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 1381–1393.
[30] J. Ma et al., "Label semantics for few shot named entity recognition," in *Proc. Assoc. Comput. Linguistics*, 2022, pp. 1956–1971.
[31] T. B. Brown et al., "Language models are few-shot learners," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 1877–1901.
[32] D.-H. Lee et al., "Good examples make a faster learner: Simple demonstration-based learning for low-resource NER," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 2687–2700.
[33] F. Qi, C. Yang, Z. Liu, Q. Dong, M. Sun, and Z. Dong, "OpenHowNet: An open sememe-based lexical knowledge base," 2019, *arXiv: 1901.09957*.
[34] Z. Dong and Q. Dong, *HowNet and the Computation of Meaning: (With CD-ROM)*. Singapore: World Scientific, 2016.
[35] Q. Liu, "Word similarity computing based on HowNet," *Comput. Linguistics Chin. Lang. Process.*, vol. 7, no. 2, pp. 59–76, 2002.
[36] B. Hou, F. Qi, Y. Zang, X. Zhang, Z. Liu, and M. Sun, "Try to substitute: An unsupervised Chinese word sense disambiguation method based on HowNet," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 1752–1757.
[37] P. Li and G. Zhou, "Employing morphological structures and sememes for Chinese event extraction," in *Proc. 24th Int. Conf. Comput. Linguistics*, 2012, pp. 1619–1634.
[38] Y. Niu, R. Xie, Z. Liu, and M. Sun, "Improved word representation learning with sememes," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 2049–2058.
[39] Y. Gu et al., "Language modeling with sparse product of sememe experts," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4642–4651.
[40] X. Zeng, C. Yang, C. Tu, Z. Liu, and M. Sun, "Chinese LIWC lexicon expansion via hierarchical classification of word embeddings with sememe attention," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 5650–5657.
[41] Z. Li, N. Ding, Z. Liu, H. Zheng, and Y. Shen, "Chinese relation extraction with multi-grained information and external linguistic knowledge," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 4377–4386.
[42] S. Liu, J. Xu, and X. Ren, "Evaluating semantic rationality of a sentence: A sememe-word-matching neural network based on HowNet," in *Proc. CCF Int. Conf. Natural Lang. Process. Chin. Comput.*, 2019, pp. 787–800.
[43] H. Jin et al., "Incorporating chinese characters of words for lexical sememe prediction," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 2439–2449.
[44] F. Qi, Y. Lin, M. Sun, H. Zhu, R. Xie, and Z. Liu, "Cross-lingual lexical sememe prediction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 358–368.
[45] F. Qi et al., "Modeling semantic compositionality with sememe knowledge," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5706–5715.
[46] B. Lyu, L. Chen, and K. Yu, "Glyph enhanced Chinese character pre-training for lexical sememe prediction," in *Proc. Assoc. Comput. Linguistics, Conf. Empir. Methods Natural Lang. Process.*, 2021, pp. 4549–4555.
[47] F. Qi, C. Lv, Z. Liu, X. Meng, M. Sun, and H.-T. Zheng, "Sememe prediction for babelnet synsets using multilingual and multimodal information," in *Proc. Assoc. Comput. Linguistics*, 2022, pp. 158–168.
[48] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "Autoprompt: Eliciting knowledge from language models with automatically generated prompts," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 4222–4235.
[49] S. Hu et al., "Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 2225–2240.
[50] X. Chen et al., "KnowPrompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction," in *Proc. ACM Web Conf.*, 2022, pp. 2778–2788.
[51] H. Li et al., "KiPT: Knowledge-injected prompt tuning for event detection," in *Proc. 29th Int. Conf. Comput. Linguistics*, 2022, pp. 1943–1952.
[52] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–14.
[53] E. T. K. Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proc. 7th Conf. Natural Lang. Learn.*, 2003, pp. 142–147.
[54] L. Derczynski, E. Nichols, M. van Erp, and N. Limsopatham, "Results of the WNUT2017 shared task on novel and emerging entity recognition," in *Proc. 3rd Workshop Noisy User-Generated Text*, 2017, pp. 140–147.
[55] J. Liu, P. Pasupat, Y. Wang, S. Cyphers, and J. Glass, "Query understanding enhanced by hierarchical parsing structures," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2013, pp. 72–77.
[56] J. Liu, P. Pasupat, S. Cyphers, and J. Glass, "ASGARD: A portable architecture for multilingual dialogue systems," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013, pp. 8386–8390.
[57] N. Collier and J.-D. Kim, "Introduction to the bio-entity recognition task at JNLPBA," in *Proc. Int. Joint Workshop Natural Lang. Process. Biomed. Appl.*, 2004, pp. 73–78.
[58] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4080–4090.
[59] Z. Wang, K. Zhao, Z. Wang, and J. Shang, "Formulating few-shot fine-tuning towards language model pre-training: A pilot study on named entity recognition," in *Proc. Assoc. Comput. Linguistics, Empir. Methods Natural Lang. Process.*, 2022, pp. 3186–3199.
[60] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," 2023, *arXiv:2307.09288*.
[61] Y. Huang et al., "COPNER: Contrastive learning with prompt guiding for few-shot named entity recognition," in *Proc. 29th Int. Conf. Comput. Linguistics*, 2022, pp. 2515–2527.
[62] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 260–270.
[63] A. Radford et al., "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019, Art. no. 9.

Jiang Liu received the BE degree from Zhengzhou University, China, in 2020, and the ME degree from Wuhan University, China, in 2023, where he is currently working toward the PhD degree with the School of Cyber Science and Engineering. He has been working on Natural Language Processing (NLP).

Hao Fei received the PhD degree from Wuhan University, China, in 2021. He is currently a research fellow of National University of Singapore, and also an associate researcher with Skywork AI. His research interests cover natural language processing (NLP) and Multimodal Learning, with specific interests in structural learning and LLMs. His research papers have been published at top-tier venues, e.g., ICML, NeurIPS, ACL, ACM MM, AAAI, IJCAI, EMNLP, *ACM Transactions on Information Systems*, *IEEE Transactions on Neural Networks and Learning Systems*. He was honored as a Rising Star at the 2023 World Artificial Intelligence Conference Yunfan Award. He has been the organizing committee, area chair and senior program committee of top-tier conferences, including ICML, ACL, EMNLP, AAAI, IJCAI, WSDM, COLING and NAACL, etc. Also he serves as the associate editor of *ACM Transactions on Asian and Low-Resource Language Information Processing* and *Neurocomputing*.



Fei Li is the associate research professor and doctoral & master supervisor of Wuhan University, the member of Chinese Information Processing Society of China and Chinese Association for Artificial Intelligence. He is mainly engaged in the research of natural language processing, has published more than 70 papers, and served as a reviewer or area chair of ACL, EMNLP, COLING, NAACL, AAAI, etc.

Jingye Li received the BE and ME degrees from Wuhan University, China, in 2019 and 2022, respectively. He has been working on Sentiment Analysis and Information Extraction. He has published related papers in top international conferences such as AAAI, ACL and COLING.

Bobo Li received the ME degree from Wuhan University in 2019. He is currently working toward the PhD degree with the School of Cyber Science and Engineering of Wuhan University. His research mainly focuses on natural language processing and deep learning.

Liang Zhao received the BS degree in computer science from Wuhan University, Wuhan, China, in 1988, and the MSc and PhD degrees in computer science from the Aeronautic Institute of Technology, in Brazil, in 1996 and 1998, respectively. He joined the University of Sao Paulo (USP), Brazil, in 2000, where he is currently a full professor with the Department of Computing and Mathematics of the Faculty of Philosophy, Science, and Letters at Ribeirao Preto (FFCLRP) of USP. His current research interests include machine learning, graph neural networks, complex networks, and pattern recognition. He is a recipient of the Brazilian Research Productivity Fellowship. He was an associate editor of *IEEE Transactions on Neural Networks and Learning Systems* from 2009 to 2012. He is currently an associate editor of *Neural Networks*.

Chong Teng is an associate professor and graduate supervisor with Wuhan University. Her research interests cover natural language processing (NLP), sentiment analysis.

Donghong Ji received the PhD degrees in computer science and linguistics from Wuhan University and the University of Oxford respectively. He is distinguished professor with Wuhan University. His research interests cover artificial intelligence (AI), natural language processing (NLP), deep learning, cognitive computing, etc. He has made outstanding contributions in text entailment reasoning, discourse semantics understanding, lexical semantics, and other areas. In the past five years, he has published more than 100 academic papers indexed by SCI and EI in prestigious domestic and international academic journals and conferences. Among them, more than 30 papers have been published in top conferences and journals in natural language processing and knowledge mining, such as ACL, EMNLP, WWW, SIGIR, IJCAI, AAAI, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE/ACM Transactions on Audio, Speech and Language Processing*, etc. He has been invited to serve as a reviewer for international prestigious journals and conferences including ACL, EMNLP, AAAI, IJCAI, SIGIR, WWW, COLING, IEEE/ACM Transactions on Audio, Speech and Language Processing, *Information Processing and Management*, etc.