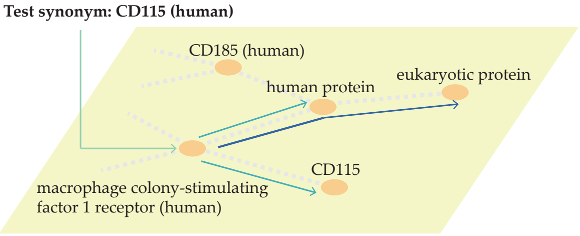
GraphPrompt: Biomedical Entity Normalization Using Graph-based Prompt Templates

Anonymous ACL submission

Abstract

Biomedical entity normalization unifies the language across biomedical experiments and studies, and further enables us to obtain a holistic view of life sciences. Current approaches mainly study the normalization of more standardized entities such as diseases and drugs, while disregarding the more ambiguous but crucial entities such as pathways, functions and cell types, hindering their real-world applications. To achieve biomedical entity normalization on these under-explored entities, we first introduce an expert-curated dataset OBO-syn encompassing 70 different types of entities and 2 million curated entity-synonym pairs. To utilize the unique graph structure in this dataset, we propose GraphPrompt, a prompt-based learning approach that creates prompt templates according to the graphs. GraphPrompt obtained 41.0% and 29.9% improvement on zero-shot and few-shot settings respectively, indicating the effectiveness of these graph-based prompt templates. We envision that our method GraphPrompt and OBO-syn dataset can be broadly applied to graph-based NLP tasks, and serve as the basis for analyzing diverse and accumulating biomedical data.


Test synonym: CD115 (human)
CD185 (human)
human protein
eukaryotic protein
CD115
macrophage colony-stimulating factor 1 receptor (human)

$\mathcal{T}^0$	**[MASK]** is identical with CD115 (human)
$\mathcal{T}^0$	**[MASK]** is identical with macrophage ... receptor (human)
$\mathcal{T}^1$	**[MASK]** is a kind of CD115
$\mathcal{T}^1$	**[MASK]** is a kind of human protein
$\mathcal{T}^2$	**[MASK]** is a kind of human protein, which is a kind of [MASK]

Figure 1: Illustration of GraphPrompt. GraphPrompt classifies a test synonym (CD115 (human)) to an entity in the graph by converting the graph into prompt templates based on the zeroth-order neighbor ($\mathcal{T}^0$), first-order neighbors ($\mathcal{T}^1$), and second-order neighbors ($\mathcal{T}^2$).

1 Introduction

Mining biomedical text data, such as scientific literature and clinical notes, to generate hypotheses and validate discovery has led to many impactful clinical applications (Zhao et al., 2021; Lever et al., 2019). One fundamental but unaddressed problem in biomedical text mining is entity normalization, which aims to map a phrase to a concept in the controlled vocabulary (Sung et al., 2020). Accurate entity normalization enables us to summarize and compare biomedical insights across studies and obtain a holistic view of biomedical knowledge. Current approaches (Wright, 2019; Ji et al., 2020; Sung et al., 2020) to biomedical entity normalization often focus on normalizing more standardized entities such as diseases (Doğan et al., 2014; Li et al., 2016), drugs (Kuhn et al., 2007; Pradhan et al., 2013), genes (Szklarczyk et al., 2016) and adverse drug reactions (Roberts et al., 2017). Despite their encouraging performance, these approaches have not yet been applied to the more ambiguous entities, such as processes, pathways, cellular components, and functions (Smith et al., 2007), which lie at the center of life sciences. As scientists rely on these entities to describe disease and drug mechanisms (Yu et al., 2016), the inconsistent terminology used across different labs inevitably hampers the scientific communication and collaboration, necessitating the normalization of these entities.

The first immediate bottleneck to achieve the normalization of these under-explored entities is the lack of a high-quality and large-scale dataset, which is the prerequisite for existing entity normalization approaches (Wright, 2019; Ji et al., 2020; Sung et al., 2020). To tackle this problem, we collected 70 types of biomedical entities from OBO Foundry (Smith et al., 2007), spanning a wide variety of biomedical areas and containing more than 2 million entity-synonym pairs. These pairs are all curated by domain experts and together form a high-quality and comprehensive controlled vocabulary for biomedical sciences, greatly augmenting existing biomedical entity normalization datasets (Doğan et al., 2014; Li et al., 2016; Roberts et al., 2017). The tedious and onerous curation of this high-quality dataset further confirms the necessity of developing data-driven approaches to automating this process and motivates us to introduce this dataset to the NLP community.

In addition to being the first large-scale dataset encompassing many under-explored entity types, this OBO-syn dataset presents a novel setting of graph-based entity normalization. Specifically, entities of the same type form a relational directed acyclic graph (DAG), where each edge represents a relationship (e.g., is_a) between two entities. Intuitively, this DAG could assist the entity normalization since nearby entities are biologically related, and thus more likely to be semantically and morphologically similar. Existing entity normalization and synonym prediction methods are incapable of considering the topological similarity from this rich graph structure (Wright, 2019; Ji et al., 2020; Sung et al., 2020), limiting their performance, especially in the few-shot and zero-shot settings. Recently, prompt-based learning has demonstrated many successful NLP applications (Radford et al., 2019; Schick and Schütze, 2020; Jiang et al., 2020). The key idea of using prompt is to circumvent the requirement of a large number of labeled data by creating masked templates and then converting supervised learning tasks to a masked-language model task (Liu et al., 2021). However, it remains unknown how to convert a large graph into text templates for prompt-based learning. Representing graphs as prompt templates might effectively integrate the topological similarity and textural similarity by alleviating the over-smoothing caused by propagating textual features on the graph.

In this paper, we propose GraphPrompt, a prompt-based learning method for entity normalization with the consideration of graph structures. The key idea of our method is to convert the graph structural information into prompt templates and solve a masked-language model task, rather than incorporating textual features into a graph-based framework. Our graph-based templates explicitly model the high-order neighbors (e.g., neighbors of neighbors) in the graph, which enables us to correctly classify synonyms that have relatively lower morphological similarity with the ground-truth entity (**Figure 1**). Experiments on the novel OBO-syn dataset demonstrate the superior performance of our method against existing entity normalization approaches, indicating the advantage of considering the graph structure. Case studies and the comparison to the conventional graph approach further reassure the effectiveness of our prompt templates, implicating opportunities on other graph-based NLP applications. Collectively, we introduce a novel biomedical entity normalization task, a large-scale and high-quality dataset, and a novel prompt-based solution to advance biomedical entity normalization.

2 Related Works

Biomedical entity normalization. Biomedical entity normalization has been studied for decades because of its significance in a variety of biomedical applications. Conventional approaches mainly relied on rule-based methods (D'Souza and Ng, 2015; Sullivan et al., 2011) or probabilistic graphical models (Leaman et al., 2013; Leaman and Lu, 2016) to model the morphological similarity, which are incapable of normalizing functional entities that are semantically similar but morphologically different. Deep learning-based approaches (Li et al., 2017; Wright, 2019; Pujary et al., 2020; Deng et al., 2019; Luo et al., 2018) and pre-trained language models (PLMs) (Ji et al., 2020; Sung et al., 2020; Lee et al., 2020; Miftahutdinov et al., 2021) have obtained encouraging results in capturing the semantics of entities through leveraging human annotations or large collections of corpus. However, these approaches focus on datasets comprising of less ambiguous entity types, such as drugs and diseases and are not able to incorporate graph structures into their framework. In contrast, we aim to utilize rich graph information to assist the normalization of more ambiguous entities such as functions, pathways and processes.

Incorporating graph structure into text modeling. Graph-based approaches, such as network embedding (Tang et al., 2015) and graph neural network (Kipf and Welling, 2016), have been used to model the structural information in the text data, such as citation networks (An et al., 2021), social networks (Masood and Abbasi, 2021; Aljohani et al., 2020) and word dependency graph (Fu et al., 2019). Among them, Kotitsas et al. (2019) considered the most similar DAG structure to our task

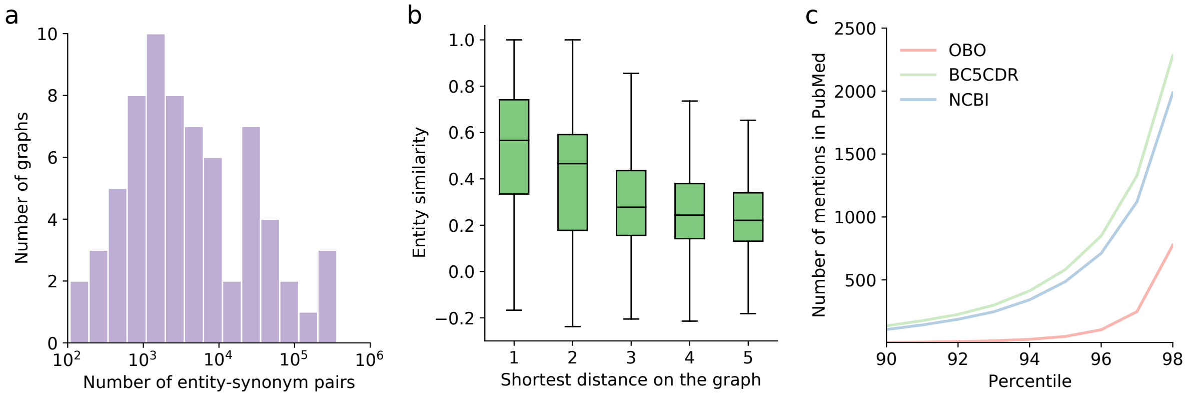


Figure 2: Analysis of the OBO-syn dataset. a, Bar plot showing the distribution of the number of entity-synonym pairs in 70 graphs. b, Box plot comparing the textual similarity of entity pairs having different shortest distances on the graph. c, Line chart comparing the phrase mentions of NCBI, BC5CDR, OBO-syn. The y-axis is the number of mentions in 28 million PubMed abstracts. The x-axis is the phrase percentile sorted by the number of mentions.

and proposed a two-stage approach to integrate graph structural with textual information. The key difference between our method and existing approaches is that we transform the graph structures into prompt templates and then solve a masked-language model task, whereas existing works represent textual information as fixed node features and then optimize a graph-based model.

Prompt-based learning. Prompt-based learning have recently shown promising results in many applications(Liu et al., 2021), such as text generation (Radford et al., 2019; Brown et al., 2020), text classification (Schick and Schütze, 2020; Gao et al., 2020) and question answering (Khashabi et al., 2020; Jiang et al., 2020). Prompt-based learning has not yet been applied to integrate the graph information. The most related prompt-based works to our task is prompt-based relation extraction (Chen et al., 2021; Han et al., 2021) and prompt-based knowledge base completion (Davison et al., 2019). These approaches only consider immediate neighbors in the graph and are not able to model more distant nodes, thus being incapable of capturing the topology of the entire graph. To the best of our knowledge, we are the first work that considers higher-order graph neighbors in the prompt-based learning framework.

3 Dataset Description and Analysis

We collected 70 relational graphs from Open Biological and Biomedical Ontology Foundry (OBO) (Smith et al., 2007). Nodes in the same relational graph represent biomedical entities belonging to the same type, such as protein functions, cell types, and disease pathways. Each edge represents a relational type, such as is_a, part_of, capable_of, and regulates. We leveraged these edge types to build templates in our prompt-based learning framework. The number of nodes in each graph ranges from 113 to 2,334,910 with a median value of 3,077. The number of synonyms for each entity ranges from 1 to 284 with a median value of 2 (ignoring the entities without synonyms). On average, each graph has 34,418 entity-synonym pairs and 72.9% of graphs have more than 1,000 entity-synonym pairs (**Figure 2a**). The graph structure and entity synonym associations are all curated by domain experts, presenting a large-scale and high-quality collection.

In comparison to other biomedical entity normalization datasets (Doğan et al., 2014; Li et al., 2016; Roberts et al., 2017), OBO-syn presents a unique graph structure among entities. Intuitively, nearby entities as well as their synonyms should be semantically similar, as their biological concepts are relevant. To validate this intuition, we investigated the consistency between graph-based entity similarity and text-based entity similarity. In particular, we used the shortest distance on the graph to calculate graph-based similarity and Sentence-BERT (Reimers et al., 2019) to calculate text-based similarity. We observed a strong correlation between these two similarity scores (**Figure 2b**), suggesting the possibility to transfer synonym annotations from nearby entities to improve the entity normalization.

We next compared this OBO-syn dataset with the existing biomedical entity normalization dataset. We first observed very small overlaps of 5.26%, 14.59%, 3.29% between our dataset and three widely-used biomedical entity normalization datasets NCBI-disease (Doğan et al., 2014), BC5CDR-disease (Li et al., 2016), and BC5CDR-chemical (Li et al., 2016), respectively. The small overlaps with existing datasets indicate the uniqueness of our dataset, and further make us question the performance of the state-of-the-art entity normalization methods on this novel dataset. More importantly, we noticed a substantially large number of out-of-vocabulary phrases in our dataset compared to existing datasets (**Figure 2c**). We calculate the number of mentions of each phrase in 29 million PubMed abstracts, which are used as the pre-training corpus for biomedical pre-trained models (Lee et al., 2020; Gu et al., 2020). The 95 percentile of the number of mentions in our dataset is only 51, substantially lower than 487 in NCBI and 582 in BC5CDR, suggesting a worse generalization ability using pre-trained language models and motivating us to exploit the graph structures for this dataset.

4 Problem Statement

The goal of entity normalization is to map a given synonym phrase s to the corresponding entity v based on their semantic similarity. One unique feature of our problem setting is that entities belonging to the same type form a relational graph. Formally, we denote this relational graph as $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$, where $\mathcal{V}$ is the set of entities, $\mathcal{R}$ is the set of relation types and $\mathcal{E} \subset \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ is the set of edges. Let $\mathcal{C}$ be the vocabulary of the corpus. Each node $v_i \in \mathcal{V}$ is represented as an entity phrase $v_i \triangleq \langle v_i^1, v_i^2, \ldots, v_i^{|v_i|} \rangle$, where $v_i^k \in \mathcal{C}$. In addition to the graph, we also have a set of mapped synonyms $\mathcal{S}$ that will be used as the training data. Each $s_j \triangleq \langle s_j^1, s_j^2, \ldots, s_j^{|s_j|} \rangle \in \mathcal{S}$ is mapped to one entity v_i in the graph $\mathcal{S}$, and $s_j^k \in \mathcal{C}$.

Our goal is to classify a test synonym s' to an entity v in the graph. Since the majority of entities only have very few synonyms (e.g., 96.9% of entities have less than 5 synonyms), we consider a few-shot and a zero-shot setting. Specifically, in the few-shot setting, the test set entities are included in the training set entities. On the contrary, the entities in the training set and the test set present no overlap in the zero-shot setting, and therefore the entities of training datasets are unobservable for test procedure. The small number of training synonyms for each entity could exacerbate over-fitting. To mitigate the over-fitting problem, we propose graph-based prompt templates, where we consider the synonyms of nearby entities in the training data.

4.1 Base model

We first introduce a base model that only considers the textual information of synonyms and entities while disregarding the graph structure. Following the previous work (Sung et al., 2020), the base model uses two encoders to calculate the similarity between the queried synonym s and the candidate entity v. The first encoder Enc_s encodes the queried synonym into the dense representation $\boldsymbol{x}_s = \mathrm{Enc}_s(t_s)$. The second encoder Enc_v encodes the candidate entity into the dense representation $\boldsymbol{x}_v = \mathrm{Enc}_v(t_v)$. Then the predicted probability of choosing entity v is calculated as:

$$P(\boldsymbol{x}_v|\boldsymbol{x}_s) = \frac{Q(\boldsymbol{x}_v, \boldsymbol{x}_s)}{\sum_{v' \in \mathcal{V}} Q(\boldsymbol{x}_{v'}, \boldsymbol{x}_s)}, \tag{1}$$

where Q is defined as $Q(\boldsymbol{x}_v, \boldsymbol{x}_s) = \exp(\boldsymbol{x}_v^\mathsf{T} \boldsymbol{x}_s)$. We select BioBERT with [CLS] readout function as Enc_v and Enc_s, and share the parameters between both encoders. Following Sung et al. (2020), the input t_v and t_s are designed as "[CLS] v [SEP]" and "[CLS] s [SEP]" respectively. In practice, we find that the initial [CLS] output vectors are fairly close. This can result in large positive $\boldsymbol{x}_v^\mathsf{T} \boldsymbol{x}_s$, which leads to slow convergence and potential numerical issues, yet it is not addressed by BioSyn (Sung et al., 2020). To alleviate this issue, we use a trainable 1-d BatchNorm layer and redefine our similarity function Q as:

$$Q(\boldsymbol{x}_v, \boldsymbol{x}_s) = \exp(\mathrm{BN}(\boldsymbol{x}_v^\mathsf{T} \boldsymbol{x}_s)). \tag{2}$$

When the candidate entity set is large, backpropagating through $\boldsymbol{x}_v$ results in high memory complexity due to the construction of $|\mathcal{V}|$ computation graphs to get $\boldsymbol{x}_{v'}$. To tackle this problem, we apply the stop gradient trick to $\boldsymbol{x}_{v'}$, following Sung et al. (2020). Besides, we utilize the *hard negative* strategy following Sung et al. (2020) by sampling difficult negative candidates $\mathcal{U} \subset \mathcal{V}$. The loss function is defined as:

$$\mathcal{L}_{\text{base}} = -\sum_{(v,s)} \log \frac{Q(\mathrm{sg}(\boldsymbol{x}_v), \boldsymbol{x}_s)}{\sum_{v' \in \mathcal{U} \cup \{v\}} Q(\mathrm{sg}(\boldsymbol{x}_{v'}), \boldsymbol{x}_s)}, \tag{3}$$

where sg denotes the stop gradient operation.Besides, To further save computation time, we cache the values of $\mathrm{sg}(\boldsymbol{x}_{v'}), v' \in \mathcal{V}$ and iteratively update them. See more details for the base model in A.2.

4.2 Prompt model

Previous work (Sung et al., 2020) and the base model use BioBERT with [CLS] readout function as the encoders, which take the synonym or entity as the input and use the hidden state of [CLS] as the output. However, using the synonym or entity as the input text might not fully capture its semantic since PLMs are often pre-trained with sentences instead of phrases. To tackle this problem, we construct two simple prompt templates $\mathcal{T}^0$ for a training entity-synonym pair (v, s) as: $\mathcal{T}^0([\text{MASK}], v)$ = "[MASK] is identical with v" and $\mathcal{T}^0([\text{MASK}], s)$ = "[MASK] is identical with s" ([CLS] and [SEP] are omitted). Then we optimize the model by solving an masked language modeling task, where we use the output of BioBERT at [MASK] token as the dense representation $\boldsymbol{x}_v$ ($\boldsymbol{x}_s$) for v (s), respectively:

$$\boldsymbol{x}_v = \text{BioBERT}(\mathcal{T}^0([\text{MASK}], v)), \quad (4)$$

$$\boldsymbol{x}_s = \text{BioBERT}(\mathcal{T}^0([\text{MASK}], s)). \quad (5)$$

Since the graph is not used here, we refer to $\boldsymbol{x}_v$ as the zeroth-order representation of entity v. The loss function of prompt model is similar to base model's, where we select the whole entity set $\mathcal{V}$ as candidates instead of its subset:

$$\mathcal{L}_\text{p} = -\sum_{(v,s)} \log \frac{Q(\text{sg}(\boldsymbol{x}_v), \boldsymbol{x}_s)}{\sum_{v' \in \mathcal{V}} Q(\text{sg}(\boldsymbol{x}_{v'}), \boldsymbol{x}_s)}. \quad (6)$$

5 GraphPrompt model

5.1 Intuition

The observation that nearby entities are more semantically similar (**Figure 2b**) motivates us to integrate textual similarity with graph topological similarity to boost the entity normalization. Conventional approaches often integrate text and graph information by adapting a graph-based framework and incorporating text features as node features (Kotitsas et al., 2019). However, such approaches might not fully utilize the strong generalization ability of pre-trained models, which have been crucial for a variety of NLP tasks (Devlin et al., 2018; Petroni et al., 2019). In contrast to conventional approaches, we propose to utilize a prompt-based learning framework to integrate text and graph information through representing the graph information as prompt templates. To the best of our knowledge, our method is the first attempt to represent the graph structure as prompt templates.

5.2 First-order GraphPrompt

GraphPrompt uses **Equation 1** for inference, but utilizes the graph information during training. GraphPrompt considers first-order neighborhood (i.e., immediate neighbors) and second-order neighborhood (i.e., neighbors of the neighbors) to construct prompt templates for a given entity.

To model first-order neighbors, GraphPrompt defines the template $\mathcal{T}_r^1(v_i, v_j)$ = "$v_i\ r'\ v_j$" for an edge between entity v_i and its immediate neighbor entity v_j with relation type r. r' is created from r with minor morphological change, as listed in **Table 3**. For a given triple (v_i, r, v_j) in the graph, we create a masked-language model task by randomly masking v_i or v_j. We also include the template that replaces the unmasked v with its training synonym s. For example, when v_i is masked and v_j is replaced with s_k, we obtain the following template: $\mathcal{T}_r^1([\text{MASK}], v_j)$ = "[MASK] r' s_k". We then use BioBERT to obtain the first-order representation $\boldsymbol{y}_{v_i}$ based on this template:

$$\boldsymbol{y}_{v_i} = \text{BioBERT}(\mathcal{T}_r^1([\text{MASK}], v_j)). \quad (7)$$

We then calculate the loss term by comparing the first-order representation of v_i with the zeroth-order presentation $\boldsymbol{x}_{v_i}$:

$$\mathcal{L}_1 = -\sum_{(v_i, v_j)} \log P(\boldsymbol{x}_{v_i} | \boldsymbol{y}_{v_i}) \quad (8)$$

5.3 Second-order GraphPrompt

To consider second-order neighbors, GraphPrompt first finds all 2-hop relational paths $(v_i, r, v_j, \text{is\_a}, v_k)$ in the graph. Since is_a relation contributes to the majority of the relation type, we fix the second relation to be is_a for simplicity. The prompt template is then defined as $\mathcal{T}_r^2(v_i, v_j, v_k)$ = "$v_i\ r'\ v_j$, which is a kind of v_k".

Different from $\mathcal{T}^0$ and $\mathcal{T}^1$, there are three tokens that can be masked in $\mathcal{T}^2$. We chose to mask two tokens in each template, resulting in two kinds of second-order templates:

$$\boldsymbol{z}_{v_i}, \boldsymbol{z}_{v_k} = \text{BioBERT}(\mathcal{T}_r^2([\text{MASK}], v_j, [\text{MASK}]))$$

$$\boldsymbol{z}_{v_j}, \boldsymbol{z}_{v_k} = \text{BioBERT}(\mathcal{T}_r^2(v_i, [\text{MASK}], [\text{MASK}]))$$

We don't consider the template of $\mathcal{T}_r^2([\text{MASK}], [\text{MASK}], v_k)$ because of the DAG structure in our dataset. The numbers of child nodes and grandchild nodes grow exponentially in DAG and will introduce too many paths using $\mathcal{T}_r^2([\text{MASK}], [\text{MASK}], v_k)$ template, slowing down the optimization.

To calculate the loss term based on $\mathcal{T}_r^2([\text{MASK}], v_j, [\text{MASK}])$, we compare the second-order dense representation $\boldsymbol{z}_{v_i}$, $\boldsymbol{z}_{v_k}$ to the zeroth-order dense representation $\boldsymbol{x}_{v_i}$, $\boldsymbol{x}_{v_k}$. $\boldsymbol{z}'_{v_j}$, $\boldsymbol{z}'_{v_k}$ and the loss term based on $\mathcal{T}_r^2(v_i, [\text{MASK}], [\text{MASK}])$ is defined similarly. We define the loss term for second-order neighbors as:

Dataset	mp				cl				hp				fbbt				doid			
#synonyms	26119				27242				20070				23870				21401			
#entities	13752				10939				16544				17475				13313			
data split	zero-shot		few-shot		zero-shot		few-shot		zero-shot		few-shot		zero-shot		few-shot		zero-shot		few-shot	
	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10
Sieve-Based	3.40	–	–	–	6.80	–	–	–	4.60	–	–	–	1.00	–	–	–	8.20	–	–	–
BNE	43.60	68.10	51.70	70.90	40.00	65.30	49.20	66.60	32.50	58.40	36.70	59.40	34.30	59.80	44.40	62.10	40.10	58.40	40.00	57.40
NormCo	–	–	41.25	53.48	–	–	52.68	59.76	–	–	49.44	55.25	–	–	32.58	42.57	–	–	58.23	67.01
TripletNet	40.67	67.54	42.39	68.75	28.81	60.07	28.61	60.96	26.54	51.99	27.69	52.29	6.21	14.14	3.15	9.46	26.59	43.34	25.79	42.92
BioSyn	62.04	74.08	73.55	82.67	53.55	66.18	64.34	75.59	50.72	67.24	59.84	72.25	40.38	61.95	57.32	68.06	40.68	53.82	48.90	61.04
GCN	70.88	88.61	83.83	93.18	60.49	84.09	74.95	90.50	54.92	79.47	68.39	87.17	48.98	73.68	54.91	78.50	46.57	65.91	55.27	75.42
Base model	76.47	88.38	85.78	93.02	65.14	82.58	76.64	89.19	61.53	78.14	72.04	86.38	65.38	79.24	69.76	84.08	50.71	64.86	59.40	74.69
Prompt	79.51	89.74	87.56	95.41	66.45	82.02	80.93	93.33	62.80	81.67	75.12	90.40	68.53	79.29	73.75	86.49	56.43	69.51	66.31	81.03
GraphPrompt (w/o $\mathcal{T}^2$)	79.40	90.95	88.05	**95.75**	68.00	**84.89**	82.04	93.81	66.36	84.24	77.60	92.33	68.10	80.08	**75.65**	89.06	56.85	**70.67**	66.70	81.99
GraphPrompt	**80.00**	**91.62**	**88.86**	95.61	**68.87**	84.34	**82.86**	**94.24**	**68.63**	**85.24**	**77.87**	**92.66**	**69.01**	**81.90**	75.55	**89.74**	**57.42**	70.55	**67.53**	**82.50**

Table 1: The performance of our method and comparison approaches on 5 datasets using zero-shot and few-shot settings. The best model in each column is colored in blue and the second best is colored in light blue. See **Appendix A.2** for detailed implementations of comparison approaches.

$$\mathcal{L}_2 = -\sum_{(v_i,v_j,v_k)} (\log P(\boldsymbol{x}_{v_i}|\boldsymbol{z}_{v_i}) + \log P(\boldsymbol{x}_{v_k}|\boldsymbol{z}_{v_k}) + \log P(\boldsymbol{x}_{v_j}|\boldsymbol{z}'_{v_j}) + \log P(\boldsymbol{x}_{v_k}|\boldsymbol{z}'_{v_k})).$$

Although we can further define higher-order templates accordingly, we observed limited improvement by including third-order or even higher-order templates in our experiments. This observation is consistent with conventional graph embedding approaches where only first-order and second-order neighborhood are explicitly modeled (Tang et al., 2015). For the 2-hop relational path, we didn't consider sibling-based templates such as "Both [MASK] and [MASK] are a kind of v" due to the large number of sibling pairs in the DAG. Nevertheless, such templates might be worth exploring on other graphs.

In practice, different entities may have similar $\boldsymbol{x}_v$, making them indistinguishable at the test stage. This issue could be exacerbated when the graph structure is incorporated. For example, for two edges $(v_i, \text{is\_a}, v_j)$ and $(v_{i'}, \text{is\_a}, v_j)$, the model tends to increase the similarity between the embeddings of siblings v_i and $v_{i'}$. To alleviate this problem, we consider another contrastive loss term $\mathcal{L}_\text{c}$ that encourages the model to distinguish different entities:

$$\mathcal{L}_\text{c} = -\sum_v \log P(\boldsymbol{x}_v|\boldsymbol{x}_v) \qquad (9)$$

The final loss of our model combines of $\mathcal{L}_\text{p}$, $\mathcal{L}_\text{c}$, $\mathcal{L}_1$ and $\mathcal{L}_2$, with weights λ_p, λ_c, λ_1 and λ_2 chosen on the validation set:

$$\mathcal{L} = \lambda_\text{p}\mathcal{L}_\text{p} + \lambda_\text{c}\mathcal{L}_\text{c} + \lambda_1\mathcal{L}_1 + \lambda_2\mathcal{L}_2. \qquad (10)$$

6 Experimental Results

6.1 Experimental settings

We selected five graphs (mp, cl, hp, fbbt, doid) with the number of entities between 10,000 and 20,000 from OBO-syn. We investigated a few-shot setting and a zero-shot setting. In the few-shot setting, we split the synonyms into six folds, and then used four folds as training set, one fold as validation set and one fold as test set. In the zero-shot setting, we split all entities into three folds, and then used two folds as training set and one fold as test set. All synonyms of training (test) entities are observable (unobservable) during training. Our method and all comparison approaches used the same data split.

We compared our method to the state-of-the-art entity normalization approaches: Sieve-Based (D'Souza and Ng, 2015) , BNE (Phan et al., 2019), NormCo (Wright, 2019), TripletNet (Mondal et al., 2020) and BioSyn (Sung et al., 2020), and a graph convolutional network (GCN) (Kipf and Welling, 2016). We also compared our method with a base model ($\mathcal{L}_{base}$), a prompt model ($\mathcal{L}_\text{p} + \mathcal{L}_\text{c}$) and a first-order GraphPrompt (w / o $\mathcal{T}^2$) ($\mathcal{L}_1 + \mathcal{L}_\text{p} + \mathcal{L}_\text{c}$). See more details for the implement of baselines and our methods in appendix A.2 .

6.2 Improved performance in few-shot setting

We first sought to evaluate the performance of our method in the few-shot setting (**Table 1**). We found that our method outperformed all other approaches in all metrics on all the datasets. When comparing to the best-performed entity normalization approach BioSyn, our method obtains an average 27.7% improvement on Acc@10 and 35.5% improvement on Acc@1, indicating the prominence of using the graph structure to leverage annotations from nearby entities. We found that using graph structure leads to large improvement on datasets with fewer training samples (39.6% improvement on doid comparing to 24.9% on mp), suggesting

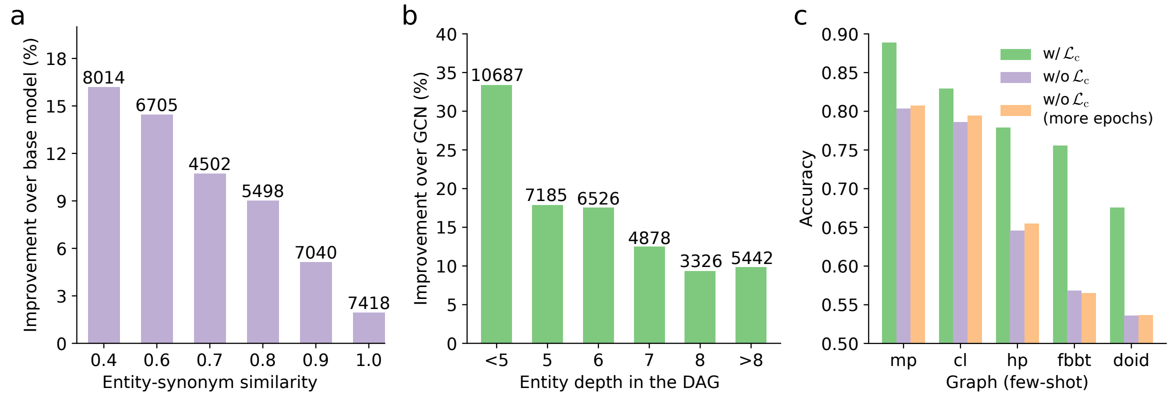


Figure 3: Performance analysis of GraphPrompt. a, Bar plot showing the improvement of GraphPrompt over the base model under different entity-synonym similarity intervals. x-axis is the upper bound of the interval (e.g., 0.6 stands for [0.4-0.6]). b, Bar plot showing the improvement of GraphPrompt over GCN under different entity depths in the DAG. c, Bar plot showing the effect of $\mathcal{L}_c$ in the few-shot setting. Orange bar stands for training 2 times more epochs without using $\mathcal{L}_c$.

GraphPrompt's ability to learn from limited samples.

We next compared our method to a graph-based approach GCN and observed a superior performance of GraphPrompt, confirming the effectiveness of modeling graph structures using prompt templates. The base model, which does not exploit the graph structure, also performed better than GCN, partially due to the over-smoothing issue in GCN. Despite showing a less superior performance comparing to our method, GCN still outperformed most of the entity normalization approaches that do not consider graph structure, reassuring the advantage of using graph structure in this dataset.

To further verify that the improvement of our method comes from using graph structure, we compared the performance of GraphPrompt with the base prompt model and the first-order prompt model. Overall, GraphPrompt is better than both approaches by utilizing the second-order neighborhood, while the first-order prompt is better than the base prompt model. Collectively, our results clearly assure the importance of considering the graph structure and the effectiveness of modeling it using prompt templates.

6.3 Improved performance in zero-shot setting

After verifying the superior performance of our method in few-shot learning, we next investigate the more challenging zero-shot setting, where ground-truth entities in the test set have no synonyms in the training set (**Table 1**). Likewise, our method outperformed all comparison approaches in all metrics on all datasets. We found that GraphPrompt obtained larger improvement over BioSyn in the zero-shot setting compared to the few-shot setting. Since ground-truth entities do not have any observed synonyms in the zero-shot setting, graph information becomes more crucial to aggregate synonym annotations from nearby entities.

The consistent improvement of GraphPrompt over GCN in both zero-shot and few-shot settings further confirms the effectiveness of using prompt templates to capture the graph structure. GraphPrompt also shows consistent improvement over the base prompt model and the first-order prompt model, indicating the importance of considering second-order neighbors in the graph.

6.4 Improvement analysis

We sought to investigate the superior performance of GraphPrompt. We first calculated the textual similarity between the test synonyms and their ground truth entities using Sentence-BERT (Reimers et al., 2019). We found that the improvement of GraphPrompt over the base model increases with the decreasing of this textual similarity (**Figure 3a**). Entity-synonym pairs that have smaller textual similarity are more difficult to be predicted correctly with only the textual information, thus obtaining larger improvement from the graph structure. Moreover, the low overlaps with pre-training corpus limit the knowledge from PLMs, necessitating the consideration of graph information.

We then sought to study the improvement of GraphPrompt over GCN. Interestingly, we found that GCN tends to have better performance on Acc@10 rather than Acc@1, whereas our method shows consistent improvement on these two met-

Test synonym	adult Leucokinin ABLK neuron of the abdominal ganglion
Ground-truth entity	adult abdominal ganglion Leucokinin neuron
Baseline predictions	adult Leucokinin neuron of the central nervous system (**GCN**), adult anterior LK Leucokinin neuron (**Base model**)
Prompt templates	[adult abdominal ganglion Leucokinin neuron] is a kind of abdominal neuron, which is a kind of [...] [...] is identical with larval Leucokinin ABLK neuron of the abdominal ganglion, which is a kind of [abdominal neuron]

Table 2: An example of GraphPrompt prediction. Selected second-order prompt templates that affect the results are listed. Masked tokens are displayed within brackets.

rics. As GCN is known to suffer from over-smoothing (Li et al., 2018; Hoang and Maehara, 2019), it might not distinguish very close entities on the graph, leading to much worse top 1 prediction performance, but better top 10 prediction performance. In contrast, our prompt-based graph learning does not show a performance degrade in top 1 prediction, suggesting that our method is less prone to over-smoothing.

To further verify this, we examined the improvement of our method against GCN at different depths in the graph (**Figure 3b**). We found that the improvement of our method over GCN becomes larger when the depth of the entity is smaller. Because of the DAG structure in our graph, entities that have smaller depth are closer to the center of the graph, and could be more disturbed by the over-smoothing issue. In contrast, our method explicitly converts the graph structure into prompt templates, successfully alleviating the over-smoothing issue caused by propagating on the entire graph.

Next, we examined the effect of the $\mathcal{L}_c$ norm in our method (**Figure 3c**). As expected, adding $\mathcal{L}_c$ greatly improved the performance on all the datasets in the few-shot setting. The improvement is much larger on datasets that have worse overall performance (e.g., fbbt, doid), indicating the importance of separating the embeddings of different entities. We also noticed that the accuracy of the state-of-the-art entity normalization approaches, such as BioSyn and NormCo, is much worse on our OBO-syn dataset than on the mainstream datasets, such as BC5CDR and NCBI (see results in Sung et al. (2020)), further confirming the difficulty of our task and dataset.

Finally, we presented two case studies of how GraphPrompt utilized the graph structure to correctly identify the entity (**Figure 1** and **Table 2**). We found that GraphPrompt performed a 'recombination' of two nearby phrases using the graph-based prompt templates during the prediction. For example, GraphPrompt correctly classified the test synonym 'adult Leucokinin ABLK neuron of the abdominal ganglion' to the entity 'adult abdominal ganglion Leucokinin neuron' by combining it with the second-order neighbor 'larval Leucokinin ABLK neuron of the abdominal ganglion', whereas comparison approaches classified to incorrect but semantically similar entities (e.g., 'adult anterior LK Leucokinin neuron') (**Table 2**). Likewise, GraphPrompt correctly classified 'CD115 (human)' to 'macrophage ... receptor (human)' by recombining it with CD115 according to the first-order prompt template. These recombinations of nearby entities reassure the effectiveness of graph-based prompts in biomedical entity normalization.

7 Conclusion and Future Work

We have presented a novel biomedical entity normalization dataset OBO-syn that encompasses 70 biomedical entity types and 2 million entity-synonym pairs. OBO-syn has demonstrated small overlaps with existing datasets and more challenging entity-synonym predictions. To leverage the unique graph structures in OBO-syn, we have proposed GraphPrompt, which converts graph structures into prompt templates and then solves a masked-language model task. GraphPrompt has obtained superior performance to the state-of-the-art entity normalization approaches on both few-shot and zero-shot settings.

Since GraphPrompt can in principle be applied to integrate other types of graphs and text information, we are interested in exploiting GraphPrompt in other graph-based NLP tasks, such as citation network analysis and graph-based text generation. The novel OBO-syn dataset can also advance tasks beyond entity normalization, such as link prediction, graph representation learning, and be integrated with other scientific literature datasets to investigate entity linking, key phrase mining, and named entity recognition. We envision that our method GraphPrompt and OBO-syn will pave the path for comprehensively analyzing diverse and accumulating biomedical data.

References

Naif Radi Aljohani, Ayman Fayoumi, and Saeed-Ul Hassan. 2020. Bot prediction on social networks of twitter in altmetrics using deep graph convolutional networks. Soft Computing, pages 1–12.

Chenxin An, Ming Zhong, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2021. Enhancing scientific papers summarization with citation graph. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 12498–12506.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Xiang Chen, Xin Xie, Ningyu Zhang, Jiahuan Yan, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Adaprompt: Adaptive prompt-based finetuning for relation extraction. arXiv preprint arXiv:2104.07650.

Joe Davison, Joshua Feldman, and Alexander Rush. 2019. Commonsense knowledge mining from pretrained models. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China. Association for Computational Linguistics.

Pan Deng, Haipeng Chen, Mengyao Huang, Xiaowen Ruan, and Liang Xu. 2019. An ensemble cnn method for biomedical entity normalization. In Proceedings of the 5th workshop on BioNLP open shared tasks, pages 143–149.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. Journal of biomedical informatics, 47:1–10.

Jennifer D'Souza and Vincent Ng. 2015. Sieve-based entity linking for the biomedical domain. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 297–302.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1409–1418.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. arXiv preprint arXiv:2012.15723.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing. arXiv preprint arXiv:2007.15779.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. arXiv preprint arXiv:2105.11259.

NT Hoang and Takanori Maehara. 2019. Revisiting graph neural networks: All we have is low-pass filters. arXiv preprint arXiv:1905.09550, 2.

Zongcheng Ji, Qiang Wei, and Hua Xu. 2020. Bert-based ranking for biomedical entity normalization. AMIA Summits on Translational Science Proceedings, 2020:269.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? Transactions of the Association for Computational Linguistics, 8:423–438.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. arXiv preprint arXiv:2005.00700.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Sotiris Kotitsas, Dimitris Pappas, Ion Androutsopoulos, Ryan McDonald, and Marianna Apidianaki. 2019. Embedding biomedical ontologies by jointly encoding network structure and textual node descriptors. arXiv preprint arXiv:1906.05939.

Michael Kuhn, Christian von Mering, Monica Campillos, Lars Juhl Jensen, and Peer Bork. 2007. Stitch: interaction networks of chemicals and proteins. Nucleic acids research, 36(suppl_1):D684–D688.

Robert Leaman, Rezarta Islamaj Doğan, and Zhiyong Lu. 2013. Dnorm: disease name normalization with pairwise learning to rank. Bioinformatics, 29(22):2909–2917.

Robert Leaman and Zhiyong Lu. 2016. Taggerone: joint named entity recognition and normalization with semi-markov models. Bioinformatics, 32(18):2839–2846.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4):1234–1240.

Jake Lever, Eric Y Zhao, Jasleen Grewal, Martin R Jones, and Steven JM Jones. 2019. Cancermine: a literature-mined resource for drivers, oncogenes and tumor suppressors in cancer. Nature methods, 16(6):505–507.

Haodi Li, Qingcai Chen, Buzhou Tang, Xiaolong Wang, Hua Xu, Baohua Wang, and Dong Huang. 2017. Cnn-based ranking for biomedical entity normalization. BMC bioinformatics, 18(11):79–86.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. Database, 2016.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In Thirty-Second AAAI conference on artificial intelligence.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586.

Yi Luo, Guojie Song, Pengyu Li, and Zhongang Qi. 2018. Multi-task medical concept normalization using multi-view convolutional neural network. In Thirty-Second AAAI Conference on Artificial Intelligence.

Muhammad Ali Masood and Rabeeh Ayaz Abbasi. 2021. Using graph embedding and machine learning to identify rebels on twitter. Journal of Informetrics, 15(1):101121.

Zulfat Miftahutdinov, Artur Kadurin, Roman Kudrin, and Elena Tutubalina. 2021. Medical concept normalization in clinical trials with drug and disease representation learning. Bioinformatics.

Ishani Mondal, Sukannya Purkayastha, Sudeshna Sarkar, Pawan Goyal, Jitesh Pillai, Amitava Bhattacharyya, and Mahanandeeshwar Gattu. 2020. Medical entity linking using triplet network. arXiv preprint arXiv:2012.11164.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2463–2473.

Minh C Phan, Aixin Sun, and Yi Tay. 2019. Robust representation learning of biomedical names. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3275–3285.

Sameer Pradhan, Noemie Elhadad, Brett R South, David Martinez, Lee M Christensen, Amy Vogel, Hanna Suominen, Wendy W Chapman, and Guergana K Savova. 2013. Task 1: Share/clef ehealth evaluation lab 2013. In CLEF (Working Notes), pages 212–31.

Dhruba Pujary, Camilo Thorne, and Wilker Aziz. 2020. Disease normalization with graph embeddings. In Proceedings of SAI Intelligent Systems Conference, pages 209–217. Springer.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

Nils Reimers, Iryna Gurevych, Nils Reimers, Iryna Gurevych, Nandan Thakur, Nils Reimers, Johannes Daxenberger, Iryna Gurevych, Nils Reimers, Iryna Gurevych, et al. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.

Kirk Roberts, Dina Demner-Fushman, and Joseph M Tonning. 2017. Overview of the tac 2017 adverse reaction extraction from drug labels track. In TAC.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. arXiv preprint arXiv:2001.07676.

Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J Mungall, et al. 2007. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. Nature biotechnology, 25(11):1251–1255.

Ryan Sullivan, Robert Leaman, and Graciela Gonzalez. 2011. The diego lab graph based gene normalization system. In 2011 10th International Conference on Machine Learning and Applications and Workshops, volume 2, pages 78–83. IEEE.

Mujeen Sung, Hwisang Jeon, Jinhyuk Lee, and Jaewoo Kang. 2020. Biomedical entity representations with synonym marginalization. arXiv preprint arXiv:2005.00239.

Damian Szklarczyk, John H Morris, Helen Cook, Michael Kuhn, Stefan Wyder, Milan Simonovic, Alberto Santos, Nadezhda T Doncheva, Alexander Roth, Peer Bork, et al. 2016. The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. Nucleic acids research, page gkw937.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In Proceedings of the 24th international conference on world wide web, pages 1067–1077.

Dustin Wright. 2019. NormCo: Deep disease normalization for biomedical knowledge base construction. University of California, San Diego.

Michael Ku Yu, Michael Kramer, Janusz Dutkowski, Rohith Srivas, Katherine Licon, Jason F Kreisberg, Cherie T Ng, Nevan Krogan, Roded Sharan, and Trey Ideker. 2016. Translation of genotype to phenotype by a hierarchy of cell subsystems. Cell systems, 2(2):77–88.

Sendong Zhao, Chang Su, Zhiyong Lu, and Fei Wang. 2021. Recent advances in biomedical literature mining. Briefings in Bioinformatics, 22(3):bbaa057.

A Appendix

A.1 Relations and phrases

Table 3 shows the relations among entities and their corresponding synonyms. The relation $\mathrm{identical}$ links a entity and a synonym to claim that the synonym refers to the entity. During training, the relation $\mathrm{identical}$ links $[\mathrm{MASK}]$ and a synonym or entity to extract the textual feature. Among other relations, $\mathrm{is\_a}$ is the most common relation, which describes the subsumption relation between a child entity and a parent entity. We transform these relations into phrases to put them in templates used by our Prompt-based model.

A.2 Implementation details

Details about prompt-based methods For prompt-based methods (Prompt, GraphPrompt (w/o $\mathcal{T}^2$), and GraphPrompt), we trained the model with $\mathcal{L}_\mathrm{c}$ for 400 iterations to warm-up entity embedding $\boldsymbol{x}_v$. For zero-shot setting, we followed the bi-encoder architecture that uses two encoders for entities and synonyms. Every time we updated the embeddings of entities $\boldsymbol{x}_{v'}, v' \in \mathcal{V}$, we had to run the encoder for every entity. For few-shot learning, we found that the entity embedding can be directly trained with an embedding layer. We used the entity side of the bi-encoder to generate entity embedding $\boldsymbol{x}^0_{v'}$, and used this embedding to initialize the embedding layer. Then we used embeddings from this trainable embedding layer to replace the $\mathrm{sg}(\boldsymbol{x}_{v'})$ and $\mathrm{sg}(\boldsymbol{x}_v)$ term in the loss.

Details about second-order GraphPrompt The second-order GraphPrompt (GraphPrompt in **Table 1**) actually didn't include zeroth-order and first-order templates, since we considered that they are sub-templates of second-order templates. We achieved this by padding a $[\mathrm{MASK}]$ neighbor. For example, $\mathcal{T}^0([\mathrm{MASK}], v)$ is implemented as $\mathcal{T}^2_{\mathrm{identical}}([\mathrm{MASK}], v, [\mathrm{MASK}])$, and $\mathcal{T}^1_r([\mathrm{MASK}], v)$ is implemented as $\mathcal{T}^2_r([\mathrm{MASK}], v, [\mathrm{MASK}])$. To get $\boldsymbol{x}_v$ and $\boldsymbol{y}_v$ from this template, you only need to ignore the output of the second mask.

Details about the base model The base model is a BioSyn-like model with some important modifications. We trained the model for 30 epochs with initial learning rate 1e-5, and decayed it to about

Table 3: Relations and phrases

Relation	Phrase	cl	fbbt	doid	mp	hp
identical	is identical with	✓	✓	✓	✓	✓
is_a	is a kind of	✓	✓	✓	✓	✓
capable_of	is capable of	✓				
negatively_regulates	negatively regulates	✓				
positively_regulates	positively regulates	✓				
regulates	regulates	✓				
part_of	is part of	✓	✓			
has_part	has	✓				
develops_from	develops from	✓	✓			
has_sensory_dendrite_in	has sensory dendrite in		✓			
sends_synaptic_output_to	sends synaptic output to		✓			
synapsed_to	is synapsed to		✓			
synapsed_by	is synapsed by		✓			
continuous_with	is continuous with		✓			
synapsed_via_type_Ib_bouton_to	is synapsed via type Ib bouton to		✓			
receives_synaptic_input_in	receives synaptic input in		✓			
overlaps	overlaps		✓			

1e-6 when the model converged. We used sparse features (Sung et al., 2020) during candidate generation. During encoding, we didn't add sparse features, since we found that sparse features had no significant impact on the results, and even caused a slight decrease of accuracy. Besides, we found that BioSyn (Sung et al., 2020) sometimes failed to retrieve positive candidates due to limited candidate size and the inaccuracy of the model. Therefore, we manually added positive candidate in order to make full use of training data. The procedure for inference is the same as BioSyn (Sung et al., 2020).

Details about baselines NormCo (Wright, 2019) was initially introduced to perform bio-entity linking with inputs being text corpora. Central to its proposed method is the modeling of coherence leveraging concept co-mentions in each text corpus. However, as NormCo is not designed to learn the semantics of concepts, it is not capable of zero-shot learning in our dataset. Therefore, we did not report its results under zero-shot setting.

In addition, to construct a coherence sequence analogous to the co-appearances of mentions in the original setting, we took the mentions (entities and synonyms) of neighbor concepts of each training concept (excluding validation and test mentions when training), where the mentions are arranged in order based on their distance from the central concept we build this sequence for.

As Sieve-Based (D'Souza and Ng, 2015) is a rule-based entity normalization method which does not need the training data, we treated the model as a zero-shot model. Besides, Sieve-Based does not include a scoring mechanism, so we could only report the results of Acc@1.