Front. Comput. Sci., 2025, 19(6): 196317
https://doi.org/10.1007/s11704-024-3900-z

RESEARCH ARTICLE

ProSyno: context-free prompt learning for synonym discovery

Song ZHANG[1,2], Lei HE[3], Dong WANG[3], Hongyun BAO[1], Suncong ZHENG[3], Yuqiao LIU[1,2], Baihua XIAO[1], Jiayue LI (✉)[5], Dongyuan LU (✉)[4], Nan ZHENG (✉)[1,2]

1 State Key Laboratory of Multimodal Artifcial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
2 School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100190, China
3 Tencent AI Platform Department, Beijing 100048, China
4 University of International Business and Economics, Beijing 100029, China
5 Beijing Academy of Blockchain and Edge Computing, Beijing 100085, China

Abstract Synonym discovery is important in a wide variety of concept-related tasks, such as entity/concept mining and industrial knowledge graph (KG) construction. It intends to determine whether two terms refer to the same concept in semantics. Existing methods rely on contexts or KGs. However, these methods are often impractical in some cases where contexts or KGs are not available. Therefore, this paper proposes a context-free prompt learning based synonym discovery method called ProSyno, which takes the world's largest freely available dictionary Wiktionary as a semantic source. Based on a pre-trained language model (PLM), we employ a prompt learning method to generalize to other datasets without any fine-tuning. Thus, our model is more appropriate for context-free situation and can be easily transferred to other fields. Experimental results demonstrate its superiority comparing with state-of-the-art methods.

Keywords synonym discovery, prompt learning, large language model

1 Introduction

Synonym discovery is a critical task of information extraction which intends to identify whether the target term and the candidates in the text corpus are the same or similar in semantics. This task is widely used in a great number of domains, such as entity/concept mining, KG construction and recommendations [1–3].

Existing deep learning methods can be classified into two categories: semantic enhanced methods [4,5] and graph-based methods [6,7]. Semantic enhanced methods mainly employ robust contextual embeddings represented by PLMs which contain rich semantic information of target terms [4,5,8]. Graph-based methods make predictions on the hypothesis that synonym concept terms have similar neighbors [6,7]. Although effective, these methods show the following shortcomings: 1) It's easy to make false positive predictions just relying on pre-trained embeddings. Because some correlated while non-synonymous terms (like antonyms) often share the same or similar contexts, leading to results that pre-training embeddings of correlated terms tend to be similar and are hard to make a distinction; 2) KGs [9–11] and context [12,13] may not be available in some domains, hindering models from generalizing to these fields.

In this paper, we propose a context-free prompt learning model, named ProSyno. To address two aforementioned challenges, domain-independent word descriptions in Wiktionary[1)] are introduced into ProSyno as a semantic source. The rationale is twofold: 1) word descriptions in Wiktionary contain informative semantics which are beneficial to distinguishing highly correlated term pairs; 2) Wiktionary is the world's largest freely available dictionary. Its large coverage ensures our model's capacity of transferring to various domains. Figure 1 depicts an example which shows that a word description helps to distinguish synonym. The first description of "*crabby*" consists of the word "*irritable*" that is highly correlated to the target term "*feeling irritable*", thus synonym relation between the term pair can be discovered easily. Specifically, a hierarchical semantic encoder is designed to extract semantic representations of words. However, there usually exist several descriptions of a target word in Wiktionary. To obtain informative word representations from multiple descriptions, a dynamical matching mechanism is designed to weigh each description and then each description of the word is fused by its corresponding matching degree. To transfer knowledge from foundation model to synonym detection task, we employ a prompt learning method to train our model. Prompt learning makes synonym discovery task cord with pre-training task by

Received November 14, 2023; accepted May 10, 2024

E-mail: lijy@baec.org.cn; ludy@uibe.edu.cn; nan.zheng@ia.ac.cn

1) See wiktionary.org website.

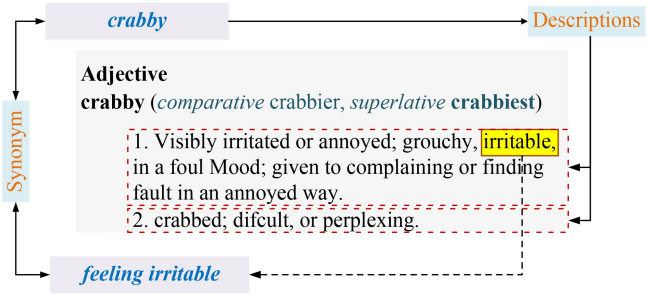


Fig. 1 A sample which shows that a word description in Wiktionary helps to distinguish synonym

converting inputs into an ordered sequence that PLM can process. This enables our model can better leverage learned knowledge from large-scale dataset.

In summary, this work makes the following contributions:

- Descriptions in the largest freely available dictionary, i.e., Wiktionary, are integrated, so that our model can mitigate the semantic gap between term pairs and get rid of the dependency on contexts and KGs.
- To dynamically obtain a highly informative representation from multiple descriptions of a word, a dynamical matching mechanism is designed to fuse them through the matching degree with the candidate term.
- To the best of our knowledge, this paper is the first try to introduce the idea of prompting into the context-free synonym discovery task, which enable our model to generalize to other datasets without any annotated data. Experimental results on four benchmarks demonstrate the effectiveness of ProSyno.

The rest of the paper are organized as follows. Literature review is concluded in Section 2. Detailed introduction of Wiktionary and formal definition of the problem are given in Section 3. ProSyno is described in Section 4 and experimental results are represented in Section 5. Finally, we conclude in Section 6.

2 Related work

2.1 Synonym discovery

Given a corpus and a term list, one can leverage surface string, co-occurrence statistics, textual pattern, distributional similarity, or their combinations to extract synonyms [14,15]. Two most commonly used knowledge-intensive synonyms discovery tools, MetaMap[2)] and cTAKES[3)] both employed rules to first generate lexical variants for each noun phrase and then conducted dictionary look-up for each variant. However, such rule-based approaches struggle when there are many variations among concept terms or no any necessary contexts, which is common, when referring user-generated query texts to product descriptions.

Recently, many deep learning methods have been proposed, which can be divided into two categories. **Context-based methods** hypothesize the meaning of a term mention can be reflected by its neighboring words, so contexts of target terms are employed to obtain robust representatons [16–18]. For example, SynonymNet [5] proposed a multi-context bilateral matching framework for synonym discovery from a free-text corpus. SurfCon [10] discovered synonyms on privacy-aware clinical data by utilizing the surface form information and the global context information. **KG-based methods** assume that synonyms have similar linking relationships, so KGs are taken as external knowledge to make predictions. For example, SynSetMine [19] learned a set-instance classifier to generate synonym sets from a given vocabulary using example sets from external knowledge bases as distant supervision. CODER [7] learned synonyms knowledge from the Unified Medical Language System (UMLS) to provide close embeddings for synonyms. However, it is usually difficult to access to large-scale raw contexts like query records and clinical data. Moreover, KG is often unavailable and need to be built, which is time consuming. Thus, the aforementioned methods might be ineffective.

2.2 Prompt learning

Recently, various PLMs have been proposed and applied in a great number of fields, such as such as GPT [20] and and BERT [21]. To make PLMs apply to downstream tasks, task-oriented fine-tuning has been proposed [22]. However, this paradigm requires to annotate sufficient supervised data for a great number of tasks, which takes time and effort. Instead of learning a new LM for each downstream task, prompt-based methods employ PLM to make predictions without additional training stage by reformulating downstream tasks as a language modeling problem to mitigate the gap between pre-training and downstream tasks [23]. Discovering the appropriate prompt is central to this line of works. Preliminary works elaborately design human-crafted prompts, which is known as prompt engineering. Since the manual design is sensitive and difficult, a series of approaches focus on automatically generating desired (discrete) prompts in the natural language space. Recently, some works [24,25], also known as prompt tuning, attempt to learn soft (continuous) prompts directly instead of searching for discrete prompts.

While prompt learning achieves excellent performance on many NLP tasks, it remains to be explored in the synonym discovery task. Thus, we exploit prompt learning by providing a task-oriented prompt with PLM, which enables PLM to understand the synonym discovery task. Further, under the configuration of missing contexts and KGs, it's insufficient for PLM to do synonym discovery by training a continuous task-specific prompts. Therefore, we introduce semantic prompts by employing term descriptions in Wiktionary to provide semantic information.

3 Preliminaries

3.1 Wiktionary as semantic source

Wiktionary is the world's largest freely available dictionary, which is a collaboratively edited, multilingual online

[2)] See lhncbc.nlm.nih.gov/ii/tools/MetaMap.html website.
[3)] See ohnlp.org website.

dictionary [26,27]. The main advantage is that it is open-source, multilingual and has a good coverage. Therefore, its large coverage ensures our model's capacity of transferring to various domains. In addition, several studies already showed its usefulness for various NLP tasks [26,28]. In Wiktionary, each entry consists of a definition with one or several descriptions and examples. As shown in Fig. 1, the definition of "*crabby*" consists of part-of-speech and two descriptions. In this paper, ProSyno employs entry's descriptions to mitigate semantic information deficiency issues resulted by missing context.

3.2 Problem statement

A term $t^p = \left[s_1^{t^p}, s_2^{t^p}, \ldots, s_{|t^p|}^{t^p}\right]$ is a string (i.e., a word or a phrase), like "*crabby*" and "*felling irritable*". The ith word $s_i^{t^p}$ of t^p has multiple descriptions in Wiktionary, denoted as $c_{i:}^{t^p} = \left[c_{i1}^{t^p}, c_{i2}^{t^p}, \ldots, c_{i|c_{i:}^{t^p}|}^{t^p}\right]$. The jth description of $s_i^{t^p}$ in t^p is also a string, denoted as $c_{ij}^{t^p} = \left[s_1^{p_{ij}}, s_2^{p_{ij}}, \ldots, s_{|c_i^{t^p}|}^{p_{ij}}\right]$. Given a pair of terms t^p and t^q with their corresponding descriptions inWiktionary, $[c_{1:}^{t^p}, c_{2:}^{t^p}, \ldots, c_{|t^p|:}^{t^p}]$ and $[c_{1:}^{t^q}, c_{2:}^{t^q}, \ldots, c_{|t^q|:}^{t^q}]$, we aim to determine whether two terms refer to the same concept in semantics. A classifying function that maps a term pair to a probability $f: (t_p, t_q) \rightarrow p(y|t_p, t_q)$, where $y \in \mathcal{Y}$ is the classification label, $\mathcal{Y} = \{0, 1\}$.

4 Methodology

Figure 2(a) shows the architecture of ProSyno, which consists of a hierarchical semantic encoder and a pattern mapper. Hierarchical semantic encoder encodes word descriptions in Wiktionary to obtain semantic representations of target terms. Pattern mapper aims to exploit large PLMs to determine synonym relations between the concept term pair by wrapping the term pair and their semantic representations into an ordered sequence that PLM can process. Below, we introduce ProSyno in details.

4.1 Hierarchical semantic encoder

To provide necessary semantic information, we design a hierarchical semantic encoder (Fig. 2(b)) consisting of a description encoder, a descriptions aggregator and a term encoder. All learnable parameters of the hierarchical semantic encoder are denoted as θ_s.

4.1.1 Description encoder

ProSyno employs Transformer encoder [29] as the description encoder. The basic concept of a transformer encoder is to utilize self-attention, which allows the encoder to focus on different parts of the sequence, modeling both short-term and long-term dependencies effectively. Self-attention is defined as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{1}$$

where d_k is the scaling factor, Q,K and V are linear transformations from the same input hidden representation M^l. MultiHead attention (MHA) is a concatenation of multiple self-attention components. Specifically, let $M^l = [e_1^{t^q}, e_2^{t^q}, \ldots, e_{|t^q|}^{t^q}]$ denote the input representation of the $(l+1)$th Transformer layer. M^0 is set to be the input of the encoder. Given an input representation M^l (where $1 \leqslant l \leqslant L$, L denotes the number of encoder layers).

$$\text{MultiHead}(M^l) = \text{Concat}(head_1, \ldots, head_h)\, W^O, \tag{2}$$

where

$$head_i = \text{Attention}(M^l W_i^Q, M^l W_i^K, M^l W_i^V), \tag{3}$$

where h is the number of heads, $W_i^Q \in \mathbb{R}^{d \times \frac{d}{h}}$, $W_i^K \in \mathbb{R}^{d \times \frac{d}{h}}$, $W_i^V \in \mathbb{R}^{d \times \frac{d}{h}}$, $W^O \in \mathbb{R}^{d \times d}$ are trainable parameters. Using MHA, the transformer encoder is constructed.

Given the jth description $c_{ij}^{t^p}$ of the ith word $s_i^{t^p}$ in term t^p, its final representation encoded by the transformer encoder is

$$M_{ij}^p = \text{MultiHead}\left(c_{ij}^{t^p}\right). \tag{4}$$

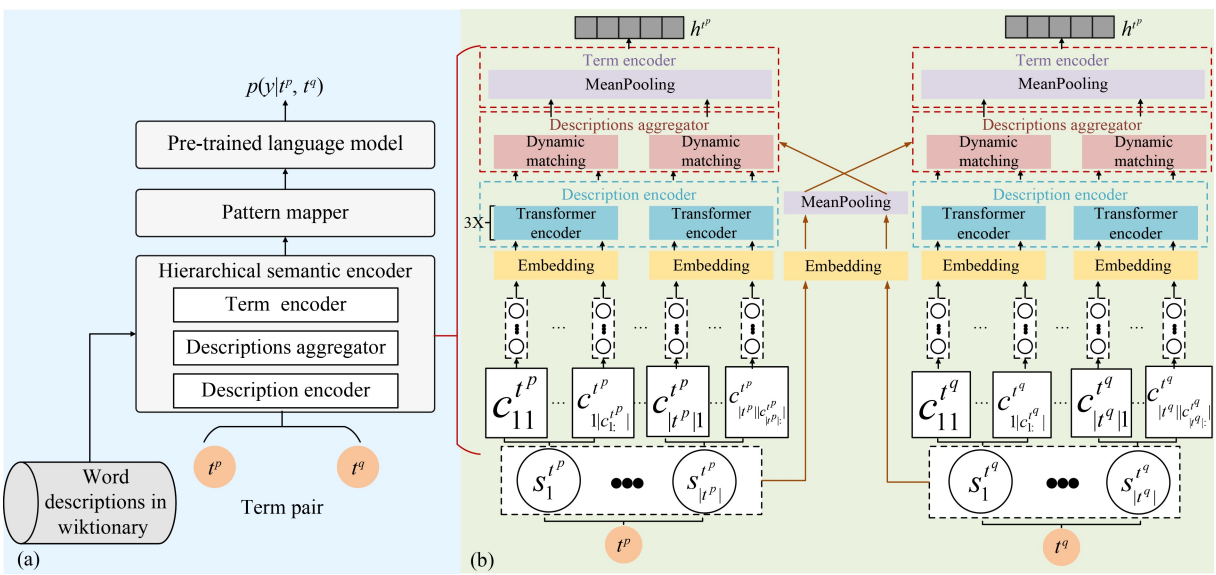


Fig. 2 (a) Overview of ProSyno; (b) hierarchical semantic encoder

Then, we use the mean-pooling to get the representation of the jth description of word $s_i^{t^p}$.

$$v_{ij}^p = \mathrm{MP}\left(M_{ij}^p\right). \tag{5}$$

Given all descriptions of the ith word $s_i^{t^p}$ in term t^p, we can obtain all the description representations of word $s_i^{t^p}$, denoted as $v_{i:}^p = \left[v_{i1}^p, v_{i2}^p, \ldots, v_{i|v_{i:}^p|}^p\right]$.

4.1.2 Descriptions aggregator

As we aim to capture the semantic meaning of the ith word $s_i^{t^p}$ in term t^p, the semantic vector of word $s_i^{t^p}$ is expected to contain all possible valuable semantic information. One naive way is to average all description representations $v_{i:}^p$. Since such representation does not depend on the other term candidate, we refer it as "static" representation.

In contrast to the static approach, we propose a dynamic matching mechanism, which weighs each description based on its matching degree with the candidate term and hence the semantic representation is dynamically changing depending on which term it is comparing with.

Given the candidate term $t^q = \left[s_1^{t^q}, s_2^{t^q}, \ldots, s_{|t^q|}^{t^q}\right]$, we convert each word $s_i^{t^q}$ in t^q to its d-dimensional vector e via an embedding matrix $E_s \in \mathbb{R}^d$, which provided by PLM. Then, the matching degree of $c_{ij}^{t^p}$ with t^q can be calculated by

$$\alpha_{ij}^{pq} = \frac{\left(\mathrm{MP}\left[e_{s_1^{t^q}}, \ldots, e_{s_{|t^q|}^{t^q}}\right]\right)^T W v_{ij}^p)}{\sum\limits_{1 \leqslant j' \leqslant |c_{i:}^{t^p}|} \left(\mathrm{MP}\left[e_{s_1^{t^q}}, \ldots, e_{s_{|t^q|}^{t^q}}\right]\right)^T W v_{ij'}^p}, \tag{6}$$

where $W \in \mathbb{R}^d$ is the learnable parameter. In this case, α_{ij}^{pq} is dependent on the candidate term pair.

Finally, the semantic vector for the ith word $s_i^{t^p}$ in term t^p is calculated through a weighted combination of its description:

$$h_i^{t^p} = \sum_{1 \leqslant j \leqslant |c_{i:}^{t^p}|} \alpha_{ij}^{pq} v_{ij}^p. \tag{7}$$

4.1.3 Term encoder

Given a term pair t^p and t^q, we employ mean-pooling to obtain the semantic representations of the target terms:

$$h^{t^p} = \mathrm{MP}\left[h_1^{t^p}, h_2^{t^p}, \ldots, h_{|t^p|}^{t^p}\right], \tag{8}$$

$$h^{t^q} = \mathrm{MP}\left[h_1^{t^q}, h_2^{t^q}, \ldots, h_{|t^q|}^{t^q}\right]. \tag{9}$$

4.2 Pattern mapper

Since PLM is trained on contiguous sequences of text, some modifications are required to be adapted to synonym discovery task. We design a pattern mapper g to wrap the term pair and their semantic representations into an ordered sequence that PLM can process.

$$\begin{aligned} x_{pq} &= g(t^p, t^q, h^{t^p}, h^{t^q}) \\ &= e_{[\mathrm{CLS}]} : e_{[\mathrm{TP}]} : e_{sem} : e_{[\mathrm{SEP}]} : e_{t^p} : e_{[\mathrm{SEP}]} \\ &\quad : e_{t^q}, \end{aligned} \tag{10}$$

where $e_{[\mathrm{CLS}]}$, e_{t^p}, $e_{[\mathrm{SEP}]}$ and e_{t^q} are embeddings of [CLS], t^p, [SEP] and t^q, respectively, which obtained by PLM, and $e_{[\mathrm{TP}]}$ is the learnable embedding of the task prompt token [TP], e_{sem} is the representation of the semantic prompt, which is computed by

$$e_{sem} = h^{t^p} : h^{t^q} : \left[h^{t^p} \odot h^{t^q}\right] : \left[h^{t^p} - h^{t^q}\right], \tag{11}$$

where $\odot$ denotes the element-wise product operation. Note that, $\left[h^{t^p} \odot h^{t^q}\right]$ and $\left[h^{t^p} - h^{t^q}\right]$ are designed to measure the closeness between the term pair in vector space.

4.3 Optimization

Since there is no inherent ordering of the two terms being compared, we first modify the input sequence to contain both possible sentence orders. And then, we process each one independently to produce two sequence representations. At last, they are added element-wise before being fed into the linear output layer. We adopt PLM f to map x_{pq} and x_{qp} into probability

$$h^{pq} = f(x_{pq}; \theta_f) + f(x_{qp}; \theta_f), \tag{12}$$

$$p(y|t^p, t^q) = \mathrm{SoftMax}(W h^{pq}), \tag{13}$$

where θ_f is paremeters of PLM, which is frozen in our model. Then, contrastive learning is adopted to optimize the parameters, which contrasts semantically similar (positive) and dissimilar (negative) pairs of data points. We employ hard soft margin loss:

$$\mathcal{L}(\theta) = \sum_{(p,q,q') \in T)} \ln(1 + e^{(p(y|t^p, t^{q'}) - p(y|t^p, t^q))}), \tag{14}$$

where T denotes the training set, in which each instance is a triplet (p, q, q') meaning t^q is a synonym of t^p while $t^{q'}$ is not, θ is all learnable parameters.

5 Experiments

5.1 Datasets

AskAPatient[4]**:** (AAP) contains 17,324 adverse drug reaction (ADR) annotations collected from blog posts. A total of 1,036 medical concepts with 22 semantic types have been mapped to 1,036 terms from the Systematized Nomenclature Of Medicine-Clinical Term subset of the Australian Medicines Terminology. We follow the 10-fold cross validation configuration.

TwADR-L[4]**:** contains 5,074 ADR expressions from social media. The terms are mapped to 2,220 Medical Dictionary for Regulatory Activities concepts with 18 semantic types. We follow the 10-fold cross validation configuration.

CADEC[5]**:** is the first richly annotated and publicly available corpus of medical forum posts taken from AAP. It

[4] See dx.doi.org/10.5281/zenodo.55013 website.
[5] See data.csiro.au/collection/csiro:10948v3 website.

contains 1253 user-generated texts about 12 drugs divided into two categories: Diclofenac and Lipitor. All posts are annotated manually for 5 types. There are 6,754 terms and 1,029 unique codes in total. We adopt the official training and test sets configuration: 5-fold cross validation configuration [9].

ANV[6]**:** is a domain-independent synonym dataset, which has 7816 synonym pairs. It was previously created from WordNet [30] and Wordnik[7]. The word pairs of synonyms were grouped according to the word class (Adjective, Noun and Verb). The dataset is split into training, validation and testing data the same as previous works.

5.2 Baselines

- **WordCNN**: [31] uses Convolutional Neural Networks over pre-trained word embeddings to generate the representation for each term, and then feeds them into a softmax layer for multi-class classification.
- **WordGRU**: [32] uses a bidirectional Gated Recurrent Units with attention over pre-trained embeddings to generate the representation for each term, and concatenates such representations with the cosine similarities of TF-IDF vectors, and then feeds the concatenated vector to a softmax layer for multi-class classification.
- **BERT**: [33] uses BERT [21] in a multi-class text-classification configuration as the candidate concept generator and a BERT-based list-wise classifier to select the most likely candidate.
- **BioBERT**: [34] is the most well-known biomedical language model. In this paper, following [33], we replace BERT by BioBERT to generate representations for each pair terms and make predictions.
- **CODER**: [7] is a medical term embedding model, which employs a KG-based contrastive learning framework to learn both term-term similarity and term-relation-term similarity.
- **MoE-ASD**: [35] proposes the mixture-of-experts framework named MoE-ASD for the discrimination between antonyms and synonyms. Specifically, MoE-ASD just leverages embedding features learned by FastText [36] and adopts a divide-and-conquer strategy to extract a few salient difference between term pairs in subspace, where each localized expert focuses on only a few salient dimensions. These salient dimensions may vary significantly throughout the whole distributional semantic space.

5.3 Implementation details

BioBERT-base[8] is used as ProSyno's backbone. Best hyper-parameters are selected based on the performance on dev set. We evaluate our model based on classification accuracy. WiktionaryParser[9] is used to fetch term description in Wiktionary. We sample 15, 9, 6 negative terms for each term pair on AAP, TwADR-L and CADEC, respectively. The number of transformer encoder layer is set to 2. The models are implemented in PyTorch. The standard evaluation of synonym detection task is accuracy [33]:

$$Accuracy = \frac{N_{corr}}{N_{all}}, \tag{15}$$

where N_{corr} is the amount of correct predictions, N_{all} is total amount of test data.

5.4 Main results

According to Table 1, ProSyno achieves a new state-of-the-art results on four datasets. We attribute such good performance to the following reasons: 1) ProSyno integrates word descriptions to mitigate semantic gaps between term pairs, which is essential for PLM to recognize some false positive term pairs; 2) Heirarchical semantic encoder enables our model to extract robust semantic representations of words by dynamically prioritizing informative ones among multiple descriptions; 3) Taking large models as backbone enables ProSyno to have the powerful inferring ability. It is notable that MoE-ASD achieves the best performance compared with knowledge-free models WordCNN and WordGRU, validating that distinguish term pairs in subspace is more effective. Because some high correlated while non-synonymous term pairs share similar embeddings which is distinctive in a few dimensions. Comparing with WordCNN and WordGRU, PLMs-based models achieve remarkable improvement, since PLMs contains extensive knowledge by trained on large scale raw text data. BioBERT outerperforms BERT, which shows that expert BERT is more appropriate for our datasets than general BERT. This is because BioBERT is pre-trained on biomedical domain corpora and contains significant biomedical knowledge.

5.5 Ablation studies

To further analyze the components of ProSyno, we design several ablation experiments.

- Prompts

By analyzing the benefits reaped by prompts, we examine five variants. 1) ProSyno-P: ProSyno without task and semantic prompts, 2) ProSyno-SP: ProSyno without the semantic prompt, 3) ProSyno-TP: ProSyno without the task prompt, 4) ProSyno-RTP takes real token "*synonym*" as the task prompt and frozen its parameters, 5) ProSyno-CAT takes

Table 1 Comparisons of ProSyno against state-of-the-art performances (%)

Datasets	AAP	TwADR-L	CADEC	ANV
WordCNN	81.41	44.78	–	–
WordGRU	85.71	–	–	–
BERT	87.46	47.02	–	–
BioBERT	88.39	48.32	–	–
CODER	–	–	59.01	–
MoE-ASD	87.30	47.65	58.24	92.16
ProSyno	**90.22**	**51.49**	**60.55**	**94.43**

[6] See github.com/Zengnan1997/MoE-ASD/tree/main/dataset website.
[7] See wordnik.com website.
[8] See github.com/dmis-lab/biobert website.
[9] See github.com/Suyash458/WiktionaryParser website.

Wiktionary descriptions as real prompts and concatenates them with its corresponding term. ProSyno-CAT is obtained by fine-tuning the task-oriented prompt $e_{[TP]}$.

As shown in Table 2, ProSyno-SP and ProSyno-TP surpass ProSyno-P, validating that both semantic and task prompts are significant for synonym discovery. Compared to ProSyno-TP, ProSyno-SP achieves worse performance, which indicates that the semantic prompt is more vital. ProSyno performs better than ProSyno-CAT, validating the effectiveness of the hierarchical semantic encoder. ProSyno-RTP underperforms ProSyno, might owing to continuous task prompts being more appliable.

● Hierarchical semantic encoder

We test ProSyno with different semantic encoders. 1) ProSyno-MLP takes the vector average of the embeddings, and puts it through 3-layer perceptrons to generate semantic representation e_s of word s. 2) ProSyno-BL takes BiLSTM to obtain semantic representation of each description. 3) ProSyno-MP replaces the dynamic aggregator with the "static" strategy.

Table 2 summaries the results. Comparing with ProSyno-MLP and ProSyno-BL, ProSyno achieves the optimal performance, and ProSyno-BL performs better than ProSyno-MLP, which show that the more advanced the semantic encoder, the better the performance. This might be because advanced encoders can obtain better representations. ProSyno-MP underperforms ProSyno, validating the effectiveness of the dynamical matching mechanism. This might be because dynamical matching mechanism can assign higher weights for informative descriptions align with the target term, thus leads to better performance.

To further investigate why dynamic matching mechanism can work, we performed some micro-level case studies. To be specifically, we randomly select a target term and the term has two synonyms. Table 3 shows the attention weights and prediction score. We have the following observations: 1) For different candidate terms, the attention weights of word descriptions vary significantly. For example, when predicting synonym relationship between the target term and the positive, relatively high attention weights of the word descriptions of "anxiety" and "disorders" are first and third, respectively. This is probably because that the semantics of these two descriptions are more close to the candidate term, and thus they are more informative in deciding whether the term pairs are synonym. 2) Comparing with the weights of the negative candidate, the weights of positive candidate are more discriminate, which validates that ProSyno is capable of assigning higher weights for informative descriptions and thus leads to better performance.

Table 2 Ablation study (%)

Datasets	AAP	TwADR-L
ProSyno-P	85.96	48.13
ProSyno-SP	86.35	49.10
ProSyno-TP	88.19	50.32
ProSyno-RTP	88.45	51.22
ProSyno-CAT	87.35	49.30
ProSyno-MLP	86.03	46.95
ProSyno-BL	88.92	50.21
ProSyno-MP	89.24	51.35
ProSyno	**90.22**	**51.49**

5.6 Further analysis

● Initiation

Previous works suggest that better performance can be achieved by taking real token embeddings to initiate prompts. We compare random initialization (ProSyno-RAND) with manual initialization (ProSyno). The former initializes task-oriented prompts randomly, which samples from a zero-mean Gaussian distribution with 0.02 standard deviation. The latter uses the embeddings of "synonym" to initialize the task-oriented prompts. Table 4 suggests a manual initialization has no significant positive effect. Further tuning of the initialization words might achieve better performance. However, using the simple random initialization method is suggested for convenience.

● Language backbones

To study the effectiveness of different PLMs, we compare ProSyno with ProSyno-BERT which replaces BioBERT with BERT_base. The experimental results are shown in Table 4. It empirically shows that BioBERT is more appropriate for our datasets than general BERT. This is due to the data distribution of raw text data, where BioBERT is pre-trained, is closer to our datasets.

● Fine-tuning methods

We further investigate whether fine-tuning BioBERT (θ_f) can achieve better performance (ProSyno-FT). The results are shown in Table 4. Obviously, fine-tuning BioBERT does not work well. One possible explanation is that our data is insufficient for BioBERT to learn optimal parameters.

Table 3 Case studies of a sampled term ("*anxiety disorders*") on the effect of the dynamic matching mechanism. Word description weights of the term for the positive candidate ("*zero stress tolerance*") and the negative candidate ("*cardiac arrhythmia*") are shown

Word	Word descriptions	Positive *"zero stress tolerance"*	Negative *"cardiac arrhythmia"*
Anxiety	An unpleasant state of mental uneasiness, nervousness, apprehension and obsession or concern about some uncertain event.	**0.6632**	**0.3521**
	An uneasy or distressing desire (for something).	0.2034	0.3215
	A state of restlessness and agitation, often accompanied by a distressing sense of oppression or tightness in the stomach.	0.1334	0.3264
Disorders	Absence of order; state of not being arranged in an orderly manner.	0.2180	0.3422
	A disturbance of civic peace or of public order.	0.2397	**0.3600**
	A physical or mental malfunction.	**0.5423**	0.2978

Table 4 Further analysis (%)

Datasets	AAP	TwADR-L
ProSyno-RAND	89.35	51.22
ProSyno-BERT	88.45	47.23
ProSyno-FT	89.95	51.30
ProSyno	**90.22**	**51.49**

● Dataset generalization

In this subsection, we train ProSyno on AAP dataset to obtain ProSyno-AA, and then ProSyno-AA is employed to make predictions on the other two medical datasets (TwADR-L and CADEC) and one general dataset (ANV) without additional fine-tuning. Table 5 summarizes the results. ProSyno-AA achieves competitive performance, comparing to state-of-the-art methods. This suggests that the learned model is generalizable to other dataset. The explanation is that ProSyno distinguishes whether a term pair is synonym on the semantic level, which is independent on datasets. However, this approach cannot achieve optimal performance unless fine-tuning the model on the corresponding dataset, which indicates that ProSyno may learn domain knowledge and harm the transferability. It's notable that the harm of the transferability is limited. Our explanation is that parameters of the PLM and embeddings of tokens are frozen and these frozen parameters are beneficial to ProSyno's transferability.

6 Conclusion

In this paper, a novel and effective synonym discovery model named ProSyno is proposed to deal with context-free terms. It first integrates word descriptions in Wiktionary by a hierarchical semantic encoder to generate semantic prompts, which can mitigate semantic gaps among term pairs. To obtain more informative semantic representations of words, a dynamical matching mechanism-based aggregator is designed to prioritize descriptions which are closer to the candidate term. At last, prompt learning method is employed to enhance the generalization ability of the model. Experimental results validate its superiority. Due to the simplicity of ProSyno, it allows easy extension for future work.

7 Limitations

While our research has broadly potential implication, there are still several limitations should be noted. One limitation of our proposed method is sensitivity to initiation of prompts. It is difficult to search for a suitable initialization. Fortunately, we find that the effect of manual initializations is not significant. In addition, the experiments were conducted with several versions of BERT models only, and it is unclear how large language models (LLMs) like GPTs (Generative Pre-Training Transformer models) may perform with synonym detection task. In the follow-up studies, we will further explore LLMs-based methods to solve this task.

Table 5 Performance of generalization (%)

Datasets	TwADR-L	CADEC	ANV
WordCNN	44.78	–	–
WordGRU	–	–	–
BERT	47.02	–	–
BioBERT	48.32	–	–
CODER	–	**59.01**	–
MoE-ASD	47.65	58.24	92.16
ProSyno-AA	**50.03**	58.92	**93.57**

Acknowledgements This work was supported by the National Key R&D Program of China (2023YFC3304104) and the National Natural Science Foundation of China (Grant No. 62172094).

Competing interests The authors declare that they have no competing interests or financial conflicts to disclose.

References

1. Luo X, Bo L, Wu J, Li L, Luo Z, Yang Y, Yang K. AliCoCo2: commonsense knowledge extraction, representation and application in E-commerce. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021, 3385−3393
2. Li M, Xing Y, Kong F, Zhou G. Towards better entity linking. Frontiers of Computer Science, 2022, 16(2): 162308
3. Zhang M, He T, Dong M. Meta-path reasoning of knowledge graph for commonsense question answering. Frontiers of Computer Science, 2024, 18(1): 181303
4. Xu D, Miller T. A simple neural vector space model for medical concept normalization using concept embeddings. Journal of Biomedical Informatics, 2022, 130: 104080
5. Zhang C, Li Y, Du N, Fan W, Yu P S. Entity synonym discovery via multipiece bilateral context matching. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence. 2021, 199
6. Pei S, Yu L, Zhang X. Set-aware entity synonym discovery with flexible receptive fields. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(1): 891−904
7. Yuan Z, Zhao Z, Sun H, Li J, Wang F, Yu S. CODER: knowledge-infused cross-lingual medical term embedding for term normalization. Journal of Biomedical Informatics, 2022, 126: 103983
8. Garcia M. Exploring the representation of word meanings in context: a case study on homonymy and synonymy. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. 2021, 3625−3640
9. Miftahutdinov Z, Tutubalina E. Deep neural models for medical concept normalization in user-generated texts. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. 2019, 393−399
10. Wang Z, Yue X, Moosavinasab S, Huang Y, Lin S, Sun H. SurfCon: synonym discovery on privacy-aware clinical data. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019, 1578−1586
11. Gao Y, Wang X, He X, Feng H, Zhang Y. Rumor detection with self-supervised learning on texts and social graph. Frontiers of Computer Science, 2023, 17(4): 174611
12. Zhang N, Jia Q, Deng S, Chen X, Ye H, Chen H, Tou H, Huang G, Wang Z, Hua N, Chen H. AliCG: fine-grained and evolvable conceptual graph construction for semantic search at Alibaba. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021, 3895−3905
13. Xie T, Wu B, Jia B, Wang B. Graph-ranking collective Chinese entity linking algorithm. Frontiers of Computer Science, 2020, 14(2): 291−303
14. Wang C, He X, Zhou A. A short survey on taxonomy learning from text corpora: Issues, resources and recent advances. In: Proceedings of 2017 Conference on Empirical Methods in Natural Language Processing. 2017, 1190−1203
15. Zhang J, Trujillo L B, Li T, Tanwar A, Freire G, Yang X, Ive J, Gupta

V, Guo Y. Self-supervised detection of contextual synonyms in a multi-class setting: Phenotype annotation use case. In: Proceedings of 2021 Conference on Empirical Methods in Natural Language Processing. 2021, 8754−8769

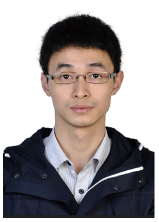
16. Zhang T, Cai Z, Wang C, Qiu M, Yang B, He X. SMedBERT: a knowledge-enhanced pre-trained language model with structured semantics for medical text mining. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. 2021, 5882−5893
17. Yang Y, Yin X, Yang H, Fei X, Peng H, Zhou K, Lai K, Shen J. KGSynNet: a novel entity synonyms discovery framework with knowledge graph. In: Proceedings of the 26th International Conference. 2021, 174−190
18. Wang C, Qiu M, Huang J, He X. KEML: a knowledge-enriched meta-learning framework for lexical relation classification. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence. 2021, 13924−13932
19. Shen J, Lyu R, Ren X, Vanni M, Sadler B, Han J. Mining entity synonyms with efficient neural set generation. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence. 2019, 249−256
20. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language models are unsupervised multitask learners. OpenAI Blog, 2019, 1(8): 9
21. Devlin J, Chang M W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics. 2019, 4171−4186
22. Zeng J, Wang Z, Yu Y, Wen J, Gao M. Word embedding methods in natural language processing: a review. Journal of Frontiers of Computer Science and Technology, 2024, 18(1): 24−43
23. Liu P, Yuan W, Fu J, Jiang Z, Hayashi H, Neubig G. Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. ACM Computing Surveys, 2023, 55(9): 195
24. Li X L, Liang P. 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. 2021, 4582−4597
25. Zhong Z, Friedman D, Chen D. Factual probing is [MASK]: learning vs. learning to recall. In: Proceedings of 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2021, 5017−5033
26. Izbicki M. Aligning word vectors on low-resource languages with wiktionary. In: Proceedings of the 5th Workshop on Technologies for Machine Translation of Low-Resource Languages. 2022, 107−117
27. Bajčetić L, Declerck T. Using wiktionary to create specialized lexical resources and datasets. In: Proceedings of the 13th Conference on Language Resources and Evaluation. 2022
28. Fang Y, Wang S, Xu Y, Xu R, Sun S, Zhu C, Zeng M. Leveraging knowledge in multilingual commonsense reasoning. In: Proceedings of the Findings of the Association for Computational Linguistics. 2022, 3237−3246
29. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017, 5998−6008
30. Miller G A. WordNet: a lexical database for English. Communications of the ACM, 1995, 38(11): 39−41
31. Limsopatham N, Collier N. Normalising medical concepts in social media texts by learning semantic representation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016, 1014−1023
32. Tutubalina E, Miftahutdinov Z, Nikolenko S, Malykh V. Medical concept normalization in social media posts with recurrent neural networks. Journal of Biomedical Informatics, 2018, 84: 93−102
33. Xu D, Zhang Z, Bethard S. A generate-and-rank framework with semantic type regularization for biomedical concept normalization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020, 8452−8464
34. Lee J, Yoon W, Kim S, Kim D, Kim S, So C H, Kang J. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 2020, 36(4): 1234−1240
35. Xie Z, Zeng N. A mixture-of-experts model for antonym-synonym discrimination. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. 2021, 558−564
36. Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 2017, 5: 135−146

Song Zhang is a PhD candidate at Institute of Automation, Chinese Academy of Sciences (CAS), China. His research interests include NLP and machine learning.

Lei He is a senior research engineer at Machine Learning Platform Department in Tencent, China. She received the PhD degree from the Institute of Computing Technology, CAS, China in 2018. Her research interests include NLP and machine learning.

Dong Wang is an algorithm engineer at Tencent, China. He received the MS degree from Tsinghua University, China in 2021. His research interests include NLP, deep learning and KG.

Hongyun Bao is an associate professor in Institute of Automation, CAS, China. She received the PhD degree from Institute of Automation, CAS, China in 2013. Her research interests include KG construction and information extraction.



Suncong Zheng is responsible for Tencent's Lexical tools, Tencent's large-scale knowledge graph Topbase. He received the PhD degree from Institute of automation, CAS, China in 2017 and obtained ACL-2017 outstanding paper award. His research interests include information extraction, KB-QA and recommendation.

Yuqiao Liu is studying for a master's degree at CAS, China. His research interests include recommendation system and data mining.

Dongyuan Lu is a professor in University of International Business and Economics, China. She received her PhD degree from Institute of Automation, CAS, China in 2012. Her research interests include data mining and natural language processing.

Baihua Xiao is a professor in Institute of Automation, CAS, China. He received the BS degree in automatic control from Northwestern Polytechnical University, China in 1995, and the PhD degree in computer science from Institute of Automation, CAS, China in 2000. His research interests include pattern recognition, computer vision, image processing, and machine learning.

Nan Zheng is an associate professor at Institute of Automation, CAS, China. She received the PhD degree from Institute of Automation, CAS, China in 2012. Her research interests include data mining and machine learning. She was a visiting scholar at University of California, Berkeley, USA in 2019.

Jiayue Li received his PhD degree in computer science and engineering from The Hong Kong University of Science and Technology, China. He did postdoctoral research in Arizona State University, USA from 2018 to 2019. His research mainly focuses on pattern recognition, medical imaging, and distributed ledger technology.