KGSynNet: A Novel Entity Synonyms Discovery Framework with Knowledge Graph

Yiying Yang[1§], Xi Yin[1§], Haiqin Yang[1*], Xingjian Fei[1], Hao Peng[2*], Kaijie Zhou[1], Kunfeng Lai[1], and Jianping Shen[1]

[1] Ping An Life Insurance Company of China, Ltd.
{yangyiying283, yinxi445, feixingjian568, zhoukaijie002, laikunfeng597, shenjianping324}@pingan.com.cn; *hqyang@ieee.org
[2] BDBC, Beihang University, Beijing, China
*penghao@act.buaa.edu.cn

Abstract. Entity synonyms discovery is crucial for entity-leveraging applications. However, existing studies suffer from several critical issues: (1) the input mentions may be out-of-vocabulary (OOV) and may come from a different semantic space of the entities; (2) the connection between mentions and entities may be hidden and cannot be established by surface matching; and (3) some entities rarely appear due to the long-tail effect. To tackle these challenges, we facilitate knowledge graphs and propose a novel entity synonyms discovery framework, named *KGSynNet*. Specifically, we pre-train subword embeddings for mentions and entities using a large-scale domain-specific corpus while learning the knowledge embeddings of entities via a joint TransC-TransE model. More importantly, to obtain a comprehensive representation of entities, we employ a specifically designed *fusion gate* to adaptively absorb the entities' knowledge information into their semantic features. We conduct extensive experiments to demonstrate the effectiveness of our *KGSynNet* in leveraging the knowledge graph. The experimental results show that the *KGSynNet* improves the state-of-the-art methods by 14.7% in terms of hits@3 in the offline evaluation and outperforms the BERT model by 8.3% in the positive feedback rate of an online A/B test on the entity linking module of a question answering system.

Keywords: Entity synonyms discovery · Knowledge graph · Fusion gate

1 Introduction

Entity synonyms discovery is crucial for many entity-leveraging downstream applications such as entity linking, information retrieval, and question answering (QA) [19, 28]. For example, in a QA system, a user may interact with a chatbot as follows:

User query: Am I qualified for the new insurance policy as I suffer from **skin relaxation** recently?

§ Equal contribution. * Corresponding authors.

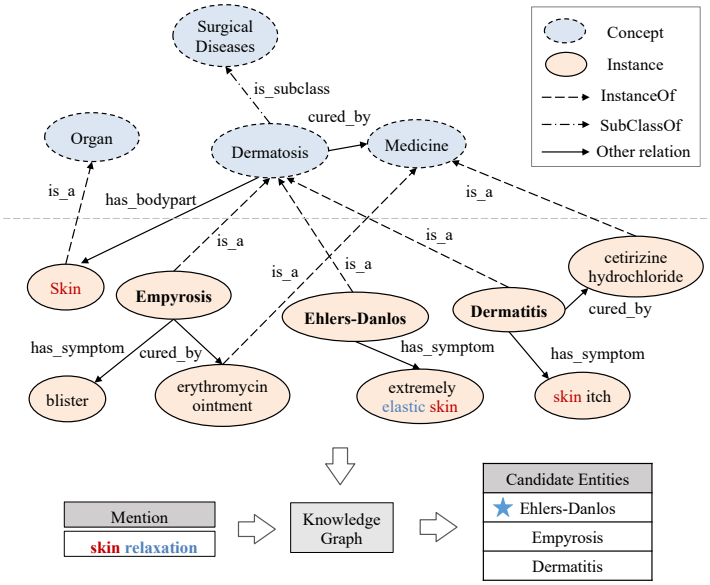

Fig. 1. An illustration of linking the synonymous entity of the mention "skin relaxation" to "Ehlers-Danlos" with the help of an external knowledge graph.

System reply: Unfortunately, based on the policy, you may fall into the terms of **Ehlers-Danlos**, which may exclude your protection. Please contact our agents for more details.

In this case, we can correctly answer the user's query only linking the mention of "skin relaxation" to the entity, "Ehlers-Danlos". This is equivent to the entity synonyms discovery task, i.e., automatically identifying the synonymous entities for a given mention or normalizing an informal mention of an entity to its standard form [8, 26].

In the literature, various methods, such as DNorm [15], JACCARD-based methods [27], and embedding-based methods [6, 11], have been proposed to solve this task. They usually rely on matching of syntactic string [8, 27] or lexical embeddings [6, 11, 25] to build the connections. Existing methods suffer from the following critical issues: (1) the input mentions and the entities are often out-of-vocabulary (OOV) and lie in different semantic spaces since they may come from different sources; (2) the connection between mentions and entities may be hidden and cannot be established by surface matching because they scarcely appear together; and (3) some entities rarely appear in the training data due to the long-tail effect.

To tackle these challenges, we facilitate knowledge graphs and propose a novel entity synonyms discovery framework, named *KGSynNet*. Our *KGSynNet* resolves the OOV issue by pre-training the subword embeddings of mentions and entities using a domain-specific corpus. Moreover, we develop a novel TransC-

TransE model to jointly learn the knowledge embeddings of entities by exploiting the advantages of both TransC [17] in distinguishing concepts from instances and TransE [4] in robustly modeling various relations between entities. Moreover, a *fusion gate* is specifically-designed to adaptively absorb the knowledge embeddings of entities into their semantic features. As illustrated in Fig. 1, our *KGSynNet* can discover the symptom of "extremely elastic skin" in the entity of "Ehler-Danlos" and link the mention of "skin relaxation" to it.

In summary, our work consists of the following contributions:

- We study the task of automatic entity synonyms discovery, a significant task for entity-leveraging applications, and propose a novel neural network architecture, namely *KGSynNet*, to tackle it.
- Our proposed *KGSynNet* learns the pre-trained embeddings of mentions and entities from a domain-specific corpus to resolve the OOV issue. Moreover, our model harnesses the external knowledge graph by first encoding the knowledge representations of entities via a newly proposed TransC-TransE model. Further, we adaptively incorporate the knowledge embeddings of entities into their semantic counterparts by a specifically-designed *fusion gate*.
- We conduct extensive experiments to demonstrate the effectiveness of our proposed *KGSynNet* framework while providing detailed case studies and errors analysis. Our model significantly improves the state-of-the-art methods by 14.7% in terms of the offline hits@3 and outperforms the BERT model by 8.3% in the online positive feedback rate.

2 Related Work

Based on how the information is employed, existing methods can be divided into the following three lines:

- The first line of research focuses on capturing the surface morphological features of sub-words in mentions and entities [8, 9, 27]. They usually utilize lexical similarity patterns and the synonym rules to find the synonymous entities of mentions. Although these methods are able to achieve high performance when the given mentions and entities come from the same semantic space, they fail to handle terms with semantic similarity but morphological difference.
- The second line of research tries to learn semantic embeddings of words or sub-words to discover the synonymous entities of mentions [6, 10, 11, 16, 19]. For example, the term-term synonymous relation has been included to train the word embeddings [11]. More heuristic rule-based string features are expanded to learn word embeddings to extract medical synonyms [26]. These methods employ semantic embeddings pretrained from massive text corpora and improve the discovery task in a large margin compared to the direct string matching methods. However, they perform poorly when the terms rarely appear in the corpora but reside in external knowledge bases.
- The third line of research aims to incorporate external knowledge from either the unstructured term-term co-occurrence graph or the structured knowl-

edge graph. For example, Wang et al. [29] utilizes both semantic word embeddings and a term-term co-occurrence graph extracted from unstructured text corpora to discover synonyms on privacy-aware clinical data. Jiang et al. [14] applies the path inference method over knowledge graphs. More powerful methods, such as SynSetMine [23], SA-ESF [13], and the contextualized method [22], have been proposed to leverages the synonym of entities in knowledge graphs or the knowledge representations. They ignore other relations among entities, e.g., the hypernym-hyponym relations, and lack a unified way to absorb the information. This motivates our further exploration in the this work.

3 Methodology

Here, we present the task and the main modules of our *KGSynNet* accordingly.

3.1 Task definition.

The task of entity synonyms discovery is to train a model to map the mention to synonymous entities as accurate as possible given a set of annotated mention-entity pairs $\mathcal{Q}$, a knowledge graph $\mathcal{KG}$, and a domain-specific corpus, $\mathcal{D}$. The mention-entity pairs, $\mathcal{Q} = \{(q_i, t_i)\}_{i=1}^{N}$, record the mentions from queries and their corresponding synonymous entities, where N is the number of annotated pairs, $q_i = q_{i1} \dots q_{i|q_i|}$ denotes the i-th mention with $|q_i|$ subwords and $t_i = t_{i1} \dots t_{i|t_i|} \in \mathcal{E}$ denotes the i-th entity in $\mathcal{KG}$ with $|t_i|$ subwords. The knowledge graph is formalized as $\mathcal{KG} = \{\mathcal{C}, \mathcal{I}, \mathcal{R}, \mathcal{S}\}$, where $\mathcal{C}$ and $\mathcal{I}$ denote the sets of concepts and instances, respectively, $\mathcal{R}$ is the relation set and $\mathcal{S}$ is the triple set. Based on the above definition, we have $\mathcal{E} = \mathcal{C} \cup \mathcal{I}$. After we train the model, for a given mention, we can recommend a list of synonymous entities from the knowledge graph. The domain-specific corpus, $\mathcal{D}$, is used for learning the embeddings of mentions and entities.

As illustrated in Fig. 2, our proposed *KGSynNet* consists of four main modules: (1) a semantic encoder module to represent mentions and entities; (2) a knowledge encoder module to represent the knowledge of entities by a jointly-learned TransC-TransE model; (3) a feature fusion module to adaptively incorporate knowledge information via a specifically designed *fusion gate*; (4) a classifier with a similarity matching metric to train the entire model.

3.2 Semantic Encoder

Given a mention-entity pair, (q, t), we may directly apply existing embeddings, e.g., Word2Vec [18], or BERT [7], on q and t to represent the semantic information of mentions and entities. However, it is not effective because many subwords are out-of-vocabulary (OOV), since the pre-trained embeddings are trained from corpora in general domains.

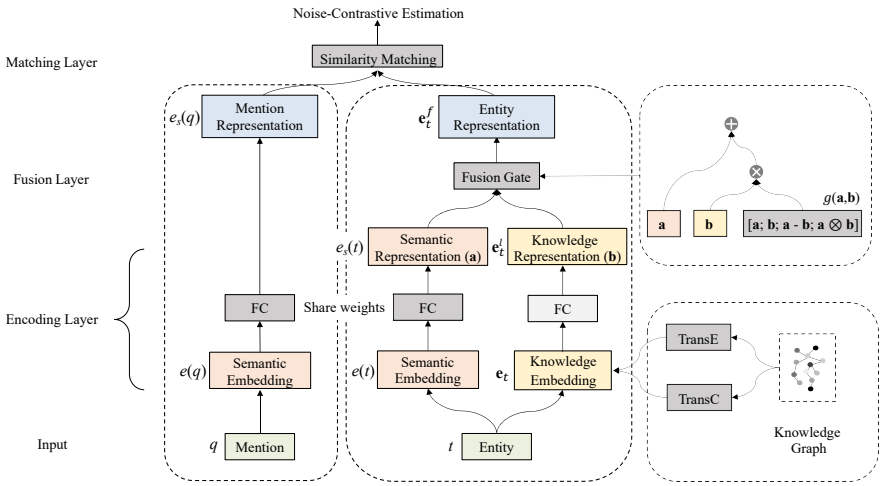

Fig. 2. The architecture of our *KGSynNet*.

To leverage the contextualized information of each mention and entity from $\mathcal{D}$, we train a set of subword-level Word2Vec embeddings from scratch on $\mathcal{D}$, and apply them to initialize the semantic representations of the subwords of the mentions and the entities in $\mathcal{Q}$. Then, similar to the fastText approach [2], we obtain the initialized semantic representations of mentions and entities by averaging their subword representations:

$$e(q) = \frac{1}{|q|}\sum_{k=1}^{|q|} e(q_k), \quad e(t) = \frac{1}{|t|}\sum_{k=1}^{|t|} e(t_k). \tag{1}$$

After that, the semantic embeddings of the mentions and the entities are further fed into a two-layer fully-connected (FC) network to extract deeper semantic features. Here, we adopt shared weights as in [5] to transform the learned embedding $e(v)$ into a semantic space of k-dimension:

$$e_s(v) = \tanh(\mathbf{W}_2 \tanh(\mathbf{W}_1 e(v) + b_1) + b_2) \in \mathbb{R}^k, \tag{2}$$

where v can be a mention or an entity. The parameters, $\mathbf{W}_1 \in \mathbb{R}^{k\times d}$ and $\mathbf{W}_2 \in \mathbb{R}^{k\times k}$, are the weights on the corresponding layers of the FC network. $b_1 \in \mathbb{R}^k$ and $b_2 \in \mathbb{R}^k$ are the biases at the corresponding layers.

3.3 Knowledge Encoder

Though entities can be encoded in the semantic space as detailed above, their representations are not precise enough due to lack of the complementary information included in the knowledge graph.

In the knowledge graph $\mathcal{KG}$, the relation set $\mathcal{R}$ is defined by $\mathcal{R} = \{r_e, r_c\} \cup \mathcal{R}_l \cup \mathcal{R}_{\mathcal{IC}} \cup \mathcal{R}_{\mathcal{CC}}$, where r_e is an instanceOf relation, r_c is a subClassOf relation, $\mathcal{R}_l$ is the instance-instance relation set, $\mathcal{R}_{\mathcal{IC}}$ is the Non-Hyponym-Hypernym (NHH) instance-concept relation set, and $\mathcal{R}_{\mathcal{CC}}$ is the NHH concept-concept relation set. It is noted that different from the three kinds of relations defined in TransC [17], we specifically categorize the relations into five types to differentiate the NHH relations of the instance-concept pairs from the concept-concept pairs. Therefore, the triple set $\mathcal{S}$ can be divided into the following five disjoint subsets:

1. The instanceOf triple set: $\mathcal{S}_e = \left\{\left(i, r_e, c\right)_k\right\}_{k=1}^{|\mathcal{S}_e|}$, where $i \in \mathcal{I}$ is an instance, $c \in \mathcal{C}$ is a concept, and r_e is the instanceOf relation.
2. The subClassOf triple set: $\mathcal{S}_c = \left\{\left(c_i, r_c, c_j\right)_k\right\}_{k=1}^{|\mathcal{S}_c|}$, where $c_i, c_j \in \mathcal{C}$ are concepts, c_i is a sub-concept of c_j, and r_c is the subClassOf relation.
3. The instance-instance triple set: $\mathcal{S}_l = \left\{\left(i, r_{ij}, j\right)_k\right\}_{k=1}^{|\mathcal{S}_l|}$, where $r_{ij} \in \mathbb{R}_l$ defines the instance-instance relation from the head instance i to the tail instance j.
4. The NHH instance-concept triple set: $\mathcal{S}_{\mathcal{IC}} = \left\{\left(i, r_{ic}, c\right)_k\right\}_{k=1}^{|\mathcal{S}_{\mathcal{IC}}|}$, where i and c are defined similarly as $\mathcal{S}_e$. $r_{ic} \in \mathcal{R}_{\mathcal{IC}}$ is an NHH instance-concept relation.
5. The NHH concept-concept triple set: $\mathcal{S}_{\mathcal{CC}} = \left\{\left(c_i, r_{c_i c_j}, c_j\right)_k\right\}_{k=1}^{|\mathcal{S}_{\mathcal{CC}}|}$, where $c_i, c_j \in \mathcal{C}$ denote two concepts, $r_{c_i c_j} \in \mathcal{R}_{\mathcal{CC}}$ is an NHH concept-concept relation.

We now learn the knowledge embeddings of entities. Since TransE [4] is good at modeling general relations between entities while TransC [17] excelling in exploiting the hierarchical relations in the knowledge graph, we propose a unified model, the TransC-TransE model, to facilitate the advantage of both models.

Specifically, TransE represents an entity by $\mathbf{v} \in \mathbb{R}^n$, where n is the size of the knowledge embedding, and defines the loss for the instance-instance triples [4]:

$$f_l(i, r_{ij}, j) = \|\mathbf{v}_i + \mathbf{v}_{r_{ij}} - \mathbf{v}_j\|_2^2, \tag{3}$$

where $(i, r_{ij}, j) \in \mathcal{S}_l$ denotes a triple in the instance-instance relation set, $\mathbf{v}_i$, $\mathbf{v}_{r_{ij}}$, and $\mathbf{v}_j$ denote the corresponding TransE representations.

In TransC, an instance i is represented by a vector, $\mathbf{v}_i \in \mathbb{R}^n$, same as that of an entity in TransE. A concept c is represented by a sphere, denoted by $(\mathbf{p}_c, m_c)$, where $\mathbf{p}_c \in \mathbb{R}^n$ and $m_c \in \mathbb{R}_+$ define the corresponding center and radius for the concept, respectively. The corresponding losses can then be defined as follows:

– The loss for the instanceOf triples [17]:

$$f_e(i, c) = \|\mathbf{v}_i - \mathbf{p}_c\|_2 - m_c, \quad \forall i \in c. \tag{4}$$

– The loss for the subClassOf triples [17]:

$$f_c(c_i, c_j) = \begin{cases} m_{c_i} - m_{c_j}, & c_j \text{ is a subclass of } c_i, \text{ or } c_j \subseteq c_i \\ \|\mathbf{p}_{c_i} - \mathbf{p}_{c_j}\|_2 + m_{c_i} - m_{c_j}, & \text{otherwise} \end{cases}. \tag{5}$$

However, the spherical representation is not precise enough to model the NHH relations. We therefore denote the concept of c by an additional node embedding, $\mathbf{v}_c \in \mathbb{R}^n$, and define the following additional loss functions:

– The loss for the NHH instance-concept triples [4]:

$$f_{\mathcal{IC}}(i, r_{ic}, c) = \|\mathbf{v}_i + \mathbf{v}_{r_{ic}} - \mathbf{v}_c\|_2^2, \tag{6}$$

where the triplet $(i, r_{ic}, c) \in \mathcal{S}_{\mathcal{IC}}$ denotes the NHH instance-concept relation r_{ic} connecting the instance i to the concept c.

– The loss for the NHH concept-concept triples [4]:

$$f_{\mathcal{CC}}(c_i, r_{c_i c_j}, c_j) = \|\mathbf{v}_{c_i} + \mathbf{v}_{r_{c_i c_j}} - \mathbf{v}_{c_j}\|_2^2, \tag{7}$$

where the triplet $(c_i, r_{c_i c_j}, c_j) \in \mathcal{S}_{\mathcal{CC}}$ denotes the NHH concept-concept relation $r_{c_i c_j}$ connecting the concept c_i to the concept c_j.

Therefore, the knowledge embeddings of entities are learned by minimizing the following objective function:

$$\begin{aligned}\mathcal{L}_k = &\sum_{(i,r_e,c)\in\mathcal{S}_e} f_e(i,c) + \sum_{(c_i,r_c,c_j)\in\mathcal{S}_c} f_c(c_i,c_j) + \sum_{(i,r_{ij},j)\in\mathcal{S}_l} f_l(i,r_{ij},j) \\ &+ \sum_{(i,r_{ic},c)\in\mathcal{S}_{\mathcal{IC}}} f_{\mathcal{IC}}(i,r_{ic},c) + \sum_{(c_i,r_{c_ic_j},c_j)\in\mathcal{S}_{\mathcal{CC}}} f_{\mathcal{CC}}(c_i,r_{c_ic_j},c_j).\end{aligned} \tag{8}$$

It is noted that our objective differs from TransC by explicitly including both the NHH instance-concept relations and the NHH concept-concept relations. Similarly, we apply the negative sampling strategy and the margin-based ranking loss to train the model as in [17].

After training the unified TransC-TrainsE model in Eq. (8), we obtain the knowledge embeddings for both instances and concepts, e.g., $\mathbf{v}_i$ for an instance i, and the representation of $(\mathbf{p}_c, m_c)$ and $\mathbf{v}_c$ for a concept c. For simplicity and effectiveness, we average the center and the node embedding of a concept to yield its final knowledge embedding $\mathbf{e}_t$:

$$\mathbf{e}_t = \begin{cases} \mathbf{v}_t & \forall t \in \mathcal{I} \\ (\mathbf{p}_t + \mathbf{v}_t)/2 & \forall t \in \mathcal{C} \end{cases}. \tag{9}$$

Similar to the semantic embeddings, the learned knowledge embeddings of entities obtained in Eq. (9) are transformed into the same k-dimensional semantic space by a two-layer fully connected network to yield $\mathbf{e}_t^l$:

$$\mathbf{e}_t^l = \tanh(\mathbf{W}_4(\tanh(\mathbf{W}_3\mathbf{e}_t + b_3)) + b_4) \in \mathbb{R}^k, \tag{10}$$

where $\mathbf{W}_3 \in \mathbb{R}^{k\times q}$ and $\mathbf{W}_4 \in \mathbb{R}^{k\times k}$ are the weights on the corresponding layers of the FC network. $b_3 \in \mathbb{R}^k$ and $b_4 \in \mathbb{R}^k$ are the biases at the layers.

3.4 Fusion Gate

A critical issue in the task is that the semantic features and the knowledge embeddings are learned separately. To effectively integrate these two types of information, we design a fusion network, named *Fusion Gate*, to adaptively absorb the transformed knowledge information $\mathbf{e}_t^l$ into the semantic information

$e_s(t)$ for an entity t. As illustrated in the upper right grid box of Fig. 2, the final representation of an entity t is computed by

$$\mathbf{e}_t^f = e_s(t) + e_t^l \otimes g(e_s(t), \mathbf{e}_t^l). \tag{11}$$

Here, the implementation is motivated by the highway network [24], but is different on the specific information carrying. Here, we directly feed all the semantic information of the entities to the next level without filtering to guarantee the consistency of the semantic representations between mentions and entities. The interaction of the semantic embeddings and knowledge embeddings of the entities is then fulfilled by the transform gate to determine the amount of knowledge incorporated into the semantic feature, defined by $g(\mathbf{a}, \mathbf{b})$:

$$g(\mathbf{a}, \mathbf{b}) = \text{Softmax}(\mathbf{W}_g[\mathbf{a}; \mathbf{b}; \mathbf{a} - \mathbf{b}; \mathbf{a} \otimes \mathbf{b}]), \tag{12}$$

where $\mathbf{W}_g \in \mathbb{R}^{k \times 4k}$ is the weight of a fully-connected network to reduce the dimension of the concatenated features. The first two features maintain the original form while the latter two measuring the "similarity" or "closeness" of the two features. This allows to compute the high-order interactions between two input vectors [5, 20]. Finally, the Softmax operator is applied to determine the proportion of the flow-in knowledge.

3.5 Similarity Matching and Classification

As the training data only consist of the positive pairs, for each pair (q_i, t_i), we additionally sample some negative pairs $\{(q_i, t_{i_j})\}_{j=1}^{N_i}$, where t_{i_j} is sampled from other mention-to-entity pairs and N_i is the number of sampled negative pairs. Hence, we derive the objective function for the final matching:

$$\mathcal{L}_m = \sum_{i=1}^{N} -\log \left(\frac{\exp\left(e_s(q_i)^T \mathbf{e}_{t_i}^f\right)}{\exp\left(e_s(q_i)^T \mathbf{e}_{t_i}^f\right) + \sum_{j=1}^{N_i} \exp\left(e_s(q_i)^T \mathbf{e}_{t_{i_j}}^f\right)} \right). \tag{13}$$

It is noted that each term in Eq. (13) defines the Noise-Contrastive Estimation (NCE) [12], which is the cross-entropy of classifying the positive pair (q_i, t_i). After training, given a new mention q, we can determine the list of the candidate entities by the rank of $e_s(q_i)^T \mathbf{e}_{t_i}^f$.

4 Experiments

In the following, we present the curated dataset along with the associated knowledge graph, as well as the experimental details.

4.1 Datasets

Knowledge graph. The existing open-source knowledge graphs [1, 3] cannot be used for this task, because they do not provide sufficient disease entities

Table 1. Data statistics.

Knowledge Graph	All	Insurance	Occupation	Medicine	Cross Domain
# Entities	75,153	1,409	2,587	71,157	0
# Entity_type	17	2	2	13	0
# Relations	1,120,792	2,827	2,580	1,098,280	17,105
# Relation_type	20	2	2	13	4
# Mention-entity pairs in Train/Dev/Test					45,500/5,896/5,743
# Regular cases/# Difficult cases					5,303/440

and relations required by the task. Therefore, we construct a specific knowledge graph ($\mathcal{KG}$) to verify this task. Table 1 records the statistics of the constructed $\mathcal{KG}$, a heterogeneous $\mathcal{KG}$ with entities collected from three categories: *Insurance Products*, *Occupation*, and *Medicine*. In *Insurance Products*, there are 1,393 insurance products and 16 concepts; while in *Occupation*, there are 1,863 instances and 724 concepts obtained from the nation's professional standards [3]. Both *Insurance Products* and *Occupation* contain only two types of relations, i.e., the instanceOf relation and the subClassOf relation. In *Medicine*, 45K disease entities and 9,124 medical concepts are extracted from three different resources: (1) raw text of insurance products' clauses; (2) users' query logs in the app; (3) the diagnostic codes of International Classification of Diseases (ICD-10). Furthermore, 18K other types of medical entities, such as symptom, body part, therapy, and treatment material, are extracted from some open-source medical knowledge graphs [4]. The relation types include not only instanceOf and subClassOf, but also the instance-instance relations, the NHH concept-instance relations, and the NHH concept-concept relations, 13 types in total.

Data. We collect a large-scale Chinese medical corpus from 14 medical textbooks [5], 3 frequently used online medical QA forums, and some QA forums [6]. We also deploy a self-developed BERT-based NER tool to extract 100K disease mentions from users' query logs in the professional app. From the extracted disease mentions and $\mathcal{KG}$ entities, we generate 300K candidate synonymous mention-entity pairs based on the similarity score computed by BERT. The extracted mention-entity candidates are double-blindly labeled to obtain 57,139 high-quality disease mention-entity synonym pairs. After that, the dataset is randomly split into the sets of training, development, and test, respectively, approximately at a ratio of 8:1:1. We further divide the test set (the All case group) into two groups based on the surface form similarity. That is, a Regular case means that there is at least one identical subword between the mention and the entity, while the rest pairs belong to the Difficult case group.

[3] http://www.jiangmen.gov.cn/attachment/0/131/131007/2015732.pdf
[4] http://openkg.cn/dataset/symptom-in-chinese; http://openkg.cn/dataset/omaha-data.
[5] https://github.com/scienceasdf/medical-books
[6] https://github.com/lrs1353281004/Chinese_medical_NLP

4.2 Compared Methods

We compare *KGSynNet* with the following strong baselines:
(1) JACCARD [21]: a frequently used similarity method based on the surface matching of mentions and entities;
(2) Word2Vec [6]: a new subword embedding is trained on the medical corpus to learn representations. Cosine similarity is then applied to the average of subword embeddings of each mention-entity pair to rank their closeness;
(3) CNN [19]: a CNN-based Siamese network is trained using the triplet loss with the newly trained word2vec embeddings for the mentions and entities.
(4) BERT [7]: the [CLS] representations of mentions and entities are extracted from the fine-tuned BERT to compute their cosine similarity;
(5) DNorm [15]: one of the most popular methods that utilizes the TF-IDF embedding and a matching matrix, trained by the margin ranking loss, to determine the similarity score between mentions and entities.
(6) SurfCon [29]: one of the most popular methods that constructs a term-term co-occurrence graph from the raw corpus to capture both the surface information and the global context information for entity synonym discovery.

4.3 Experimental Setup and Evaluation Metrics

The number of sampled negative mention-entity pairs is tuned from {10, 50, 100, 200, 300} and set to 200 as it attains the best performance in the development set. ADAM is adopted as the optimizer with an initial learning rate of 0.001. The training batch size is 32, and the dimension of the knowledge graph embedding is 200. Besides, the dimension of the semantic embeddings of both mentions and entities are set to 500, and the dimensions of the first and the second FC networks are set to 300. These parameters are set by a general value and tuned in a reasonable range. Dropout is applied in the FC networks and selected as 0.5 from {0.3, 0.5, 0.7}. The knowledge embedding is trained by an open-source package [7]. Early stopping is implemented when the performance in the development set does not improve in the last 10 epochs.

To provide fair comparisons, we set the same batch size, embedding sizes, and dropout ratio to all baseline models. For SurfCon, we construct a co-occurrence graph of 24,315 nodes from our collected Chinese medical corpus, and obtain the graph embedding according to [29] [8].

Filtered hits@k, the proportion of correct entities ranked in the top k predictions by filtering out the synonymous entities to the given mention in our constructed $\mathcal{KG}$, because it is an effective metric to determine the accuracy of entity synonyms discovery [17]. We follow the standard evaluation procedure [4, 17] and set $k = 3, 5, 10$ to report the model performance.

[7] https://github.com/davidlvxin/TransC
[8] https://github.com/yzabc007/SurfCon

4.4 Experimental Results

Rows three to nine of Table 2 report the experimental results of the baselines and our *KGSynNet*. It clearly shows that

- JACCARD yields no hit on the difficult case because it cannot build connections on mentions and entities when they do not contain a common sub-word.
- Word2Vec yields the worst performance on the *All* case and the *Regular* case since the representations of mentions and entities are simply obtained by their mean subword embeddings, which blur the effect of each subword.
- CNN improves Word2Vec significantly because of the Siamese network, but cannot even beat JACCARD due to the poor semantic representation learned from Word2Vec.
- BERT gains further improvement over JACCARD, Word2Vec, and CNN by utilizing the pre-trained embeddings. The improvement is not significant enough especially in the Difficult case because the representation of the token [CLS] does not fully capture the relations between mentions and entities.
- DNorm further improves the performance by directly modeling the interaction between mentions and entities. SurfCon yields the best performance among all baselines because it utilizes external knowledge bases via the term-term co-occurrence graph.
- Our *KGSynNet* beats all baselines in all three cases. Especially, we beat the best baseline, SurfCon, by 14.7%, 10.3%, and 5.6% for the *All* case, 14.2%, 10.0%, and 5.4% for the *Regular* case, and 45.7%, 24.4%, and 10.2% for the *Difficult* case with respect to Hits@3, Hits@5, and Hits@10, respectively. We have also conducted the statistical significance tests, and observe that for the *All* case group, $p << 0.01$ under the paired t-tests. The significant improvement clearly shows that our *KGSynNet* is effective in integrating the knowledge information with the semantic features.

Table 2. Experimental results: − means that *KGSynNet* removes the component while → means that *KGSynNet* replaces the fusion method.

Methods	hits@3			hits@5			hits@10		
	All	Regular	Difficult	All	Regular	Difficult	All	Regular	Difficult
JACCARD [21]	52.28%	56.61%	0.00%	58.03%	62.83%	0.00%	63.76%	69.04%	0.00%
Word2Vec [6]	47.00%	50.88%	0.00%	52.28%	56.59%	2.30%	58.31%	63.10%	4.60%
CNN [19]	51.76%	55.69%	4.33%	57.75%	61.98%	6.38%	65.13%	69.72%	9.34%
BERT [7]	54.60%	58.87%	2.96%	60.41%	65.02%	4.78%	66.50%	71.39%	7.52%
DNorm [15]	56.23%	59.78%	12.76%	63.79%	67.58%	17.77%	71.89%	75.64%	26.42%
SurfCon [29]	58.29%	62.02%	12.98%	66.27%	70.11%	19.59%	75.20%	79.03%	28.93%
KGSynNet	**66.84%**	**70.81%**	18.91%	**73.09%**	**77.13%**	24.37%	**79.41%**	**83.35%**	31.89%
−KE	64.91%	69.07%	14.58%	71.56%	75.77%	20.73%	79.12%	83.14%	30.52%
−TransC	65.80%	69.92%	15.95%	71.44%	75.79%	18.91%	78.94%	83.18%	27.80%
→DA	63.51%	67.19%	**19.13%**	70.85%	74.47%	**27.10%**	78.13%	81.77%	**34.17%**
→EF	61.98%	65.85%	15.26%	68.63%	72.54%	21.41%	76.28%	80.29%	27.79%

4.5 Ablation Study

To better understand why our *KGSynNet* works well, we compare it with four variants: (1) −KE: removing the knowledge embedding and the Fusion Gate; (2) −TransC: removing losses of Eq. (4) and Eq.(5) from Eq. (8) of TransC, to learn the knowledge embedding by utilizing only TransE; (3) →DA: directly adding the learned semantic features and knowledge features of entities together; and (4) →EF: fusing the learned semantic features and knowledge information via a FC network [30].

Table 2 reports the results of the variants in the last four rows and clearly shows three main findings:

– By excluding the knowledge embedding (see the last fourth row in Table 2), our *KGSynNet* drops significantly for the All case, i.e., 1.93 for hits@3, 1.53 for hits@3, and 0.29 for hits@10, respectively. Similar trends appear for the Regular case and the Difficult case. The performance decay is more serious than those in other variants, −TransC and →DA. This implies the effectiveness of our *KGSynNet* in utilizing the knowledge information.
– By removing TransC, we can see that the performance decays accordingly in all cases. The results make sense because learning the knowledge representation by TransE alone does not specifically model the *InstanceOf* relation and the *SubclassOf* relation. This again demonstrates the effectiveness of our proposed TransC-TransE framework.
– In terms of the fusion mechanism, the performance exhibits similarly under the three metrics. Here, we only detail the results of hits@3. It shows that the performance by *Fusion Gate* beats "DA" and "EF" 3.3 to 5.0 in both the All and Regular cases. However, "DA" improves the performance significantly on the Difficult case, i.e., no common sub-word appearing in the mention-entity pairs. The results make sense because in the Difficult case, the model depends heavily on the external knowledge. Setting the weight to 1, i.e., the largest weight, on the learned knowledge features can gain more knowledge information. On the contrary, "EF" yields the worst performance on the All and Regular cases, but gains slightly better performance than −KE on the Difficult case. We conjecture one reason is that the available data is not sufficient to trained a more complicated network in "EF".

4.6 Online Evaluation

Our *KGSynNet* has been deployed in the entity linking module, a key module of the KBQA system of a professional insurance service app, served more than one million insurance agents. The architecture of the online system is shown in Fig. 3. On average, the requests of the KBQA service of the app are 700K per day with more than 50 requests per second at the peak.

We conducted an A/B test to compare the original BERT model and our *KGSynNet* on the entity linking module of the KBQA system for two weeks. The traffic was evenly split into two groups. Approximately 10% of users' queries

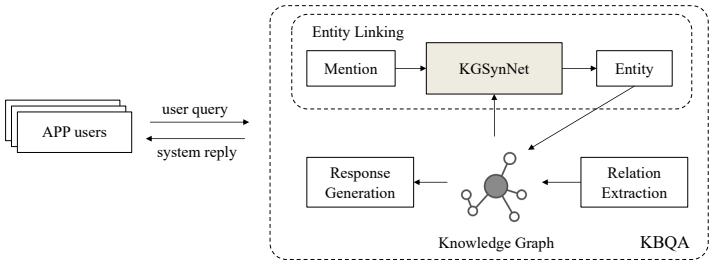

Fig. 3. The architecture of online system.

involve disease mentions, within which the proportion of queries with user experience feedback is around 5%. Eventually, BERT and *KGSynNet* received about 25K and 26K user feedbacks, respectively. The positive rate of the feedback for BERT is about 34.9%, while the positive rate of *KGSynNet* is about 37.8%, significantly better with $p < 0.05$ under the paired t-test.

Moreover, we randomly selected and labeled 1000 disease related queries from each of the two groups. The proportion of queries involving difficult cases was around 3% in both groups. Results in Table 3 show that *KGSynNet* consistently outperforms BERT in terms of hits@3, hits@5, and hits@10, respectively.

Table 3. Online evaluation results

Metric	BERT			*KGSynNet*		
	All	Regular	Difficult	All	Regular	Difficult
hits@3	58.2%	59.9%	3.3%	**68.4%**	70.0%	18.8%
hits@5	63.2%	64.9%	6.7%	**75.4%**	77.1%	25.0%
hits@10	70.0%	71.9%	10.0%	**81.7%**	83.4%	31.3%

4.7 Case Studies

We provide several typical examples to show the effectiveness of our *KGSynNet*. In Table 4, four query mentions are selected with the top-5 discovered synonymous entities. The results show that:

- Our *KGSynNet* can successfully detect at least one annotated synonym for each mention. For example, for the mention, "hyperelastic skin", our found top-5 synonymous entities are all correct.
- For the mention of "facial paralysis", other than its synonym "facioplegia", our *KGSynNet* can discover "prosopoplegia" through the semantic equivalence. Other top predicted terms, e.g., "neonatal facial paralysis", "peripheral facial paralysis", and "idiopathic facial paralysis", are all hyponyms of the mention with specific clinical manifestations.

Table 4. Query mentions and the corresponding top 5 synonymous entities: the correct synonyms are underlined.

Mention	Top 5 Found Entities
弹力过度性皮肤 hyperelastic skin	埃莱尔-当洛综合症,埃勒斯-当洛斯综合症,皮肤松垂,埃莱尔-当洛,皮肤松弛, Ehlers-Danlos syndrome, Ehlers-Danlos syndrome, dermatolysis, Ehlers-Danlos, cutis laxa
肚子痛 stomachache	急性腹泻, 疼痛, 下腹痛, 全身疼痛, 疼痛性脂肪过多症 collywobbles, pain, hypogastralgia, generalized pain, lipomatosis dolorosa
歪嘴风 facial paralysis	面瘫, 面神经麻痹, 新生儿面部神经麻痹, 周围性面瘫, prosopoplegia, facioplegia, neonatal facial paralysis, peripheral facial paralysis, 特发性面神经瘫痪 idiopathic facial paralysis
倦怠 exhaustion	虚弱,乏力, 张力失常, 失眠症, 弱精 debility, asthenia, dystonia, insomnia, asthenozoospermia

4.8 Error Analysis

We provide a concrete error analysis by sampling 10% of the incorrectly predicted mention-entity pairs in our *KGSynNet*. Table 5 lists the main error types:

- More than half of the errors (54%) occur due to the lack of knowledge in the knowledge graph. For example, since the entity "bow legs" is not in the $\mathcal{KG}$, the mention "knee varus" mistakenly found "knee valgus" and "congenital knee valgus" through surface matching.
- The second largest error comes from hypernyms distraction, which accounts for 29% of the total errors. For example, the mention "pituitary gland cancer" is distracted to its hypernym "brain cancer" and "cerebral cancer", and failed to identify the true entity "pituitary gland malignant tumor".
- Another 12% of the errors are due to the keyword extraction error. For example, the golden entity for the mention, "lung calcification", is "lung mineralization". Our *KGSynNet* makes a wrong extraction on the keyword "calcification" and discovers a wrong entity, "bronchial calcification", for this mention. It seems that this problem may be alleviated by adding an fine-grained feature interaction between mentions and entities in our *KGSynNet*.

5 Conclusion

In this paper, we tackle the task of entity synonyms discovery and propose *KGSynNet* to exploit external knowledge graph and domain-specific corpus. We resolve the OOV issue and semantic discrepancy in mention-entity pairs. Moreover, a jointly learned TransC-TransE model is proposed to effectively represent knowledge information while the knowledge information is adaptively absorbed into the semantic features through *fusion gate* mechanism. Extensive experiments and detailed analysis conducted on the dataset show that our model significantly improves the state-of-the-art methods by 14.7% in terms of the offline hits@3 and outperforms the BERT model by 8.3% in the online positive feedback rate.

Table 5. Error analysis. The "Golden Entity" is the correct entity for the corresponding mention.

Error Type	Proportion	Mention	Golden Entity	Top 2 Predicted Entities
Lack of Knowledge	54%	膝内翻 knee varus	O型腿 bow legs	膝外翻, knee valgus, 先天性膝外翻 congenital knee valgus
Hypernym Distraction	29%	脑垂腺癌 pituitary gland cancer	垂体恶性肿瘤 pituitary gland malignant tumor	脑癌, brain cancer, 癌性脑病 cerebral cancer
Keyword Extraction Error	12%	肺部钙化 lung calcification	肺矿化 lung mineralization	支气管钙化, bronchial calcification, 肺转移瘤 pulmonary metastasis
Others	5%	奥尔布赖特综合症 albright's syndrome	mccune-albright综合征 mccune-albright's syndrome	莱特尔综合症, leiter's syndrome, 吉尔伯特综合症 gilbert's syndrome

Regarding future work, we can extend our *KGSynNet* to other domains, e.g., education or justice, to verify its generalization ability.

Acknowledgement

The authors of this paper were supported by NSFC under grants 62002007 and U20B2053. For any correspondence, please refer to Haiqin Yang and Hao Peng.

References

1. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia - A crystallization point for the web of data. J. Web Semant. **7**(3), 154–165 (2009)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguistics **5**, 135–146 (2017)
3. Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD. pp. 1247–1250. ACM (2008)
4. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS. pp. 2787–2795 (2013)
5. Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., Inkpen, D.: Enhanced LSTM for natural language inference. In: ACL. pp. 1657–1668 (2017)
6. Cho, H., Choi, W., Lee, H.: A method for named entity normalization in biomedical articles: application to diseases and plants. BMC Bioinform. **18**(1), 451:1–12 (2017)
7. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL. pp. 4171–4186 (2019)
8. Dogan, R.I., Lu, Z.: An inference method for disease name normalization. In: AAAI (2012)
9. D'Souza, J., Ng, V.: Sieve-based entity linking for the biomedical domain. In: ACL and IJCNLP. pp. 297–302 (2015)

10. Faruqui, M., Dodge, J., Jauhar, S.K., Dyer, C., Hovy, E.H., Smith, N.A.: Retrofitting word vectors to semantic lexicons. In: NAACL. pp. 1606–1615 (2015)
11. Fei, H., Tan, S., Li, P.: Hierarchical multi-task word embedding learning for synonym prediction. In: ACM SIGKDD. pp. 834–842 (2019)
12. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: AISTATS. vol. 9, pp. 297–304 (2010)
13. Hu, S., Tan, Z., Zeng, W., Ge, B., Xiao, W.: Entity linking via symmetrical attention-based neural network and entity structural features. Symmetry **11**(4), 453 (2019)
14. Jiang, L., Luo, P., Wang, J., Xiong, Y., Lin, B., Wang, M., An, N.: GRIAS: an entity-relation graph based framework for discovering entity aliases. In: IEEE ICDM. pp. 310–319 (2013)
15. Leaman, R., Dogan, R.I., Lu, Z.: Dnorm: disease name normalization with pairwise learning to rank. Bioinform. **29**(22), 2909–2917 (2013)
16. Li, H., Chen, Q., Tang, B., Wang, X., Xu, H., Wang, B., Huang, D.: Cnn-based ranking for biomedical entity normalization. BMC Bioinform. **18**(S-11), 79–86 (2017)
17. Lv, X., Hou, L., Li, J., Liu, Z.: Differentiating concepts and instances for knowledge graph embedding. In: EMNLP. pp. 1971–1979 (2018)
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS. pp. 3111–3119 (2013)
19. Mondal, I., Purkayastha, S., Sarkar, S., Goyal, P., Pillai, J., Bhattacharyya, A., Gattu, M.: Medical entity linking using triplet network. In: Clinical NLP (2019)
20. Mou, L., Men, R., Li, G., Xu, Y., Zhang, L., Yan, R., Jin, Z.: Natural language inference by tree-based convolution and heuristic matching. In: ACL (2016)
21. Niwattanakul, S., Singthongchai, J., Naenudorn, E., Wanapu, S.: Using of jaccard coefficient for keywords similarity. In: IMECS (2013)
22. Schumacher, E., Dredze, M.: Learning unsupervised contextual representations for medical synonym discovery. JAMIA Open (2019)
23. Shen, J., Lyu, R., Ren, X., Vanni, M., Sadler, B.M., Han, J.: Mining entity synonyms with efficient neural set generation. In: AAAI. pp. 249–256 (2019)
24. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks. In: NIPS. pp. 2377–2385 (2015)
25. Sung, M., Jeon, H., Lee, J., Kang, J.: Biomedical entity representations with synonym marginalization. In: ACL. pp. 3641–3650 (2020)
26. Wang, C., Cao, L., Zhou, B.: Medical synonym extraction with concept space models. In: IJCAI. pp. 989–995 (2015)
27. Wang, J., Lin, C., Li, M., Zaniolo, C.: An efficient sliding window approach for approximate entity extraction with synonyms. In: EDBT. pp. 109–120 (2019)
28. Wang, X., Kapanipathi, P., Musa, R., Yu, M., Talamadupula, K., Abdelaziz, I., Chang, M., Fokoue, A., Makni, B., Mattei, N., Witbrock, M.: Improving natural language inference using external knowledge in the science questions domain. In: AAAI. pp. 7208–7215 (2019)
29. Wang, Z., Yue, X., Moosavinasab, S., Huang, Y., Lin, S.M., Sun, H.: Surfcon: Synonym discovery on privacy-aware clinical data. In: ACM SIGKDD. pp. 1578–1586 (2019)
30. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: ERNIE: enhanced language representation with informative entities. In: ACL. pp. 1441–1451 (2019)