Enriching Medcial Terminology Knowledge Bases via Pre-trained Language Model and Graph Convolutional Network

Jiaying Zhang[1,+], Zhixing Zhang[1,+], Huanhuan Zhang[1], Zhiyuan Ma[1,*], Yangming Zhou[1,*] and Ping He[2]
[1]School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China
[2]Shanghai Hospital Development Center, Shanghai 200041, China
[*]Corresponding authors
[+]Equal contributor
Emails: yuliar3514@gmail.com, ymzhou@ecust.edu.cn

***Abstract*—Enriching existing medical terminology knowledge bases (KBs) is an important and never-ending work for clinical research because new terminology alias may be continually added and standard terminologies may be newly renamed. In this paper, we propose a novel automatic terminology enriching approach to supplement a set of terminologies to KBs. Specifically, terminology and entity characters are first fed into pre-trained language model to obtain semantic embedding. The pre-trained model is used again to initialize the terminology and entity representations, then they are further embedded through graph convolutional network to gain structure embedding. Afterwards, both semantic and structure embeddings are combined to measure the relevancy between the terminology and the entity. Finally, the optimal alignment is achieved based on the order of relevancy between the terminology and all the entities in the KB. Experimental results on clinical indicator terminology KB, collected from 38 top-class hospitals of Shanghai Hospital Development Center, show that our proposed approach outperforms baseline methods and can effectively enrich the KB.**

***Index Terms*—Knowledge base, Terminology enriching, Entity alignment, Pre-trained language model, Graph convolutional network**

I. Introduction

Recently, terminology knowledge bases (KBs) have attracted increasing attentions and are widely used in clinical domains. However, constructing medical terminology KBs cannot be done once and for all, and terminology enriching (see Fig. 1) never ends. The enriching is mainly caused by two reasons, namely terminology renaming and synonym adding. The former is common because the standard terminology names are not permanent, they will be replaced by more accurate names over time. For example, in the specimen of venous whole blood, the clinical indicator “血色素” used to be the traditional name of “血红蛋白” (hemoglobin, HGB) in Chinese. The latter owes to the fact that every region, even every hospital, has various names for the same terminology, and it is impossible to incorporate all synonyms into a single KB at once. For instance, collected from different hospitals, in the specimen of venous serum, the clinical indicator “泌乳素” (prolactin, PRL) may have 7 different synonymous names, namely “催乳素” (lactogen), “垂体泌乳素” (pituitary prolactin), “泌乳素测定” (prolactin measurement), “垂体催乳素” (pituitary lactogen) “催乳素(PRL)” (lactogen PRL), “垂体泌乳素(PRL)” (pituitary prolactin PRL) and “泌乳素(PRL)” (prolactin PRL). In this case, terminology enriching can be considered as a supplementary to existing KBs.

Existing well-known terminology KBs, such as SNOMED-CT [1], LONIC [2] and UMLS [3], usually enrich terminologies manually to ensure the professional authority, and they have a big team of experts. For example, over 350 individuals are devoted to the original work of SNOMED-CT [1], and 350 is a large number. Consequently, enriching the existing terminology KBs can not be a timely job, and these well-known KBs are typically enriched and released by few years. To solve the time-consuming and labor-intensive task, designing an automatic terminology enriching method is necessary.

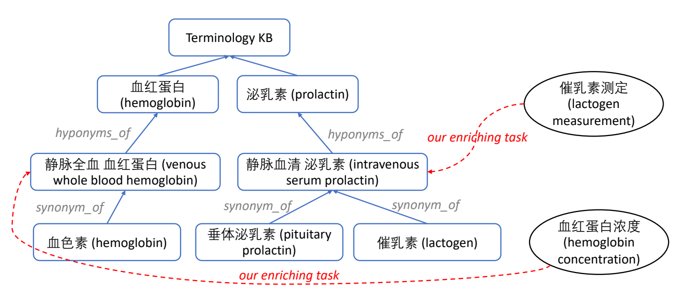

Fig. 1. An example for our terminology enriching task.

The most relevant work to this task is entity alignment. To achieve the alignment, conventional feature-based methods manually design various features [4], [5] and embedding-based methods encode KBs into embeddings [6], [7]. However, entity alignment aims to align KBs to other KBs, while terminology enriching need to align a set of terminologies to a terminology KB. It means that the existing entity alignment methods cannot be directly adopted in the enriching task. Moreover, pre-trained language representations, which are popular in many natural language processing (NLP) tasks, have not been employed in existing works. We believe that pre-trained model can further improve our task. Therefore, we try the pre-trained model to hot-start KB embedding and enhance semantic information.

In this paper, we propose a novel terminology enriching model to align a set of terminologies to a terminology KB via a pre-trained language model and graph convolutional network (GCN). Specifically, to predict the relevancy between a candidate terminology and an entity in KB, our model consists of three parts: (1) **BERT-based Semantic Embedding**. We learn the semantic relevancy between the entity and terminology through the pre-trained language model. (2) **GCN-based Structure Embedding**. We learn the structure relevancy using GCN. We firstly utilize BERT to initialize the representations of the entity and terminology, respectively, and then further optimize them through GCN. Finally, we calculate the relevancy of the two representations on element-level. (3) **Embeddings Integration**. We integrate the semantic embedding and the structure embedding for mutual fusion by the multi-Layer perception (MLP) model. The output of the model is the relevancy probability between the entity and terminology. For alignment prediction, we compute the relevancy between the terminology and all the entities in KB, and rank the entities by the relevancy to align. Our experiments are conducted on clinical indicator terminology KB[1] (see Fig. 2), which contains different clinical indicator names from 38 topcalss hospitals of Shanghai Hospital Development Center.

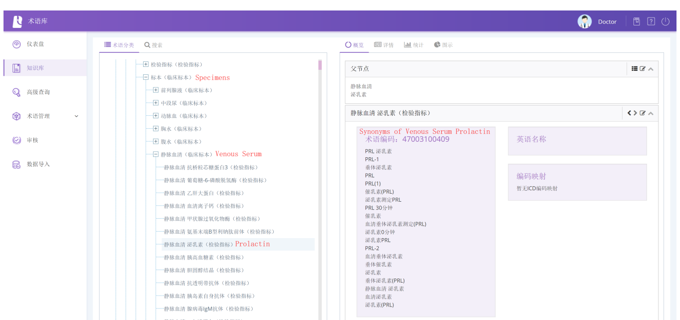

Fig. 2. The screenshot of the clinical indicator terminology KB.

The main contributions of this work can be summarized as follows.

- We propose a novel terminology enriching method to align a set of terminologies to a terminology KB for the first time. It sequentially integrates BERT-based semantic embedding and GCN-based structure embedding. It is the first time to introduce pre-trained language model to hot-start KB embedding and to enhance semantic information. We also adapt GCN to our enriching task.
- Experimental results show that our proposed model achieves better performance than other baseline methods. In addition, we also show that introducing pre-trained model has a great improvement than pure GCN-based alignment methods.

[1]We show a demo in http://dcakb.ecustnlplab.com/

The rest of the paper is organized as follows. Section II briefly reviews the related work on entity alignment, pre-trained language model and KB embedding. Section III introduces terminology enriching task. In Section IV, we detail the proposed model. Exerperimental results are described in Section V. Finally, the paper is concluded in Section VI.

II. Related Work

A. Entity Alignment

The most relevant work to the task of terminology enriching is entity alignment.

Earliest works manually aligned entities. In order to reduce workload, various crowdsourcing algorithms were applied [8]–[10] and the alignment quality was promised. Some other conventional feature-based works observe KBs manually and carefully designed various features, such as literal information [11], external lexicons [4], [12] and attribute values [5], [13]. The effectiveness of these methods largely depended on human experience.

Recently, with the emergence of semantic representation learning, many embedding-based methods were proposed, which embedded KBs and achieved the alignment with these embeddings. There existed four basic ideas: MTransE [6] encoded KBs in separated embedding spaces and a transformation was learned to align them. JE [14] jointly learned embeddings in a unified space. JAPE [15] introduced attribute embedding in addition to structure embedding. GCN-Align [7] generated entity representation based on neighborhood information and attribute information. Many other works [16]–[18] had been extended according to these four ideas. For example, Zhu et al. [19] proposed an iterative and parameter sharing method, which encoded both entities and relations of heterogeneous KBs by TransE and PTransE to obtain knowledge embeddings, and joined these embeddings into a unified semantic space. Zhang et al. [20] proposed a multi-view embedding method, including name view, relation view and attribute view, and learned their embeddings by Skip-gram, TransE and Convolutional Neural Networks (CNNs) before combining them together for alignment. Pang et al. [21] improved GCN-Align by considering both local and global information of attribute representation, incorporated neighbouring attributes as local information, and discarded most frequent attributes as global information.

However, there are three major differences between the above approaches and our task. Firstly, the above methods performed entity alignment between different KBs, but what our enriching task aligns to a terminology KB is a set of terminologies, meaning that we cannot directly adopt the existing methods. Secondly, instead of random initialization, we use a pre-trained language model to hot-start KB embedding. Thirdly, we also employ the pre-trained language model to enhance semantic information.

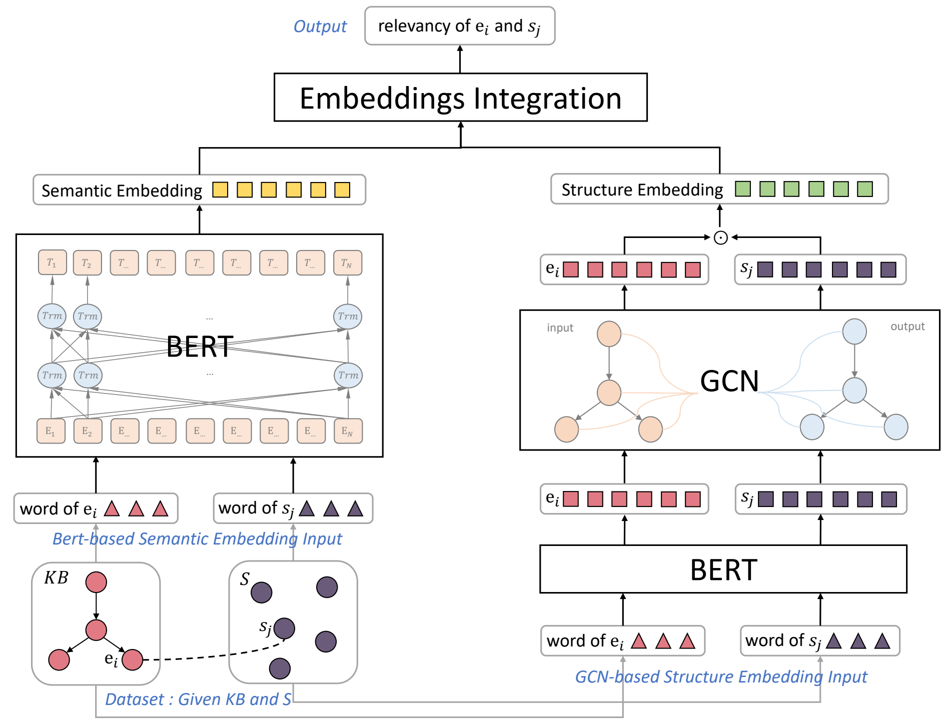

Fig. 3. Overview of the model for terminology enriching.

B. Pre-trained Language Model

With the popularity of pre-trained language models in many NLP tasks, the existing methods for applying their pre-trained models to downstream tasks can be divided into two classes: feature-based and fine-tuning. The feature-based methods encoded words into representations and fed these pre-trained representations into tasks-specific architectures as input embeddings, such as ELMo [22]. The fine-tuning methods directly trained their pre-trained models on the downstream tasks and provided task-specific parameters for fine-tuning, such as BERT [23], GPT [24], ERNIE [25] and XLNet [26]. In this paper, we choose BERT as our pre-trained model for its state-of-art performance.

C. KB Embedding

KB embedding has been considered as an effective way to encode components of KB including entities and relations into a low-dimensional vector space without losing inherent information [27]. TransE [28] was the most representative method, which interpreted the relation as a translation vector from the head entity to the tail one. TransH [29], TransR [30] and TransD [31] were successively proposed to improve TransE in dealing with multi-mapping relations. However, these translation-based embedding models required aligned or shared relations. As another solution, neural-based embedding models were proposed by exploiting deep learning techniques [20], such as MLPs [32], CNNs [33], and GCNs [34]. In this paper, we use GCN for KB embedding.

III. Problem Formulation

Formally, the existing terminology knowledge base is defined as $KB = (E, R)$, where E denotes the set of entities (i.e. terminologies) and R denotes the set of relations between entities. Each knowledge can be described by one of the following two triples: $T^{Syn} = \{(h,r,t)|h,t \in E, r = synonymous\_of \in R\}$ and $T^{Hyp} = \{(h,r,t)|h,t \in E, r = hyponyms\_of \in R\}$. For example, in the specimen of venous serum, (催乳素, $synonym\_of$, 静脉血清泌乳素) means that the clinical indicator “催乳素” (lactogen) is a synonym of “静脉血清泌乳素” (intravenous serum prolactin), and (泌乳素, $hyponyms\_of$, 静脉血清泌乳素) means that “静脉血清泌乳素”(intravenous serum prolactin) belongs to “泌乳素”.

Given existing terminology knowledge base KB and a set of candidate terminologies $S = \{s_1, s_2, \ldots, s_m\}$ to be updated, where m is the number of terminologies, the task is to automatically pick the *synonymous* pair set $P = \{(e_i, s_j)|e_i \in E, s_j \in S\}$ and align s_j to e_i respectively.

IV. Proposed Model

In this section, we present our proposed model for terminology enriching. Given the terminology KB and a candidate terminology s_j, each entity in KB is sequentially extracted to calculate a relevancy with s_j, and then the optimal alignment results are ranked by relevancy.

The overview model to obtain the relevancy of an entity e_i in KB and s_j is shown in Fig. 3, in which the relevancy is computed by integrating semantic information and structural information into MLP. To get the semantic embedding, pre-trained language model BERT is utilized. To get the structure embedding, GCN is used. The representations of e_i and s_j are initialized by BERT, and further embedded through GCN respectively. The relevancy embedding with structural information is obtained by the dot production of the two representations. Each stage is trained separately and sequentially.

A. BERT-based Semantic Embedding

Intuitively, synonymous terminologies are assumed to share similar semantics in context. Based on the assumption, pre-trained language representation models have been proved to be effective in capturing semantic relevancy and work as a routine component in many NLP tasks [25]. In this paper, BERT [23] is chosen to represent semantic embedding, the structure of which are shown in Fig. 4.

BERT Embedding Model. To be specific, given an entity e_i in the terminology KB and a candidate terminology s_j in S, these two inputs are merged into a sequence {[CLS] x [SEP] y [SEP]}, where $x = s_j$, $y = e_i$, and they are word-level tokenized. For each token in the sequence, its corresponding token embedding, segment embedding and positional embedding are summed as the input embedding.

The BERT model consists of multiple bidirectional Transformer encoder layers. The output of the l-th layer is the input of the $l+1$-th layer, and the semantic embedding is computed as follows:

$$\tilde{\boldsymbol{H}}^{(l)} = LayerNorm(\boldsymbol{H}^{(l)} + MhAtt(\boldsymbol{H}^{(l)})) \quad (1)$$

$$\boldsymbol{H}^{(l+1)} = LayerNorm(\tilde{\boldsymbol{H}}^{(l)} + FFN([\tilde{\boldsymbol{H}}^{(l)}]_+)) \quad (2)$$

where MhAtt($\cdot$) is the multi-head self-attention [35], LayerNorm($\cdot$) is the layer normalization [36] and $[x]_+ = \max\{0, x\}$ represents the maximum between 0 and x.

Then, the [CLS] token embedding is taken as BERT-based semantic embedding, which represents the relevancy of e_i and s_j.

Loss Function. To fine-tune BERT, we use a set of known *synonymous* pair set P^+, and P^- stands for the negative sample set of P^+. If $(e_i, s_j) \in P^+$ (or $(e_i^{'}, s_j^{'}) \in P^-$), we define $y_{se} = 1$ (or 0). Then we put BERT-based semantic embedding into the sigmoid activation function to get $\hat{y}_{se}$.

When training the model, the optimizing objective is to minimize the binary cross-entropy function:

$$L_{se} = -y_{se}\log(\hat{y}_{se}) - (1 - y_{se})\log(1 - \hat{y}_{se}) \quad (3)$$

B. GCN-based Structure Embedding

BERT-based semantic embedding model only utilize entity information without considering the structure of whole graph. However, relations in a KB usually offer additional guidelines for the model and synonyms are non-ignorable in a terminology KB. GCNs are a type of neural network that can produce a node-level embedding using the structure of a graph and embedding of nodes, so that neighborhood information such

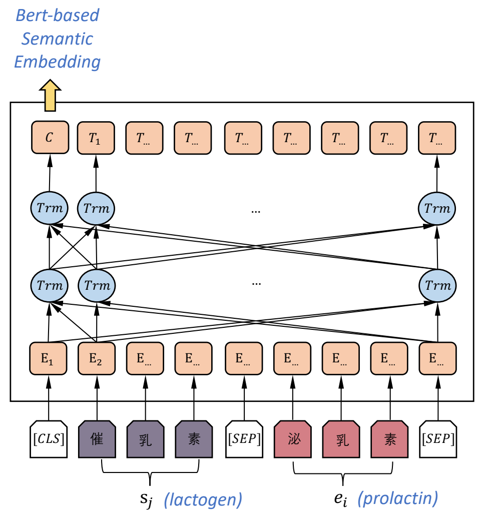

Fig. 4. BERT model structure for semantic embedding. Note that the figure is used to show the model structure, the entity “静脉血清泌乳素” (intravenous serum prolactin) is replaced by the “泌乳素” (prolactin) for the drawing reason.

as synonym embedding and hyponymy relation can be encoded through an end-to-end learning approach.

GCN Embedding Model. Note that synonyms describe entities alternatively and hyponymy relation offers categorical information of the entity. GCN is able to combine these information and projects the entities into the same continuous vector space where entities and their synonyms are close to each other.

Our GCN model consists of L GCN layers, and the l-th GCN layer has two inputs. One is an $n \times d^{(l)}$ node embedding matrix $\boldsymbol{H}^{(l)}$, where n is the number of nodes and $d^{(l)}$ represents the node embedding dimension of the l-th layer. The other is the adjacent matrix $\boldsymbol{A}$ which is an $n \times n$ matrix. The l-th layer uses the following normalized graph Laplacian transformation [37] to obtain the output.

$$\boldsymbol{H}^{(l+1)} = \sigma(\hat{\boldsymbol{D}}^{-\frac{1}{2}}\hat{\boldsymbol{A}}\hat{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}) \quad (4)$$

where $\sigma(\cdot)$ is the non-linear activation function, $\boldsymbol{A}$ is the $n \times n$ adjacent matrix describing the connectivity of the KG, and $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ where $\boldsymbol{I}$ represents the identity matrix. $\hat{\boldsymbol{D}}$ is a diagonal matrix with entries $\hat{\boldsymbol{D}}_{ii} = \Sigma_j \hat{\boldsymbol{A}}_{ij}$. $\boldsymbol{W}^{(l)}$ is the weight matrix of the l-th layer.

Node Embedding. In the GCN embedding model, $\boldsymbol{H}^{(0)}$ is initialized by the node embedding generated by BERT. Specifically, BERT is firstly fine-tuned using the aforementioned method, then we transformed the entities e_i and candidate s_j terminologies into the specific sequence {[CLS] x [SEP][SEP]}, i.e. we set $x = e_i$, $x = s_j$ separately. Then we utilize the above-mentioned model to compute the embedding of every word. Note that entities and candidate terminologies are both treated as nodes in GCN model for GCN can only handle node inputs, where candidate terminologies can be seen as isolated nodes in graph. Finally the token embedding of

[CLS] is taken out as node embedding, since y in the aforementioned sequence is set empty, it only contains information of x.

Adjacent Matrix. There are two types of adjacent matrices for entities and the terminologies, respectively.

Unlike Wang et al. [7] designed particular connectivity matrix, for entities in terminology KB, we simply set entries in adjacent matrix $\boldsymbol{A}_{ij}$ to 1 when an edge from entity e_j to entity e_i exists. The reason is, there are two relations, namely 'synonym of' and 'hyponyms of', which we believe deliver important information from entity e_j to entity e_i equally.

For candidates terminologies, we use an all-zero matrix as adjacent matrix, indicating no edge exists between candidates terminologies.

Relevancy Embedding with Structural Information. The output of the L-th GCN layer are the node embeddings of entities and candidate terminologies. We then proceed with an element-wise multiply operation on these two node embeddings and obtain GCN-based structure embedding, which is the relevancy embedding with structural information.

Loss Function. In order to project the node embedding into the vector space where entities and their corresponding candidate terminologies are close to each other, we utilize margin-based distance loss functions to optimize the problem. The distance definition are shown in Fig. 5, pairs of entity node embeddings and candidate terminology node embeddings are taken out to calculate the distance. We define the n-th moment distance function as:

$$D(e_i, s_j) = \|e_i - s_j\|_n \tag{5}$$

And the loss function is:

$$L = \sum_{(e_i,s_j)\in P^+,(e_i^{'},s_j^{'})\in P^-} [D(e_i, s_j) + \gamma - D(e_i^{'}, s_j^{'})]_+ \tag{6}$$

where $\gamma > 0$ is the hyper-parameter which represents the margin between positive samples and negative samples. We adopt Adam [38] to minimize the loss function, in order to minimize the distance between positive pair while maximizing the distance between negative pair.

C. Embeddings Integration

Both the semantic relevancy and the structure relevancy contain important information for alignment, an MLP model is adopted for mutual fusion of the semantic embedding $\boldsymbol{X}_{se}$ and the structure embedding $\boldsymbol{X}_{st}$.

MLP Model. The output of the hidden layers are represented as follows:

$$\boldsymbol{H}^{(0)} = [\boldsymbol{X}_{se}; \boldsymbol{X}_{st}] \tag{7}$$

$$\hat{\boldsymbol{H}}^{(l)} = \sigma(\boldsymbol{W}^{(l)}\boldsymbol{H}^{(l)} + \boldsymbol{b}^{(l)}) \tag{8}$$

$$\boldsymbol{H}^{(l+1)} = \sigma(\hat{\boldsymbol{W}}^{(l)}\hat{\boldsymbol{H}}^{(l)} + \hat{\boldsymbol{b}}^{(l)}) \tag{9}$$

$$\boldsymbol{H}^{(f)} = sigmoid(\boldsymbol{W}^{(f-1)}\boldsymbol{H}^{(f-1)} + \boldsymbol{b}^{(f-1)}) \tag{10}$$

where $\boldsymbol{H}^{(0)}$ is the input and $\boldsymbol{H}^{(f)}$ is the output.

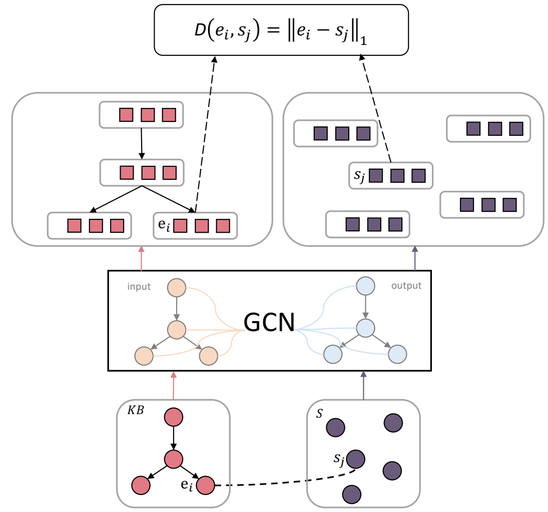

Fig. 5. GCN model structure for structure embedding.

Loss Function. Similar to BERT, if $(e_i, s_j) \in P^+$ (or $(e_i^{'}, s_j^{'}) \in P^-$), $y_{in} = 1$ (or 0) and $\hat{y}_{in} = \boldsymbol{H}^{(f)}$. The loss function is still the binary cross-entropy function:

$$L_{in} = -y_{in}\log(\hat{y}_{in}) - (1 - y_{in})\log(1 - \hat{y}_{in}) \tag{11}$$

V. Experimental Analysis

In this section, to evaluate the effectiveness of our model, we compare our method with basic methods, feature-based methods and embedding-based methods. To evaluate the importance of different components in our model, we vary our model in different ways, including ablation analysis, varying data size and model hyper-parameters, measuring the changes in performance of terminology enriching.

A. Dataset

Clinical indicator terminology KB [11], which contains different clinical indicator names from 38 top-class hospitals of Shanghai Hospital Development Center, are included in our experiments. The terminology KB contains 15,960 entities, including 3,636 standard terminology names and 12,324 terminology synonyms, and 17,930 relations including 5,606 hyponymy relations and 12,324 description relations.

Table I outlines the detail statistics of our dataset. We choose 743 standard terminology names which contains at least three synonyms so that we can take full advantage of structure embedding. Subsequently, we split the corresponding synonyms into two parts, namely KB structure set and candidate set in ratio of 2:8. We take KB structure set as entities in graph. Candidate set is removed from KB and treated as the set of candidate terminologies S. We further split the candidate set into training set, validation set and test set in ratio of 3:2:5. The KB we used in our experiment has 2,554 nodes, and the number of standard terminology names is 743. In total, there are 1,343 candidate terminologies in the training set, 895 in the validation set and 2,239 in the test set.

TABLE I
STATISTICS OF THE KB AND THE CANDIDATE TERMINOLOGIES

Dataset	Type	Total
KB	Standard Terminology	743
	Terminology Synonym	1,212
	Hyponyms Relation	1,486
	Synonym Relation	1,212
Candidate Terminology	Training Set	1,343
	Test Set	2,239
	Validation Set	895

B. Experiment Settings

The model is implemented using Keras [39] with TensorFlow [40] backend run on NVIDIA GeForce GTX 1080Ti GPU. All three parts in the model are optimized by Adam. Due to the high cost of pre-train BERT and lack of large scale corpus, we directly adapt parameters pre-trained in Chinese by Google. During fine-tuning process, we lock first 11 layer and train the last layer only. BERT is optimized in learning rate of 5×10^{-5}. Learning rate of GCN model is 1×10^{-5}, layer output dimension is set to 768. The embedding integration model is optimized in learning rate of 1×10^{-4}.

We use *Hit@k*, which is the percentage of properly aligned entities ranked in the top k candidates, as metric, and we take k in values of 1, 5, 10.

C. Comparison with State-of-the-art Methods

We firstly compare our proposed model with one basic method and four state-of-the-art methods published in the last year. The basic method firstly preprocess the data, including special character replacement and abbreviation separation before utilizing longest common subsequence threshold to filter synonyms. Besides the basic method, the state-of-the-art methods tried to design feature vectors or obtain KB embeddings. For example, Zhang et al. [11] combined different character similarity algorithms (e.g. cosine similarity), and trained a binary classifier to find synonyms. Wang et al. [12] used a knowledge graph for both hypernymy and synonym detection between Chinese symptoms. Wang et al. [7] generated entity representation and attribute representation based on neighborhood information. Pang et al. [21] improved Wang et al. [7] by considering both local and global information of attribute representation. Note that there is neither attribute information nor external lexicons in our data, these existing methods are adjusted and then applied to the experiments.

The experimental results are shown in Table II. From this table, we clearly observe that our model achieves much higher scores among all these reference algorithms with the Hits@1 of 59.58%, the Hits@5 of 84.01% and the Hits@10 of 87.63%, which means the combination of the pre-trained model and GCN in our model is complementary. The introduction of pre-trained representations can make full use of both the unsupervised pre-training and supervised training data for better enriching. GCN can effectively utilize the structural characteristics of KB and integrate the neighbor information

TABLE II
EXPERIMENTAL RESULTS OF STATE-OF-THE-ART METHODS AND OUR PROPOSED MODEL

Methods	Hits@1	Hits@5	Hits@10
Basic method	20.10	50.92	63,96
Zhang et al. [11]	45.42	74.90	80.88
Wang et al. [12]	14.60	17.95	18.22
Wang et al. [7]* Pang et al. [21]*	31.76	53.68	61.81
Our model	**59.58**	**84.01**	**87.63**

* Since Pang et al. [21] improved the attribute embedding, which is missing in our data, of Wang et al. [7], these two methods share the same scores.

of the entity into its own representation. Meanwhile, among all baselines, Zhang et al. [11] is the strongest one while Wang et al. [12] is the worst. The reason for the phenomenon is that a small amount of training data is enough for literal similarity calculated by Zhang et al. [11], and the performance of Wang et al. [12] depends largely on their knowledge graph, which does not cover enough required knowledge in the experiments.

D. Ablation Analysis

To investigate the importance of model components, we explore the effects of the BERT and GCN for our model. In addition, we also study the effects of different initialization of GCN representations in our model.

TABLE III
COMPARATIVE RESULTS FOR ABLATION ANALYSIS OF OUR PROPOSED MODEL

Components	Hits@1	Hits@5	Hits@10
Our model	**59.58**	**84.01**	**87.63**
- w/o BERT*	40.24	68.11	72.85
- w/o GCN*	52.88	81.60	86.69
- GCN w/o initialized by BERT°	49.04	81.69	86.78
- GCN initialized by BERT w/o fine-tuning°	56.41	83.11	87.32

* **w/o BERT** and **w/o GCN** refer to our model without BERT as semantic embedding and GCN as structure embedding respectively.
° **GCN w/o initialized by BERT** refers to initialization of GCN representations is generated randomly without BERT. **GCN initialized by BERT w/o fine-tuning** refers to that BERT is used to initialize GCN representation, but fine-tuning process is omitted and original BERT pre-training parameters remain unchanged.

As demonstrated in Table III, we have the following observations: (1) Both BERT and GCN play important roles in our model. The pre-trained language model BERT show its ability to disambiguate, and can improve Hits@k score by more than 15%. GCN model can capture structural information contained in the graph, so that it can help to improve Hits@1 score to 59.58%. (2) Since the most nodes in terminology KB are synonyms and lack of distinctive relations, BERT node embedding without fine-tuning can improve Hits@1 score from random initialized node embedding from 49.04% to 56.41%, which is more than 7%, and specific fine-tuning can further improve Hits@1 score of our proposed model by more than 3%.

E. Impacts of Different Sizes of Training Data

To study how the size of training set influences the performance and test the scalability of our model, we use different proportions of the training data and calculate the Hits@k scores. Intuitively, the more training data are used, the better results can be obtained. In this paper, considering the total number of training data and the cost of time, we pick 5%, 10%, 20%, 50%, 75% of the training data and summarize the comparative results in Fig. 6.

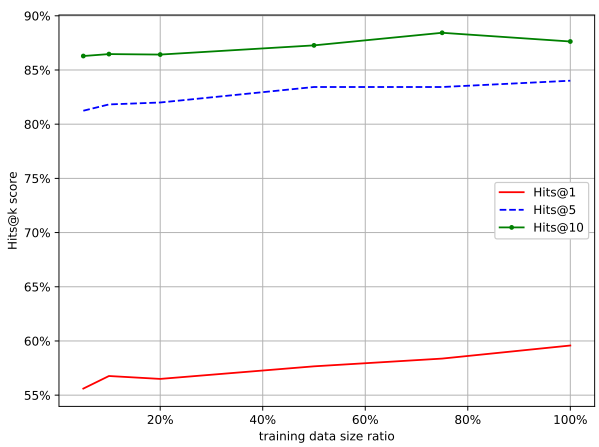

Fig. 6. Hit@k score in different data size.

From the results of different data size in Fig. 6 we can observe that our model performs better as the training data size increases, although the score improves slightly and the increasing rate is low. The results show that the model can perform well in small training data. This is reasonable because our model uses pre-trained representations to enhance semantic embedding and hot-start structure embedding. Additionally, note that our test set is much larger than training set, it also proves that our model can enrich the terminology KB with a small amount of labeled training data.

F. Impacts of Different Graph Convolutional Networks

1) Impacts of Different GCN Layers: We investigate the influence of different GCN layer number by running the proposed model on layer numbers L of 1, 2 and 3. Specifically, GCN layers except the last layer use activation function $[\cdot]_+$, and the last layer uses no activation function to output node embedding. Table IV presents the comparative results.

TABLE IV
COMPARATIVE RESULTS FOR DIFFERENT GCN LAYERS

GCN Layers	Hits@1	Hits@5	Hits@10
$L = 1$	**59.58**	**84.01**	**87.63**
$L = 2$	56.19	82.85	86.91
$L = 3$	54.35	82.80	86.55

From Table IV, we can observe that our proposed model gets the best performance when GCN layer number $L = 1$, which owes to the fact that complex GCN model will overfit the train data while model is hot-started by pre-trained language model. Therefore, we set hyper-parameter $L = 1$ in the rest of our experiments.

2) Impacts of Different GCN Loss Functions: To further analyze the impacts of the margin and the distance function used in the GCN loss function, we compare the performances of models with different parameter values. Experimental results are displayed in Table V.

TABLE V
COMPARATIVE RESULTS FOR DIFFERENT GCN LOSS FUNCTIONS

Distance Function	First Moment			Second Moment		
Margin	Hits@1	Hits@5	Hits@10	Hits@1	Hits@5	Hits@10
$\gamma = 3$	59.36	**84.46**	88.25	59.04	83.92	88.30
$\gamma = 5$	**59.58**	84.01	87.63	59.45	84.41	**88.34**

From Table V, we can observe that the best Hits@1 score is obtained when $\gamma = 5$ and the first moment is used. Meanwhile, Hits@5 is better when $\gamma = 3$ while Hits@10 is better when the second moment is used. Overall, margin γ and distance function affect the score in a small scale, and GCN model shows robustness in these hyper-parameters. As we pay more attention to the Hits@1, we choose $\gamma = 5$ and the first moment in GCN loss function for the best Hits@1 score.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel terminology enriching method which aligns a set of terminologies to a terminology KB based on semantic embedding learned by BERT and structure embedding learned by GCN. These two embeddings are integrated to measure the relevancy of the terminology and the entity. The optimal alignment is acquired by ranking the relevancy between the terminology and all the entities in the KB. Our approach is the first one to make use of pre-trained language model to hot-start KB embedding and to enhance semantic information, and adapt GCN to our task. We evaluate our method on clinical indicator terminology KB, collected from 38 top-class hospitals of Shanghai Hospital Development Center, and experimental results show the advantages of our proposed model over the compared baselines and the ability to enrich the KB.

For future work, we plan to explore more advanced pre-trained models and GCN models to improve our proposed method. Also, we will apply our approach to other types of terminology enriching, such as disease, operation and symptom.

ACKNOWLEDGMENT

We want to thank Jun Wang (Shanghai SimMed Technology Limited Company) who has helped us a lot in the construction of clinical indicator terminology KB. This work is supported by the National Key R&D Program of China for "Precision Medical Research" (No. 2018YFC0910500).

References

[1] K. Donnelly, "Snomed-ct: The advanced terminology and coding system for ehealth," *Studies in Health Technology and Informatics*, vol. 121, p. 279, 2006.

[2] C. J. McDonald, S. M. Huff, J. G. Suico, G. Hill, D. Leavelle, R. Aller, A. Forrey, K. Mercer, G. DeMoor, J. Hook *et al.*, "Loinc, a universal standard for identifying laboratory observations: A 5-year update," *Clinical Chemistry*, vol. 49, no. 4, pp. 624–633, 2003.

[3] O. Bodenreider, "The unified medical language system (umls): Integrating biomedical terminology," *Nucleic Acids Research*, vol. 32, no. suppl_1, pp. 267–270, 2004.

[4] Z. Wang, J. Li, and J. Tang, "Boosting cross-lingual knowledge linking via concept annotation," in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2013, pp. 2733–2739.

[5] F. M. Suchanek, S. Abiteboul, and P. Senellart, "Paris: Probabilistic alignment of relations, instances, and schema," *Proceedings of the VLDB Endowment*, vol. 5, no. 3, pp. 157–168, 2011.

[6] M. Chen, Y. Tian, M. Yang, and C. Zaniolo, "Multilingual knowledge graph embeddings for cross-lingual knowledge alignment," in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 1511–1517.

[7] Z. Wang, Q. Lv, X. Lan, and Y. Zhang, "Cross-lingual knowledge graph alignment via graph convolutional networks," in *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 349–357.

[8] S. Wang, X. Xiao, and C.-H. Lee, "Crowd-based deduplication: An adaptive approach," in *Proceedings of ACM Special Interest Group on Management of Data (SIGMOD)*, 2015, pp. 1263–1277.

[9] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: Is the problem solved?" *Proceedings of the VLDB Endowment*, vol. 10, no. 5, pp. 541–552, 2017.

[10] Y. Zhuang, G. Li, Z. Zhong, and J. Feng, "Hike: A hybrid human-machine method for entity alignment in large-scale knowledge bases," in *Proceedings of ACM on Conference on Information and Knowledge Management (CIKM)*, 2017, pp. 1917–1926.

[11] J. Zhang, Q. Wang, Z. Zhang, Y. Zhou, Q. Ye, H. Zhang, J. Qiu, and P. He, "An effective standardization method for the lab indicators in regional medical health platform using n-grams and stacking," in *Proceedings of IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2018, pp. 1602–1609.

[12] Q. Wang, T. Wang, and C. Xu, "Using a knowledge graph for hypernymy detection between chinese symptoms," in *Proceedings of International Conference on Advanced Computational Intelligence (ICACI)*, 2018, pp. 601–606.

[13] H. Wang, Z. Fang, L. Zhang, J. Z. Pan, and T. Ruan, "Effective online knowledge graph fusion," in *Proceedings of International Semantic Web Conference (ISWC)*, 2015, pp. 286–302.

[14] Y. Hao, Y. Zhang, S. He, K. Liu, and J. Zhao, "A joint embedding method for entity alignment of knowledge bases," in *Proceedings of China Conference on Knowledge Graph and Semantic Computing*, 2016, pp. 3–14.

[15] Z. Sun, W. Hu, and C. Li, "Cross-lingual entity alignment via joint attribute-preserving embedding," in *Proceedings of International Semantic Web Conference (ISWC)*, 2017, pp. 628–644.

[16] Z. Sun, W. Hu, Q. Zhang, and Y. Qu, "Bootstrapping entity alignment with knowledge graph embedding." in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 4396–4402.

[17] B. D. Trsedya, J. Qi, and R. Zhang, "Entity alignment between knowledge graphs using attribute embeddings," in *Proceedings of Association for the Advance of Artificial Intelligence (AAAI)*, 2019, pp. 297–304.

[18] M. Chen, Y. Tian, K.-W. Chang, S. Skiena, and C. Zaniolo, "Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment," in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAL)*, 2018, pp. 3998–4004.

[19] H. Zhu, R. Xie, Z. Liu, and M. Sun, "Iterative entity alignment via joint knowledge embeddings." in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 4258–4264.

[20] Q. Zhang, Z. Sun, W. Hu, M. Chen, L. Guo, and Y. Qu, "Multi-view knowledge graph embedding for entity alignment," in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, pp. 5429–5435.

[21] N. Pang, W. Zeng, J. Tang, Z. Tan, and X. Zhao, "Iterative entity alignment with improved neural attribute embedding," in *Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG) Co-located with Extended Semantic Web Conference (ESWC)*, 2019, pp. 41–46.

[22] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2018, pp. 2227–2237.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2019, pp. 4171–4186.

[24] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," in *Proceedings of Technical Report, OpenAI*, 2018.

[25] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "Ernie: Enhanced language representation with informative entities," in *Proceedings of Annual Meeting of the Association for Computational Linguistics Conference (ACL)*, 2019, pp. 1441–1451.

[26] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.

[27] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724–2743, 2017.

[28] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 2787–2795.

[29] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of Association for the Advance of Artificial Intelligence (AAAI)*, 2014.

[30] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of Association for the Advance of Artificial Intelligence (AAAI)*, 2015, pp. 2181–2187.

[31] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL) and International Joint Conference on Natural Language Processing (IJCNLP)*, 2015, pp. 687–696.

[32] B. Shi and T. Weninger, "Proje: Embedding projection for knowledge graph completion," in *Proceedings of Association for the Advance of Artificial Intelligence (AAAI)*, 2017, pp. 1236–1242.

[33] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of Association for the Advance of Artificial Intelligence (AAAI)*, 2018, pp. 1811–1818.

[34] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proceedings of European Semantic Web Conference (ESWC)*, 2018, pp. 593–607.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5998–6008.

[36] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *CoRR*, vol. abs/1607.06450, 2016.

[37] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2002, pp. 585–591.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

[39] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[40] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.