A Knowledge-driven Generative Model for Multi-implication Chinese Medical Procedure Entity Normalization

Jinghui Yan[1]*, Yining Wang[2,3], Lu Xiang[2,3], Yu Zhou[2,3,5], Chengqing Zong[2,3,4]
[1]School of Computer Science and Information Technology & Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing, China
[2]National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
[3]University of Chinese Academy of Sciences, Beijing, China
[4]CAS Center for Excellence in Brain Science and Intelligence Technology, Beijing, China
[5] Beijing Fanyu Technology Co., Ltd
jh_yan@bjtu.edu.cn
{yining.wang, lu.xiang, yzhou, cqzong}@nlpr.ia.ac.cn

Abstract

Medical entity normalization, which links medical mentions in the text to entities in knowledge bases, is an important research topic in medical natural language processing. In this paper, we focus on Chinese medical procedure entity normalization. However, non-standard Chinese expressions and combined procedures present challenges in our problem. The existing strategies relying on the discriminative model are poorly to cope with normalizing combined procedure mentions. We propose a sequence generative framework to directly generate all the corresponding medical procedure entities. we adopt two strategies: category-based constraint decoding and category-based model refining to avoid unrealistic results. The method is capable of linking entities when a mention contains multiple procedure concepts and our comprehensive experiments demonstrate that the proposed model can achieve remarkable improvements over existing baselines, particularly significant in the case of multi-implication Chinese medical procedures.

1 Introduction

Named entity normalization (NEN), which is also known as entity linking, is one of the fundamental tasks within natural language processing (Hachey *et al.*, 2013; D'Souza, 2015; Fang *et al.*, 2016; Wu *et al.*, 2018). Medical entity normalization is a typical problem of NEN in the medical domain, which aims at linking references or mentions of medical terminology to standard entities in a given medical knowledge base (KB) such as the International Statistical Classification of Diseases and Related Health Problems 9th Revision (ICD-9).

Due to the nature of the domain, although different occupational or writing habits can result in standard entities having different literal expressions, the linked standard entities of a given medical mention should always be unique. Thus, unlike NEN as applied generally(Hachey *et al.*, 2013; Luo *et al.*, 2015; Wu *et al.*, 2018; Aguilar *et al.*, 2019), in the medical domain, the main challenge is not ambiguity – the same entity mention may be linked to different concepts, it is variation – the same underlying concept can be linked by different entity mentions. However, different from the normalization task of the medical entity with simple nominal structure, such as disease (Kang *et al.*, 2012; D'Souza and Ng, 2015) or anatomical body (Wang *et al.*, 2019), Chinese medical procedure normalization have to face the challenge of multi-implication – a mention which contains multiple procedure concepts should link to multiple standard procedure entities in KB. To clarify, these linked entities are instances of concepts in KB and have no parent-child relationships on each other.

Figure 1 shows some examples of Chinese medical combined procedure entity normalization. In case 1, the mention left of the dotted line implicates two procedure concepts and links to two different standard entities, where the word "颅神经 (cranial nerve)" is omitted from the mention. Besides, features in the textual level are not detailed enough to identify the exact number of procedures in given mentions such as case 2, which is a Tri-combined procedure but with only one "+" delimiter. Hence how to identify the number of linking entities for a given mention is crucial for Chinese medical procedure normalizing.

Previous studies which adopt the discriminative model to solve the problem of variation in medical entity normalization (Li *et al.*, 2017; Luo *et al.*, 2018; Ji *et al.*, 2019; Deng *et al.*, 2019) involve two basic steps: First, entity candidates are selected from all entities in KB through artificially designed rules (Li *et al.*, 2017) or text similarity methods,

* Contribution during internship at National Laboratory of Pattern Recognition, Institute of Automation, CAS

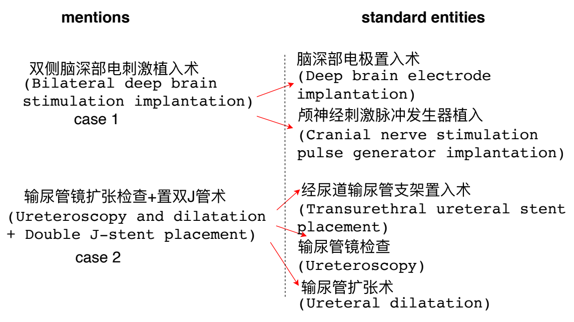

mentions
standard entities
双侧脑深部电刺激植入术
(Bilateral deep brain stimulation implantation)
case 1
脑深部电极置入术
(Deep brain electrode implantation)
颅神经刺激脉冲发生器植入
(Cranial nerve stimulation pulse generator implantation)
输尿管镜扩张检查+置双J管术
(Ureteroscopy and dilatation + Double J-stent placement)
case 2
经尿道输尿管支架置入术
(Transurethral ureteral stent placement)
输尿管镜检查
(Ureteroscopy)
输尿管扩张术
(Ureteral dilatation)

Figure 1: Example of multi-implication Chinese medical procedure entity normalization

such as BM25 (Ji *et al.*, 2019); Then, a discriminative model is used to measure semantic similarity score between the original mention and selected candidates and get the highest one as the normalization result. We refer to these approaches simply as "selecting and re-ranking (SR)." SR strategies based on neural networks have proven to be efficient when dealing with "uni-implication" entity normalization problems, in which a mention links to only one standard entity in a given KB. However, those works fail to pay attention to the multi-implication problem. The most significant weakness of SR is that it cannot identify the number of linking entities but default to the "uni-implication" problem for each given mention.

To tackle the "multi-implication" challenge, we propose a sequence generative framework to directly generate all the corresponding standard entities. However, since a generative model makes unrealistic independence assumptions about the joint distribution of features and classes (Toutanova, 2006), two methods are introduced to constrain the model output, as follows. **1) Constraint decoding**. Normally, entities in KB is grouped together in categories and each entity is assigned to a corresponding category label (e.g. label 8 refers to Operations on eyelids in ICD-9-CM Vol. 3 Procedure Codes). For this reason, we give each entity in the KB a unique category label and derive a label prefix tree for each category to accommodate all the entities belonging to it. Then, the generative model will, in turn, decode the category label and standard entities when given the input mention. At each decoding step, we construct a constraint character set by tracing previously generated characters with corresponding label prefix trees. Finally, we integrate the constraint set into our model to restrict the generated characters belonging to its corresponding category. **2) Catebory-based refining**. Entities under the same label always share common information. Inspired by Li *et al.* (2018), we propose a category-based refinement strategy in order to make the model parameters better fit the category of input mentions. To achieve this, for each input mention, we first adopt the general model to normalize it and obtain the category information from the output. Then we redistribute the original test dataset into several sub-test datasets based on category. Finally, for each sub-test dataset, we find sentence pairs within the same categories from the training data and use them to fine-tune the parameters of the general model. In addition, we propose a "generating and re-ranking" strategy. For each mention, several standard entity candidates are produced by a generative model via a beam search method instead of selecting candidates from the given KB. Then, a pre-trained discriminative model is used to score and re-rank all the candidates.

Overall, we make the following contributions in this paper.

- We propose a sequence generative framework to handle the "multi-implication" problem in Chinese medical procedure entity normalization tasks. To the best of our knowledge, the "multi-implication" problem has not been addressed in any previous research.

- We design novel approaches to constrain the generative model to capture category labels and word-formation from a KB while avoiding unrealistic results.

- Our detailed experimental analysis on Chinese medical procedure entity normalization tasks realizes remarkable improvements over existing methods.

2 Proposed method

The main idea underlying this work is based on introducing a sequence generative framework to directly generate all linked standard entities at once, which can then normalize both "uni-implication" and "multi-implication" types of mentions in an end-to-end fashion. In this section, we introduce the proposed category-based constraint decoding and model-refining methods, which can avoid generating unrealistic results and adapt the test sets belonging to particular categories. As the basis of our work, we first introduce two definitions used in

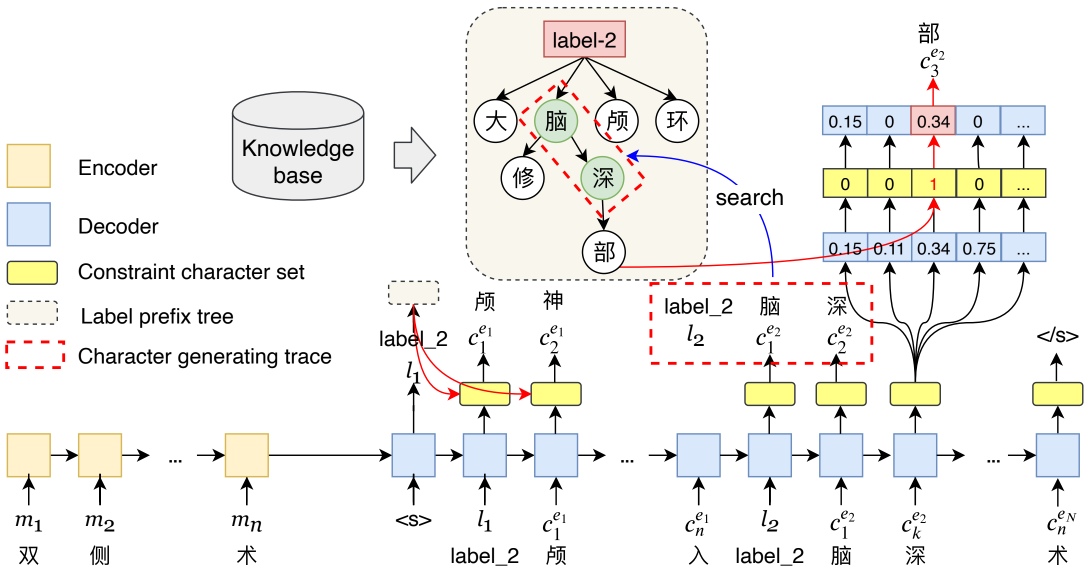


Figure 2: The overall structure of category-based constraint decoding

the proposed methods.

Definition 1 (character generating trace): The character generating trace is the order of characters in standard entities.

Definition 2 (previous trace): The previous trace of a character c_i consists of two parts: a categary label l and a generated character sequence $s = c_1, c_2, ..., c_{i-1}$ under l.

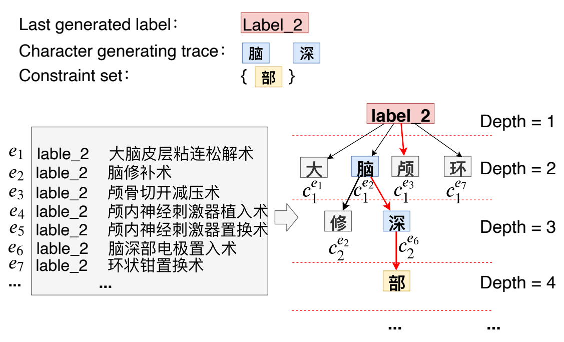
Considering the excellent performance of an existing self-attention-based transformer (Vaswani *et al.*, 2017), we implement our method based on this architecture. The input and output of encoder and decoder are all character-based sequences.

2.1 Category-based constraint decoding

Given an input mention M, the decoder is used to generate all corresponding entities $\{e_1, e_2, ..., e_N\}$ in one output sequence. To apply the category constraint to make the results more reliable, the model decoder should ensure that the character generated at each time step can follow the **character-generating trace** of the previous output. The proposed model decodes both entities and their corresponding category labels. First, the decoder generates a category label for an entity. Next, the decoder reads all entities under the generated label in a KB and constructs a prefix tree. Finally, the decoder generates the entity characters. At each decoding step a character constraint set will be looked up from the prefix tree based on the label and previously generated characters, and be used to constrain the present time-step output. This process is repeated until a sequence stop symbol is generated by the decoder, to indicate that all entities have been generated. Whenever a new category label is generated, a new prefix tree is created from the KB and replaces the old one.

Figure 2 presents an example to illustrate the proposed decoding method. To generate the third character of the second entity c_3^{e2}, we first obtain the character constraint set from a prefix tree. The prefix tree is created based on the latest generated label l_2. Then, the decoder traces the prefix tree in the order of $l_2 c_1^{e2} c_2^{e2}$ to obtain the character constraint set for c_3^{e2}. Finally, the constraint set is transformed into a mask matrix, with the conditional probability output by the model. The details of the label prefix tree and constraint character set are introduced in the following subsections.

2.1.1 Label prefix tree



Last generated label: Label_2
Character generating trace: 脑 深
Constraint set: { 部 }
e_1 lable_2 大脑皮层粘连松解术
e_2 lable_2 脑修补术
e_3 lable_2 颅骨切开减压术
e_4 lable_2 颅内神经刺激器植入术
e_5 lable_2 颅内神经刺激器置换术
e_6 lable_2 脑深部电极置入术
e_7 lable_2 环状钳置换术
label_2
Depth = 1
Depth = 2
Depth = 3
Depth = 4
大 脑 颅 环
修 深
部

Figure 3: Example of tracing a prefix tree

To improve the efficiency of searching a character-generating trace, we derive each category in the KB using a prefix tree to accommodate

Algorithm 1: Construction of prefix tree

```
Input: Category label $l$; Knowledge base KB
Output: prefix tree of $l$
1  Extracting all entities under the category $l$ in KB $\rightarrow S_l$ ;
2  initialize a tree $T$ ;
3  rootNode($l$)$\rightarrow T$ ;
4  for $i$ in $1, 2, ...,$ len($S_l$) do
5      selectEntity($i$) $\rightarrow e_i$ ;
6      root $\rightarrow$ parentNode ;
7      for $m$ in $1, 2, ...,$ len($e_i$) do
8          selectChar($m$)$\rightarrow c_m^{e_i}$ ;
9          initNode($c_m^{e_i}$, $m$,parentNode)$\rightarrow d_m^i$ ;
10         getChildList(parentNode) $\rightarrow cList$ ;
11         if $d_m^i$ not in $cList$ then
12             addNode($d_m^i$)$\rightarrow T$ ;
13         end
14         $d_m^i$ $\rightarrow$ parentNode ;
15     end
16 end
```

the character-generating trace of all the candidates. Here, we use the example in Figure 3 to illustrate the construction and tracing of a prefix tree. We assume that the previously generated characters are "脑，深", which are the first and second decoder steps after the latest generated category label "label_2." The procedure for constructing prefix tree T is illustrated in **Algorithm 1**. The first step is to search the KB and collect n entities under the category "label_2" into a set $S_{label\_38} = e_1, e_2, ..., e_n$ (line 1). The formation of entities in the KB is the same as in the textbox (bottom left). The root of this prefix tree is the category label and its depth is the maximum length of the entity in S (line 2-3). For each entity e_i in S_{label_38}, we traverse its character $c_m^{e_i}$, and add it to the tree if its parent node $c_{m-1}^{e_i}$ has no node that is the same as it, where m indicates the position within e_i (lines 4-16).

With the prefix tree in place, we can now obtain the character constraint set C by following the previous trace of "(label_2, [脑，深])". **Algorithm 2** shows the procedure. If the generated character sequence s is null, the returned constraint set will be the list of child nodes from the root node (lines 1, 2); otherwise, the prefix tree will be traced until the last character in s and the returned constraint set $\{$部$\}$ is a child node list of the last character "深" (lines 5-13). Note that if the constraint set is empty, which means the last character in s is a leaf

Algorithm 2: Obtaining character constraint set

```
Input: prefix tree $T$; previous trace tuple($label, s$); label set $L$; ENDSYMBOL
Output: character constraint set $C$
1  if len($s$) = 0 then
2      $C$ = getChildList(root) ;
3  end
4  else
5      root $\rightarrow$ parentNode ;
6      for $i$ in $1, 2, ...,$len($s$) do
7          getChildList(parentNode) $\rightarrow cList$ ;
           for $d$ in $cList$ do
8              if getValue($d$) = $s_i$ then
9                  $d$ $\rightarrow$ parentNode ;
10             end
11         end
12     end
13     getChildList(parentNode) $\rightarrow C$ ;
14 end
15 if $C = \emptyset$ then
16     $L \cup$ ENDSYMBOL $\rightarrow C$ ;
17 end
```

node, it will be changed to the set of all category labels joined with a sequence-ending symbol.

2.1.2 Constraint character set

The constraint character set C_t is used to restrict the original conditional probability $p(y_t|y_{<t})$ of the generative model output:

$$\hat{p}(y_t|y_{<t}) = \lambda p(y_t|y_{<t})$$
$$\lambda = \begin{cases} 0 & y_t \in C_t \\ 1 & y_t \notin C_t \end{cases} \tag{1}$$

where y_t is the character generated by the decoder at time step t, and λ is a constraint operator.

2.2 Category-based model refining

The parameters of the general fixed model cannot best-fit each test item (Li *et al.*, 2018). Thus, we propose to refine the general model to obtain a category-specific model for each test data. As illustrated in Figure 4, the proposed model-refining strategy comprises the following steps.

(1) Learn a general model M_G (Section 2.1) based on all of the training data $D = (s_1, t_1), (s_2, t_2), ..., (s_n, t_n)$.

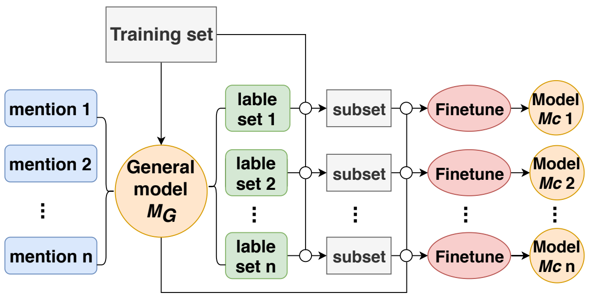


Figure 4: Procedure of category-based model refining

(2) Use the general model M_G to generate the corresponding results $\hat{t}$ for each given mention $\hat{s}$ in the test set and record the generated category label set $\hat{l}$ (Algorithm 3, line 1-2).

(3) For each $\hat{s}$, we extract a training subset $\hat{D}$ from the training data, consisting of mention-entity pairs whose category label set is a subset of $\hat{l}$ (Algorithm 3, line 3-9).

(4) Fine-tune the general model with each new training subset $\hat{D}$ to obtain a category-specific model M_C (Algorithm 3, line 10).

This procedure can be formulated as two-stage optimization. The first stage is to to find a set of network parameters θ to maximize the log-likelihood of the whole training data D.

$$\hat{\theta} = \arg\max_{\theta}\{log \prod_{i=1}^{n} P(t_i|s_i;\theta)\} \quad (2)$$

The second stage is to find a set of parameters in the neighbourhood of $\hat{\theta}$ to maximize the log likelihood of a subset of training data $\hat{D}$.

$$\bar{\theta} = \arg\max_{\theta\in\mathcal{N}(\hat{\theta})}\{log \prod_{s_i\sim s} P(t_i|s_i;\theta)\} \quad (3)$$

One thing to note is that the data size used for fine tuning is small, usually containing only a few mention-entity pairs. So we need to be careful about overfitting. To this end, we go over the tuning data for only one pass.

2.3 Generating and re-ranking

As the standard entities can be generated by the proposed sequence generative model, we now call the strategy "generating and re-ranking" (GR). As illustrated in Figure 5, instead of selecting candidates from the KB and re-ranking the semantic similarity between candidates with input mentions, we generate the candidates directly by adopting a beam search in our decoder. Given an input mention, the beam search decoder with beam size k will output k results, where each result may contain one or more entities due to the "multi-implication" problem. All these entities are grouped as a candidate set of the input mention and any duplicates are removed. The calculation of semantic similarity can be implemented by various neural network structures, such as ABCNN (Yin *et al.*, 2016), LSTM (Hochreiter and Schmidhuber, 1997; Limsopatham and Collier, 2016), and bidirectional encoder representations from Transformers (BERT) (Devlin *et al.*, 2019; Ji *et al.*, 2019). Considering the state-of-the-art performance of BERT in semantic similarity learning, which could also be applied to entity normalization tasks (Ji *et al.*, 2019), we calculate the semantic similarity between input mentions with candidates by adopting a BERT-based method as introduced

```
Algorithm 3: Obtain category-specific model
  Input: Training data
         $D = (s_1,t_1),(s_2,t_2),...,(s_n,t_n)$;
         Testing mention $\hat{s}$; General model $M_G$
  Output: category-specific model $M_C$ ;
1 Inference($M_G$,$\hat{s}$) $\rightarrow \hat{t}$ ;
2 getLableSet($\hat{t}$) $\rightarrow \hat{l}$ ;
3 initialize a training subset $\hat{D}$ ;
4 foreach $(s_i,t_i)$ in $D$ do
5     getLableSet($t_i$) $\rightarrow l_i$ ;
6     if $l_i \subset \hat{l}$ then
7         addItem($s_i,t_i$) $\rightarrow \hat{D}$ ;
8     end
9 end
10 Finetune($M_G,\hat{D}$) $\rightarrow M_C$ ;
```

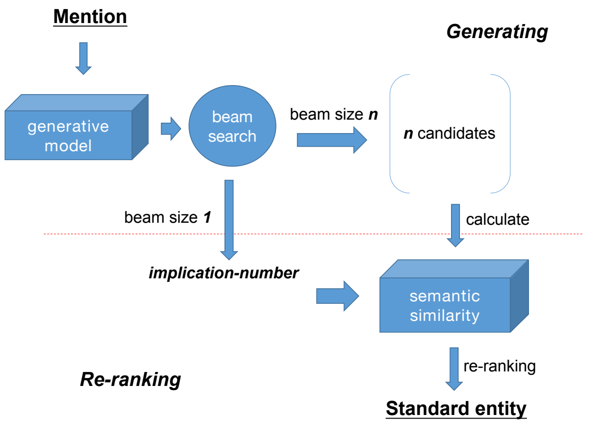


Figure 5: The overall framework of generating and re-ranking

by (Ji *et al.*, 2019). For the "multi-implication" mentions, we calculate the similarity score for each generated entity candidate and select the top n candidates as the final result of the linked standard entities of the input mention. The selecting number n is determined by the number of entities output by beam size 1.

3 Experimental settings

In this section, we describe the datasets used in our experiments, and the details of training and evaluation.

3.1 Data

The datasets used in this paper were provided from the CHIP 2019 clinical entity normalization task[1]. Table 1 shows examples of training data provided for the task. The special characters "##" in standard entities is used as a delimiter between each entity for multi-implication mentions. All the mentions in the training and evaluation datasets refer to clinical procedures extracted from Chinese electronic medical records, and their corresponding standard entities are annotated based on the ICD9-2017-PUMCH procedure codes knowledge base. The knowledge base contains 9,867 standard entities and each entity is assigned to a unique category label. Table 2 shows the detailed statistics on "uni-implication" and "multi-implication" procedure mentions. We annotated the data with category labels by using the given procedure codes dictionary, and arranged the data into the following formation.

- **Mention**: 双 侧 脑 深 部 电 刺 激 植 入 术
- **Standard entity**: label_02 脑 深 部 电 极 置 入 术 label_01 颅 神 经 刺 激 脉 冲 发 生 器 植 入

The category labels are annotated ahead of entities, and both mentions and standard entities are split into characters. Owing to the sparsity of training data, we augmented our training set by simply adding all the data in the given knowledge base and taking each of the standard entities as the mention of itself.

3.2 Training details

We used the Transformer toolkit[2] to implement all the described methods. The vocabulary is shared and its size is 1,550. We used the same configuration as *Transformer base* adopted by (Vaswani *et al.*, 2017), which contains a six-layer encoder and a six-layer decoder with 512-dimensional hidden representations. The mini-batch size was set to 128 and 150 training epochs. We used the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.0003. Owing to the high cost of pre-training BERT, we directly adopted *BERT-base*[3] parameters pre-trained by Google in the Chinese general corpus and fine-tuned the semantic similarity task (in Section 2.3) with a batch size of 32 and 30 training epochs.

Mentions	**Translation**	**Standard entities**	**Translation**
左甲状腺切除术	Left thyroidectomy	单侧甲状腺切除术	Unilateral thyroidectomy
输尿管镜检 查+扩张+置双J管术	Ureteroscopy + dilatation+Double J-stent placement	经尿道输尿管支架置入术##输尿管镜检查##输尿管扩张术	Transurethral ureteral stent placement ## Ureteroscopy## Ureteral dilatation
右侧甲状腺全切除术+左 侧甲状腺次全切除术	Right totalthyroidectomy + Left subthyroidectomy	单侧甲状腺切除伴他叶部分切除术	Unilateral thyroidectomy with partial lobectomy
双侧脑深部电刺激植入术	Bilateral deep brain stimulation implantation	脑深部电极置入术##颅神经刺激脉冲发生器植入	Deep brain electrode implantation ## Cranial nerve stimulation pulse generator implantation

Table 1: Examples of training data

Datesets	uni-implication	multi-implication	total
Train	3,801	199	4,000
Test	2889	111	3,000

Table 2: Data statistics

3.3 Baseline and evaluation metrics

We compared our models with three baselines, as follows.

- **Edit-distance**. A basic method focusing on the number of procedures transforming one string to another.
- **BERT-based ranking** (Devlin *et al.*, 2019). A typical "search and re-rank" method that gives the best performance on English biomedical entity normalization. Since the data used in this study are in Chinese, we replaced the original English pre-trained BERT model with a pre-trained Chinese model *BERT-base*.

[1]http://cips-chip.org.cn/evaluation
[2]https://github.com/Kyubyong/transformer
[3]https://github.com/google-research/bert

Method	uni-implication	multi-implication	total
Edit-distance	50.8	—	48.3
BERT-based ranking	88.6	—	84.2
Transformer$_{\text{base}}$	87.5	24.5	84.3
Transformer$_{\text{CD}}$	88.5	32.7	85.6
Transformer$_{\text{CD+refine}}$	90.2	40.9	87.7
Transformer$_{\text{CD+refine+GR}}$	**91.1**	**52.4**	**89.3**

Table 3: Performance of different methods in terms of accuracy. Emboldened scores denote the best performance.

- **Transformer**$_{\text{base}}$. The basic transformer model without the proposed constraint decoding and model-refining methods.

Following (Devlin *et al.*, 2019), we evaluate the performance of different methods in terms of accuracy, which was the percentage of entity mentions that were correctly normalized. Considering the "multi-implication" problem in our data, we adopt a more strict definition of accuracy as follow:

$$acc = \frac{\sum_{i=1}^{n} \lfloor p_i r_i \rfloor}{n} \quad (4)$$

n is the total number of test data, a_i and c_i represent the precision rate and recall rate of implicated standard entities of the ith mention respectively. In other word, the normalization result of a given procedure mention can only be correct when both the number and text of generated standard entities are completely correct.

4 Results and analysis

Table 3 presents the accuracy scores of the baseline and proposed models. The subscripts "CD", "refine", and "GR" correspond to the three methods proposed in Section 2. As shown in Table 3, all of the proposed models gave better results than the three baseline models in the total column. The proposed Transformer$_{\text{CD+refine+GR}}$ model, which applies all the methods proposed in Section 2, achieves the best accuracy in both uni-implication and multi-implication data, being over 5% better than that of the state-of-the-art discriminative model-based method BioBERT.

It can also be observed that confined to the "SA" strategy, although the BioBERT model can achieve competitive accuracy in uni-implication mention normalization, it cannot deal with multi-implication mentions at all. In contrast, including the baseline Transformer$_{\text{base}}$, all sequence generative models have the ability to normalize multi-implication mentions. Hence, our proposed models can achieve much better performance in both uni-implication and multi-implication mention normalization than the baseline models.

Methods	uni-implication	multi-implication	total
BERT-based ranking	100%	—	96.3%
Delimiter "+"	96.6%	70.3%	95.6%
Transformer$_{\text{CD+refine+GR}}$	98.6%	76.4%	97.7%

Table 4: Accuracy of implication-number prediction

Method	OOD rate
Transformer$_{\text{base}}$	8.3%
Transformer$_{\text{CD}}$	0

Table 5: Rate of generated entities out of dictionary (OOD)

4.1 Prediction of the number of references

We conducted further experiments to verify the ability of the proposed model to predict the number of standard entities that should be linked. For comparison, we adopt a text-feature based method to identifies the number of linking entities by simply thinking of "+" in a given procedure mention as a delimiter. Table 4 shows the accuracy of Transformer$_{\text{CD+refine+GR}}$, delimiter "+", and BioBERT in predicting the number of references. As BioBERT always gives one standard entity for each input procedure mention, its accuracy of prediction for uni-implication mentions is naturally 100% correct. The proposed sequence generative method Transformer$_{\text{CD+refine+GR}}$ can get 2% and 6% higher accuracy than simply adopting delimiter "+" in uni-implication and multi-implication mentions respectively. The results demonstrate that our method has good ability in the implication-number prediction of both uni-implication and multi-implication.

4.2 Effect of integrating category-based constraint decoding

A generated entity cannot be the correct standard entity if it is not contained in the given dictionary (or the KB referenced in the above sections), which we call the out-of-dictionary (OOD) problem. We are interested to see whether our proposed category-based constraint decoding can avoid the OOD problem. Table 5 presents the OOD rate of the proposed Transformer$_{\text{CD}}$ and baseline Transformer$_{\text{base}}$. As shown in Table 5, the OOD problem is com-

	Candidates per mention	Standard entity recall
BM25	10	86%
Beam size 2	2.4	88%
Beam size 3	3.7	88.7%
Beam size 5	5.2	89.1%
Beam size 7	7.3	89.2%

Table 6: Number of candidates per mention and rate of standard entity recall for the candidate set generated by different beam sizes of the proposed sequence generative model, compared against those selected by the traditional IR model BM25

pletely solved by category-based constraint decoding.

Figure 6 provides an illustrative example of entity normalization. In this example, the baseline model generates the character "头", which should never appear after a previous trace of "(label_77,[尺, 骨])". The proposed model avoids this error because the constraint set masks all unexpected characters based on the previous trace.

Input mention:	右尺骨头切除术 (Right ulna resection)
Reference:	尺骨部分切除术 (Partial ulna resection)
Transformer$_{\text{base}}$:	label_77 尺骨头切除
Transformer$_{\text{CD}}$:	label_77 尺骨部分切除术

Figure 6: Entity normalization example, in which the proposed method is able to obtain a realistic result while the baseline model does not.

4.3 Candidates generated by beam search decoder

As described in Section 2.3, we adopted a beam-search decoder to generate candidates instead of selecting candidates from a given dictionary. As we used the same BERT-based text similarity method, i.e., *BioBERT*, the quality of generated candidates has a decisive influence on the final performance. Table 6 reports the number of candidates per mention and the rate of standard entity recall for the candidate sets that were conducted using two types of strategy. For the traditional IR model BM25, which is used by *BioBERT*, the top 10 candidates are retrieved for each mention and a standard entity recall of 86% was obtained. Regarding the proposed method, although there is not much difference in standard entity recall between the candidate sets generated by different beam sizes, all candidate sets could achieve a better standard entity recall than BM25 with a much smaller number of candidates per mention, which proves that our GR strategy is more efficient.

Mention \ Entity	token	character
token	81.2	83.6
character	82.5	84.3

Table 7: Performance of different granularity

4.4 Character-based vs. token-based

We investigate the influence of different granularity, we compare four different combined conditions in Table 5. Here we use Transformer$_{\text{base}}$ (the 4th row in Table 3) for the sake of simplicity. However the token-based generated results are unsatisfactory (e.g. token2token with accuracy of 81.2% while the character-based baseline achieves 84.3%), in our view, from three faults: 1) Error propagation could be introduced by tokenizer because of the medical domain specificity. 2)Chinese is written without spaces between words and it is difficult to determine word boundaries. 3)token-based vocabulary brings more OOV problems.

5 Related work

There are two areas related to our work:

Entity normalization: Most of the entity normalization studies consider the domain-specific knowledge base or dictionary as the scope of standard entities. Early methods Bunescu and Paşca (2006) and Zheng *et al.* (2010) design discriminative features, such as the TF-IDF, to compare the similarity of candidate entity with entity description and feed to the ranking framework. Popular approaches Leaman *et al.* (2013); Limsopatham and Collier (2016); Li *et al.* (2017); Ji *et al.* (2019) handle this as a sentence-pair classification task. Leaman *et al.* (2013) first proposed a pairwise learning-to-rank technique that adopts a vector-space model to measure the text similarity between medical entity mentions and standard entity in KB. Deep neural networks have also been proposed to normalize biomedical entities. Limsopatham and Collier (2016) and Li *et al.* (2017) adopted a convolutional neural network (CNN) and a recurrent neural network (RNN) to present the deep semantic matching between query mentions and candidate entities in a KB. Kolitsas *et al.* (2018) proposed a neural end-to-end entity linking system that jointly discovers and links entities in a text document. Luo *et al.* (2018) propose a multi-task framework in the clinical setting to normalize the disease and procedure mentions jointly. Ji *et al.* (2019) proposed an entity normalization architecture by fine-tuning a pre-trained BERT. However, these studies all concentrate on normalizing the mention with one unique standard entity in the KB.

Sequence generation: Sutskever *et al.* (2014) was the first to propose an encoder-decoder method to facilitate end-to-end learning of the sequence generation. After that, many studies (Bahdanau *et al.*, 2014; Luong *et al.*, 2015; Vaswani *et al.*, 2017) were focused on perfecting the encoder-decoder architecture. Zhao *et al.* (2018) proposed a recommendation strategy to alleviate the erroneous translations problem in neural machine translation by integrating a phrase table. Different from general sentence generation, the scope of generated standard entities should be strictly restricted, and our model can extract the constraint from a given knowledgeable and integrate it into the decoder. Li *et al.* (2018) proposed a dynamic neural machine translation that gives each test sentence a best-fitting network parameter. This work uses a similarity search to obtain training data according to the test sentence.

6 Conclusions and Future Work

In this paper, we propose a sequence generative learning framework with a category-based constraint decoding and model-refining mechanism for Chinese medical procedure normalization. The proposed model can achieve the end-to-end generation of all corresponding standard entities for all types of input mentions, especially for "multi-implication" mentions. The "generating and re-ranking" strategy is employed to integrate the proposed generative model with a discriminative similarity re-ranking method to further improve normalization performance. Our comprehensive experimental results demonstrate that the proposed model significantly outperforms the baseline methods. Furthermore, the proposed model can be applied to the normalization of both "uni-implication" and "multi-implication" Chinese medical procedure mentions. Notwithstanding, considering the complexity of the domain specificity and the scarcity of training data, this challenging task is far from being solved. In the future, we plan to focus on how to improve the performance of medical entity normalization when resources are limited.

Acknowledgments

We thank the three anonymous reviewers for their careful reading of our paper and their many insightful comments and suggestions. We would like to thank Jian Sun, Si Chen, and Chen Xue for their invaluable contributions in shaping the early stage of this work. We thank Jinan Xu, Yang Zhao, and Junnan Zhu for their valuable and constructive suggestions.

References

Gustavo Aguilar, Suraj Maharjan, Adrián Pastor López-Monroy, and Thamar Solorio. A multi-task approach for named entity recognition in social media data. *arXiv preprint arXiv:1906.04135*, 2019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Razvan Bunescu and Marius Paşca. Using encyclopedic knowledge for named entity disambiguation. In *11th conference of the European Chapter of the Association for Computational Linguistics*, 2006.

Pan Deng, Haipeng Chen, Mengyao Huang, Xiaowen Ruan, and Liang Xu. An ensemble cnn method for biomedical entity normalization. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, pages 143–149, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

Jennifer D'Souza. A multi-pass sieve for name normalization. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Jennifer D'Souza and Vincent Ng. 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, 2015.

Wei Fang, Jianwen Zhang, Dilin Wang, Zheng Chen, and Ming Li. Entity disambiguation by knowledge and text jointly embedding. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*, pages 260–269, 2016.

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R Curran. Evaluating entity linking with wikipedia. *Artificial intelligence*, 194:130–150, 2013.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Zongcheng Ji, Qiang Wei, and Hua Xu. Bert-based ranking for biomedical entity normalization. *arXiv preprint arXiv:1908.03548*, 2019.

Ning Kang, Bharat Singh, Zubair Afzal, Erik M van Mulligen, and Jan A Kors. Using rule-based natural language processing to improve disease normalization in biomedical text. *Journal of the American Medical Informatics Association*, 20(5):876–881, 2012.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, 2018.

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

Haodi Li, Qingcai Chen, Buzhou Tang, Xiaolong Wang, Hua Xu, Baohua Wang, and Dong Huang. Cnn-based ranking for biomedical entity normalization. *BMC bioinformatics*, 18(11):385, 2017.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. One sentence one model for neural machine translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.

Nut Limsopatham and Nigel Collier. Normalising medical concepts in social media texts by learning semantic representation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1014–1023, 2016.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888, 2015.

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

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.

Kristina Toutanova. Competitive generative models with structure learning for nlp classification tasks. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 576–584. Association for Computational Linguistics, 2006.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010. Curran Associates Inc., 2017.

Yipei Wang, Xingyu Fan, Luoxin Chen, I Eric, Chao Chang, Sophia Ananiadou, Junichi Tsujii, and Yan Xu. Mapping anatomical related entities to human body parts based on wikipedia in discharge summaries. *BMC bioinformatics*, 20(1):1–11, 2019.

Gongqing Wu, Ying He, and Xuegang Hu. Entity linking: an issue to extract corresponding entity with knowledge base. *IEEE Access*, 6:6220–6231, 2018.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272, 2016.

Yang Zhao, Yining Wang, Jiajun Zhang, and Chengqing Zong. Phrase table as recommendation memory for neural machine translation. *arXiv preprint arXiv:1805.09960*, 2018.

Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. Learning to link entities with knowledge base. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 483–491, 2010.