A Neural Transition-based Joint Model for Disease Named Entity Recognition and Normalization

Zongcheng Ji[1], **Tian Xia**[1], **Mei Han**[1], and **Jing Xiao**[2]

[1]PAII Inc., Palo Alto, CA, USA
[2]Ping An Technology, Shenzhen, China
[1]{jizongcheng, SummerRainET2008, hanmei613}@gmail.com
[2]xiaojing039@pingan.com.cn

Abstract

Disease is one of the fundamental entities in biomedical research. Recognizing such entities from biomedical text and then normalizing them to a standardized disease vocabulary offer a tremendous opportunity for many downstream applications. Previous studies have demonstrated that joint modeling of the two sub-tasks has superior performance than the pipelined counterpart. Although the neural joint model based on multi-task learning framework has achieved state-of-the-art performance, it suffers from the boundary inconsistency problem due to the separate decoding procedures. Moreover, it ignores the rich information (e.g., the text surface form) of each candidate concept in the vocabulary, which is quite essential for entity normalization. In this work, we propose a neural transition-based joint model to alleviate these two issues. We transform the end-to-end disease recognition and normalization task as an action sequence prediction task, which not only jointly learns the model with shared representations of the input, but also jointly searches the output by state transitions in one search space. Moreover, we introduce attention mechanisms to take advantage of the text surface form of each candidate concept for better normalization performance. Experimental results conducted on two publicly available datasets show the effectiveness of the proposed method.

1 Introduction

Disease is one of the fundamental entities in biomedical research, thus it is one of the most searched topics in the biomedical literature (Dogan et al., 2009) and the internet (Brownstein et al., 2009). Automatically identifying diseases mentioned in a text (e.g., a PubMed article or a health webpage) and then normalizing these identified mentions to their mapping concepts in a standardized disease vocabulary (e.g., with primary name, synonyms and definition, etc.) offers a tremendous opportunity for many downstream applications, such as mining chemical-disease relations from the literature (Wei et al., 2015), and providing much more relevant resources based on the search queries (Dogan et al., 2014), etc. Examples of such disease vocabularies includes MeSH (http://www.nlm.nih.gov/mesh/) and OMIM (http://www.ncbi.nlm.nih.gov/omim).

Previous studies (Leaman and Lu, 2016; Lou et al., 2017; Zhao et al., 2019) show the effectiveness of the joint methods for the end-to-end disease recognition and normalization (aka linking) task to alleviated the error propagation problem of the traditional pipelined solutions (Strubell et al., 2017; Leaman et al., 2013; Xu et al., 2016, 2017). Although TaggerOne (Leaman and Lu, 2016) and the discrete transition-based joint model (Lou et al., 2017) successfully alleviate the error propagation problem, they heavily rely on hand-craft feature engineering. Recently, Zhao et al. (Zhao et al., 2019) proposes a neural joint model based on the multi-task learning framework (i.e., MTL-feedback) which significantly outperforms previous discrete joint solutions. MTL-feedback jointly shares the representations of the two sub-tasks (i.e., joint learning with shared representations of the input), however, their method suffers from the boundary inconsistency problem due to the separate decoding procedures (i.e., separate search in two different search spaces). Moreover, it ignores the rich information (e.g., the text surface form) of each candidate concept in the vocabulary, which is quite essential for entity normalization.

In this work, we propose a novel neural transition-based joint model named **NeuJoRN** for disease named entity recognition and normalization, to alleviate these two issues of the multi-task learning based solution (Zhao et al., 2019). We transform the end-to-end disease recognition and

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 2819–2827
August 1–6, 2021.

normalization task as an action sequence prediction task. More specifically, we introduce four types of actions (i.e., OUT, SHIFT, REDUCE, SEGMENT) for the recognition purpose and one type of action (i.e., LINKING) for the normalization purpose. Our joint model not only jointly learns the model with shared representations, but also jointly searches the output by state transitions in one search space. Moreover, we introduce attention mechanisms to take advantage of text surface form of each candidate concept for better linking action prediction.

We summarize our contributions as follows.

- We propose a novel neural transition-based joint model, NeuJoRN, for disease named entity recognition and normalization, which not only jointly learns the model with shared representations, but also jointly searches the output by state transitions in one search space.
- We introduce attention mechanisms to take advantage of text surface form of each candidate concept for normalization performance.
- We evaluate our proposed model on two public datasets, namely the NCBI and BC5CDR datasets. Extensive experiments show the effectiveness of the proposed model.

2 Task Definition

We define the end-to-end disease recognition and normalization task as follows. Given a sentence x from a document d (*e.g.,* a PubMed abstract) and a controlled vocabulary KB (*e.g.,* MeSH and OMIM) which consists of a set of disease concepts, the task of ***end-to-end disease recognition and normalization*** is to identify all disease mentions $M = \{m_1, m_2, ..., m_{|M|}\}$ mentioned in x and to link each of the identified disease mention m_i with its mapping concept c_i in KB, $m_i \rightarrow c_i$. If there is no mapping concept in KB for m_i, then $m_i \rightarrow NIL$, where NIL denotes that m_i is unlinkable.

3 Neural Transition-based Joint Model

We first introduce the transition system used in the model, and then introduce the neural transition-based joint model for this task.

3.1 Transition System

We propose a novel transition system, inspired by the arc-eager transition-based shift-reduce parser (Watanabe and Sumita, 2015; Lample et al., 2016), which constructs the output of each given sentence x and controlled vocabulary KB through state transitions with a sequence of actions A.

Table 1: Defined transition actions used in the proposed model. We use the subscript $i \in \{0, 1, ...\}$ to denote the item index in *stack* σ and *buffer* β, starting from right and left, respectively.

Actions	Change of State
OUT	$\frac{(\sigma\|\sigma_0,\beta_0\|\beta,O)}{(\sigma\|\sigma_0,\beta',O)}$
SHIFT	$\frac{(\sigma\|\sigma_1\|\sigma_0,\beta_0\|\beta,O)}{(\sigma\|\sigma_0\|\beta_0,\beta',O)}$
REDUCE	$\frac{(\sigma\|\sigma_1\|\sigma_0,\beta_0\|\beta,O)}{(\sigma\|\sigma_1\sigma_0,\beta_0\|\beta,O)}$
SEGMENT-t	$\frac{(\sigma\|\sigma_0,\beta_0\|\beta,O)}{(\sigma',\beta_0\|\beta,O\cup\sigma_0^t)}$
LINKING-c	$\frac{(\sigma\|\sigma_0,\beta_0\|\beta,O\|\sigma_0^t)}{(\sigma\|\sigma_0,\beta_0\|\beta,O\|\sigma_0^{t,c})}$

We define a state as a tuple (σ, β, O), which consists of the following three structures:

- *stack* (σ): the *stack* is used to store tokens being processed.
- *buffer* (β): the *buffer* is used to store tokens to be processed.
- *output* (O): the *output* is used to store the recognized and normalize mentions.

We define a start state with the *stack* σ and the *output* O being both empty, and the buffer β containing all the tokens of a given sentence x. Similarly, we define an end state with the *stack* σ and buffer β being both empty, and the *output* O saving the recognized and normalized entity mention. The transition system begins with a start state and ends with an end state. The state transitions are accomplished by a set of transition actions A, which consume the tokens in β and build the output O step by step.

As shown in Table 1, we define 5 types of transition actions for state transitions, and their logics are summarized as follows:

- OUT pops the first token β_0 from the *buffer*, which indicates that this token does not belong to any entity mention.
- SHIFT moves the first token β_0 from the *buffer* to the *stack*, which indicates that this token is part of an entity mention.

Table 2: An example of state transitions for the recognition and normalization of disease mentions given a sentence "Most colon cancers arise from mutations" and a controlled vocabulary MeSH. State 0 and 9 are the start state and end state, respectively, and ϕ denotes empty.

State	**Actions** A	**Stack** σ	**Buffer** β	**Output** O
0		ϕ	Most colon cancers arise from mutations	ϕ
1	OUT	ϕ	colon cancers arise from mutations	ϕ
2	SHIFT	colon	cancers arise from mutations	ϕ
3	SHIFT	colon | cancers	arise from mutations	ϕ
4	REDUCE	colon cancers	arise from mutations	ϕ
5	SEGMENT-disease	ϕ	arise from mutations	colon cancers^{disease}
6	LINKING-D003110	ϕ	arise from mutations	colon cancers^{disease,D003110}
7	OUT	ϕ	from mutations	colon cancers^{disease,D003110}
8	OUT	ϕ	mutations	colon cancers^{disease,D003110}
9	OUT	ϕ	ϕ	colon cancers^{disease,D003110}

- REDUCE pops the top two tokens (or spans) σ_0 and σ_1 from the *stack* and concatenates them as a new span, which is then pushed back to the *stack*.
- SEGMENT-t pops the top token (or span) σ_0 from the *stack* and creates a new entity mention σ_0^t with entity type t, which is then added to the *output*.
- LINKING-c links the previous recognized but unnormalized mention σ_0^t in the *output* with its mapping concept with id c and updates the mention with $\sigma_0^{t,c}$.

Table 2 shows an example of state transitions for the recognition and normalization of disease mentions given a sentence "Most colon cancers arise from mutations" and a controlled vocabulary MeSH. State 0 is the start state where ϕ denotes that the stack σ and output O are initially empty, and the buffer β is initialized with all the tokens of the given sentence. State 9 is the end state where ϕ denotes that the stack σ and buffer β are finally empty, and colon cancers^{disease,D003110} in the output O denote that the mention "colon cancers" is a disease mention and is normalized to the concept with id D003110 in MeSH. More specifically, state 5 creates a new disease mention colon cancers^{disease} and add it to the *output*. State 6 links the previous recognized but unnormalized disease mention in the *output* with its mapping concept with id D003110 in MeSH.

3.2 Action Sequence Prediction

Based on the introduced transition system, the end-to-end disease recognition and normalization task becomes a new sequence to sequence task, i.e., the action sequence prediction task. The input is a sequence of words $x_1^n = (w_1, w_2, ..., w_n)$ and a controlled vocabulary KB, and the output is a sequence of actions $A_1^m = (a_1, a_2, ..., a_m)$. The goal of the task is to find the most probable output action sequence A^* given the input word sequence x_1^n and KB, that is

$$A^* = \arg\max_A p(A_1^m | x_1^n, \text{KB}) \tag{1}$$

Formally, at each step t, the model predicts the next action based on the current state S_t and the action history A_1^{t-1}. Thus, the task is models as

$$(A^*, S^*) = argmax_{A,S} \prod_t p(a_t, S_{t+1} | A_1^{t-1}, S_t) \tag{2}$$

where a_t is the generated action at step t, and S_{t+1} is the new state according to a_t.

Let r_t denote the representation for computing the probability of the action a_t at step t, thus

$$p(a_t | r_t) = \frac{exp(w_{a_t}^\intercal r_t + b_{a_t})}{\Sigma_{a' \in \mathcal{A}(S_t)} exp(w_{a'}^\intercal r_t + b_{a'})} \tag{3}$$

where w_a and b_a denote the learnable parameter vector and bias term, respectively, and $\mathcal{A}(S_t)$ denotes the next possible valid actions that may be taken given the current state S_t.

Finally, the overall optimization function of the action sequence prediction task can be written as

$$\begin{aligned}(A^*, S^*) &= argmax_{A,S} \prod_t p(a_t, S_{t+1} | A_1^{t-1}, S_t) \\ &= argmax_{A,S} \prod_t p(a_t | r_t)\end{aligned} \tag{4}$$

3.3 Dense Representations

We now introduce neural networks to learn the dense representations of an input sentence x and each state in the whole transition process to predict the next action.

Input Representation We represent each word x_i in a sentence x by concatenating its character-level word representation, non-contextual word representation, and contextual word representation:

$$x_i = [v_i^{char}; v_i^w; \text{ELMo}_i] \tag{5}$$

where v_i^{char} denotes its character-level word representation learned by using a CNN network (Ma and Hovy, 2016), v_i^w denotes its non-contextual word representation initialized with Glove (Pennington et al., 2014) embeddings, which is pre-trained on 6 billion words from Wikipedia and web text, and ELMo_i denotes its contextual word representation initialized with ELMo (Peters et al., 2018). We can also explore the contextual word representation from BERT (Devlin et al., 2018) by averaging the embeddings of the subwords of each word. We leave it to the future work.

We then run a BiLSTM (Graves et al., 2013) to derive the contextual representation of each word in the sentence x.

State Representation At each step t in the transition process, let's consider the representation of the current state $S_t = (\sigma_t, \beta_t, A_t)$, where $\sigma_t = (..., \sigma_1, \sigma_0)$, $\beta_t = (\beta_0, \beta_1, ...)$ and $A_t = (a_{t-1}, a_{t-2}, ...)$.

The *buffer* β_t is represented with BiLSTM (Graves et al., 2013) to represent the words in the buffer:

$$b_t = \text{BiLSTM}([\beta_0, \beta_1, ...]) \tag{6}$$

The *stack* σ_t and the *actions* A_t are represented with StackLSTM (Dyer et al., 2015):

$$\begin{aligned} s_t &= \text{StackLSTM}([..., \sigma_1, \sigma_0]) \\ a_t &= \text{StackLSTM}([a_{t-1}, a_{t-2}, ...]) \end{aligned} \tag{7}$$

We classify all the actions defined in Table 1 into two categories corresponding to two different purposes, *i.e.,* the recognition and normalization purposes. OUT, SHIFT, REDUCE, SEGMENT-t are used for the recognition purpose, and LINKING-c is used for the normalization purpose. As shown in Figure 1(a) and 1(b), we define two different state representations for predicting the actions in different purposes.

Specifically, for predicting the actions in the recognition purpose, we represent the state as

$$r_t^{NER} = \text{ReLU}(W[s_t^1; s_t^0; b_t^0; a_t^{-1}] + d) \tag{8}$$

where ReLU is an activation function, W and d denote the learnable parameter matrix and bias term, respectively, and

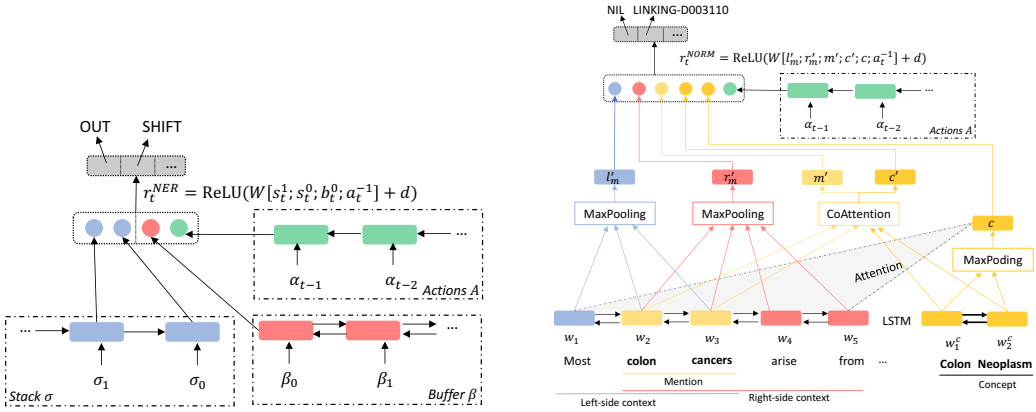
- s_t^0 and s_t^1 denote the first and second representations of the *stack* σ.
- b_t^0 denotes the first representation of the *buffer* β.
- a_t^{-1} denotes the last representation of the action history A.

For predicting the actions in the normalization purpose, we represent the state as

$$r_t^{NORM} = \text{ReLU}(W[l'_m; r'_m; m'; c'; c; a_t^{-1}] + d) \tag{9}$$

where ReLU is an activation function, W and d denote the learnable parameter matrix and bias term, respectively, and

- l'_m and r'_m denotes the left-side and right-side context representations by (i) first applying attention with the concept representation c to highlight the relevant parts in mentions' local context, and (ii) then applying max-pooling operation to aggregate the reweighted representations of all the context words.
- m' and c' are the representations of the mention and candidate concept by applying CoAttention mechanism (Tay et al., 2018; Jia et al., 2020).
- c denotes the candidate concept representation by (i) first run a BiLSTM (Graves et al., 2013) to derive the contextual representation of each word in the candidate concept, and (ii) then applying max-pooling operation to aggregate the representations of all concept words.
- a_t^{-1} denotes the last representation of the action history A.



(a) State representation for predicting recognition related actions, *i.e.,* SHIFT, OUT, REDUCE, SEGMENT-t.

(b) State representation for predicting normalization related actions, *i.e.,* LINKING-c.

Figure 1: State representations for predicting actions in different purposes (i.e., recognition and normalization).

3.4 Search and Training

Decoding is the key step in both training and test, which is to search for the best output structure (*i.e.,* action sequence) under the current model parameters. In this work, we use two different search strategies with different optimizations.

Greedy Search For efficient decoding, a widely-used greedy search algorithm (Wang et al., 2017) can be adopted to minimize the negative log-likelihood of the local action classifier in Equation (3, 8, 9).

Beam Search The main drawback of greedy search is error propagation (Wang et al., 2017). An incorrect action will fail the following actions, leading to an incorrect output sequence. One solution to alleviate this problem is to apply beam search. In this work, we use the Beam-Search Optimization (BSO) method with LaSO update (Wiseman and Rush, 2016) to train our beam-search model, where the max-margin loss is adopted.

4 Experiments

4.1 Datasets

We use two public available datasets in this study, namely NCBI - the NCBI disease corpus (Dogan et al., 2014) and BC5CDR - the BioCreative V CDR task corpus (Li et al., 2016b). NCBI dataset contains 792 PubMed abstracts, which was split into 692 abstracts for training and development, and 100 abstracts for testing. A disorder mention in each PubMed abstract was manually annotated with its mapping concept identifier in the MEDIC lexicon. BC5CDR dataset contains 1,500 PubMed abstracts, which was equally split into three parts for training, development and test, respectively. A disease mention in each abstract is manually annotated with the concept identifier to which it refers to a controlled vocabulary. In this study, we use the July 6, 2012 version of MEDIC, which contains 7,827 MeSH identifiers and 4,004 OMIM identifiers, grouped into 9,664 disease concepts. Table3 show the overall statistics of the two datasets.

Table 3: Overall statistics of the datasets.

corpus	#documents	#mentions	#concepts
NCBI	792	6,881	1,049
BC5CDR	1,500	12,852	5,818

To facilitate the generation of candidate linking actions, we perform some preprocessing steps of each candidate mention and each concept in KB with the following strategies: (i) *Spelling Correction* - for each candidate mention in the datasets, we replace all the misspelled words using a spelling check list as in previous work (D'Souza and Ng, 2015; Li et al., 2017). (ii) *Abbreviation Resolution* - we use Ab3p (Sohn et al., 2008) toolkit to detect and replace the abbreviations with their long forms within each document and also expand all possible abbreviated disease mentions using a dictionary collected from Wikipedia as in previous work (D'Souza and Ng, 2015; Li et al., 2017). (iii) *Numeric Synonyms Resolutions* - we replace all the numerical words in the mentions and concepts to their corresponding Arabic numerals as in previous work (D'Souza and Ng, 2015; Li et al., 2017).

Table 4: Architecture hyper-parameters.

Architecture hyper-parameters	
word embedding size	100
character embedding size	16
ELMo embedding size	1024
action embedding size	20
LSTM cell size	200
LSTM layers	2
dropout rate	0.2
learning rate	0.001
AdamW weight decay	0.00001
search top k	10

We generate candidate linking actions (*i.e.,* candidate concepts) for each mention with the commonly used information retrieval based method, which includes the following two steps. We first index all the concept names and training mentions with their concept ids. Then, the widely-used BM25 model provided by Lucene is employed to retrieve the top 10 candidate concepts $\{c_i\}_{i=1}^{10}$ for each mention m.

4.2 Evaluation Metrics and Settings

Following previous work (Leaman and Lu, 2016; Lou et al., 2017; Zhao et al., 2019), we utilize the evaluation kit[1] for evaluating the model performances. We report F1 score for the recognition task at the mention level, and F1 score for the normalization task at the abstract level.

We use the AdamW optimizer (Loshchilov and Hutter, 2019) for parameter optimization. Most of the model hyper-parameters are listed in Table 4. Since increasing the beam size will increase the decoding time, we only report results with beam size 1, 2, and 4.

4.3 Results and Discussion

4.3.1 Main results

Table 5 shows the overall comparisons of different models for the end-to-end disease named entity recognition and normalization task. The first part shows the performance of different pipelined methods for the task. DNorm (Leaman et al., 2013) is a traditional method, which needs feature engineering. IDCNN (Strubell et al., 2017) is a neural model based on BiLSTM-CRF, which requires few effort of feature engineering. The second part shows the performance of different joint models for the task. TaggerOne (Leaman et al., 2013) is a joint solution based on semi-CRF. Transition-based Model (Lou et al., 2017) is a joint solution based on discrete transition-based method. Both of these two models rely heavily on feature engineering. MTL-feedback (Zhao et al., 2019) is neural joint solution based on multi-task learning. NeuJoRN is our neural transition-based joint model for the whole task.

From the comparisons, we find that (1) IDCNN does not perform well enough although it relies few efforts of feature engineering. (2) All the joint models significantly outperform the pipelined methods. (3) The deep-learning based joint models significantly outperform the traditional machine learning based methods. (4) Our proposed NeuJoRN outperforms MTL-feedback by at least 0.57% and 0.59% on the recognition and normalization tasks, respectively.

4.3.2 Effectiveness of different search strategies

Table 6 shows the comparisons of different search strategies of our proposed NeuJoRN. From the results, we find that (1) The methods based on beam search strategies outperforms the greedy search strategy, which indicates that the beam search solutions could alleviate the error propagation problem of the greedy search solution. (2) The model with beam size 4 achieves the best performance. The larger the beam size, the better the performance, however the lower the decoding speed. (3) Our greedy search based solution doesn't outperform the MLT-feedback method.

4.3.3 Effectiveness of attention mechanisms

Table 7 shows the effectiveness of the proposed attention mechanisms. When we remove the attention mechanism for representing the left-side and right-side local context, the performance dropped a little bit. However, when we remove the CoAttention mechanism, which is used for directly modeling the matching between the mention and candidate concept, the performance dropped significantly. This group of comparisons indicates that importance of the matching between the mention and candidate concept for the entity normalization task.

[1] http://www.biocreative.org/tasks/biocreative-v/track-3-cdr

Table 5: Overall comparisons of different models for disease named entity recognition and normalization.

Method	NCBI		BC5CDR	
	Recognition	Normalization	Recognition	Normalization
DNorm (Leaman et al., 2013)	0.7980	0.7820	-	0.8064
IDCNN (Strubell et al., 2017)	0.7983	0.7425	0.8011	0.8107
TaggerOne (Leaman et al., 2013)	0.8290	0.8070	0.8260	0.8370
Transition-based Model (Lou et al., 2017)	0.8205	0.8262	0.8382	0.8562
MTL-feedback (Zhao et al., 2019)	0.8743	0.8823	0.8762	0.8917
NeuJoRN (Ours)	**0.8857**	**0.8882**	**0.8819**	**0.8986**

Table 6: Performance comparisons of different search strategies.

Method	NCBI		BC5CDR	
	Recognition	Normalization	Recognition	Normalization
greedy (b1)	0.8682	0.8792	0.8735	0.8866
beam (b1)	0.8734	0.8818	0.8765	0.8910
beam (b2)	0.8779	0.8843	0.8794	0.8949
beam (b4)	**0.8857**	**0.8882**	**0.8819**	**0.8986**

5 Related Work

Disease Named Entity Recognition DNER has been widely studied in the literature. Most previous studies (Leaman et al., 2013; Xu et al., 2015, 2016) transform this task as a sequence labeling task, and conditional random fields (CRF) based methods are widely adopted to achieve good performance. However, these methods heavily rely on hand-craft feature engineering. Recently, neural models such as BiLSTM-CRF based methods (Strubell et al., 2017; Wang et al., 2019) and BERT-based methods (Kim et al., 2019) have achieved state-of-the-art performance.

Disease Named Entity Normalization DNEN has also been widely studied in the literature. Most studies assume that the entity mentions are pre-detected by a separate DNER model, and focus on developing methods to improve the normaliation accuracy (Lou et al., 2017), resulting in developing rule-based methods (D'Souza and Ng, 2015), machine learning-based methods (Leaman et al., 2013; Xu et al., 2017), and recent deep learning-based methods (Li et al., 2017; Ji et al., 2020; Wang et al., 2020; Vashishth et al., 2021; Chen et al., 2021). However, the pipeline architecture which performs DNER and DNEN separately suffers from the error propagation problem. In this work, we propose a neural joint model to alleviate this issue.

Joint DNER and DNEN Several studies (Leaman and Lu, 2016; Lou et al., 2017; Zhao et al., 2019) show the effectiveness of the joint methods to alleviated the error propagation problem. Although TaggerOne (Leaman and Lu, 2016) and the discrete transition-based joint model (Lou et al., 2017) successfully alleviated the error propagation problem, they heavily rely on hand-craft feature engineering. Recently, Zhao et al. (Zhao et al., 2019) propose a neural joint model based on the multi-task learning framework (i.e., MTL-feedback) which significantly outperforms previous discrete joint solutions. However, their method suffers from the boundary inconsistency problem due to the separate decoding procedures (i.e., separate search in two different search spaces). Moreover, it ignores the rich information (e.g., the text surface form) of each candidate concept in the vocabulary, which is quite essential for entity normalization. In this work, we propose a neural joint model to alleviate these two issues.

Transition-based Models Transition-based models are widely used in parsing and translation (Watanabe and Sumita, 2015; Wang et al., 2018; Meng and Zhang, 2019). Recently, these models are successfully applied to information extraction tasks, such as joint POS tagging and dependency parsing (Yang et al., 2018), joint entity and relation extraction (Li and Ji, 2014; Li et al., 2016a; Ji et al., 2021). Several studies propose discrete transition-based joint model for entity recognition and normalization(Qian et al., 2015; Ji et al., 2016; Lou et al., 2017). In this work, we propose a neural transition-based joint model for disease named entity recognition and normalization.

Table 7: Performance comparisons of different attention mechanisms.

Method	NCBI		BC5CDR	
	Recognition	Normalization	Recognition	Normalization
beam (b4)	**0.8857**	**0.8882**	**0.8819**	**0.8986**
-Attention	0.8827	0.8868	0.8803	0.8964
-CoAttention	0.8673	0.8779	0.8729	0.8853

6 Conclusions

In this work, we proposed a novel neural transition-based joint model for disease named entity recognition and normalization. Experimental results conducted on two public available datasets show the effectiveness of the proposed method. In the future, we will apply this joint model to more different types of datasets, such as the clinical notes, drug labels, and tweets, etc.

References

John S Brownstein, Clark C Freifeld, and Lawrence C Madoff. 2009. Digital disease detection—harnessing the Web for public health surveillance. *New England Journal of Medicine*, 360(21):2153–2157.

Lihu Chen, Gaël Varoquaux, and Fabian M Suchanek. 2021. A Lightweight Neural Model for Biomedical Entity Linking. *AAAI*.

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

Rezarta Islamaj Dogan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: a resource for disease name recognition and concept normalization. *JBI*, 47:1–10.

Rezarta Islamaj Dogan, G Craig Murray, Aurélie Névéol, and Zhiyong Lu. 2009. Understanding PubMed® user search behavior through log analysis. *Database*, 2009:bap018.

Jennifer D'Souza and Vincent Ng. 2015. Sieve-Based Entity Linking for the Biomedical Domain. In *ACL*, pages 297–302.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. Transition-Based Dependency Parsing with Stack Long Short-Term Memory. In *ACL-IJCNLP*, pages 334–343, Beijing, China. Association for Computational Linguistics.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *IEEE ICASSP*, pages 6645–6649.

Zongcheng Ji, Omid Ghiasvand, Stephen Wu, and Hua Xu. 2021. A Discrete Joint Model for Entity and Relation Extraction from Clinical Notes. In *AMIA 2021 Informatics Summit*.

Zongcheng Ji, Aixin Sun, Gao Cong, and Jialong Han. 2016. Joint Recognition and Linking of Fine-Grained Locations from Tweets. In *WWW*, pages 1271–1281.

Zongcheng Ji, Qiang Wei, and Hua Xu. 2020. BERT-based Ranking for Biomedical Entity Normalization. In *AMIA 2020 Informatics Summit*, pages 269–277.

Ningning Jia, Xiang Cheng, Sen Su, and Liyuan Ding. 2020. CoGCN: Combining co-attention with graph convolutional network for entity linking with knowledge graphs. *Expert Systems*, page e12606.

Donghyeon Kim, Jinhyuk Lee, Chan Ho So, Hwisang Jeon, Minbyul Jeong, Yonghwa Choi, Wonjin Yoon, Mujeen Sung, and Jaewoo Kang. 2019. A Neural Named Entity Recognition and Multi-Type Normalization Tool for Biomedical Text Mining. *IEEE Access*, 7:73729–73740.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *NAACL*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Robert Leaman, Rezarta Islamaj Dogan, and Zhiyong Lu. 2013. DNorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, 29:2909–2917.

Robert Leaman and Zhiyong Lu. 2016. TaggerOne: joint named entity recognition and normalization with semi-Markov Models. *Bioinformatics*, 32(18):2839–2846.

Fei Li, Yue Zhang, Meishan Zhang, and Donghong Ji. 2016a. Joint models for extracting adverse drug events from biomedical text. In *IJCAI*, pages 2838–2844.

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

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016b. BioCreative V CDR task corpus:

a resource for chemical disease relation extraction. *Database*, 2016:baw068.

Qi Li and Heng Ji. 2014. Incremental Joint Extraction of Entity Mentions and Relations. In *ACL*, pages 402–412.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *ICLR (Poster)*. OpenReview.net.

Yinxia Lou, Yue Zhang, Tao Qian, Fei Li, Shufeng Xiong, and Donghong Ji. 2017. A Transition-based Joint Model for Disease Named Entity Recognition and Normalization. *Bioinformatics*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *ACL*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Fandong Meng and Jinchao Zhang. 2019. DTMT: A Novel Deep Transition Architecture for Neural Machine Translation. In *AAAI*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *NAACL-HLT*, pages 2227–2237.

Tao Qian, Yue Zhang, Meishan Zhang, Yafeng Ren, and Dong-Hong Ji. 2015. A Transition-based Model for Joint Segmentation, POS-tagging and Normalization. In *EMNLP*, pages 1837–1846.

Sunghwan Sohn, Donald C Comeau, Won Kim, and W John Wilbur. 2008. Abbreviation definition identification based on automatic precision estimates. *BMC Bioinformatics*, 9.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and Accurate Entity Recognition with Iterated Dilated Convolutions. In *EMNLP*, pages 2670–2680. Association for Computational Linguistics.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Hermitian Co-Attention Networks for Text Matching in Asymmetrical Domains. In *IJCAI*, pages 4425–4431. ijcai.org.

Shikhar Vashishth, Denis Newman-Griffis, Rishabh Joshi, Ritam Dutt, and Carolyn Rose. 2021. Improving Broad-Coverage Medical Entity Linking with Semantic Type Prediction and Large-Scale Datasets. *arXiv preprint arXiv:2005.00460*.

Qiong Wang, Zongcheng Ji, Jingqi Wang, Stephen Wu, Weiyan Lin, Wenzhen Li, Li Ke, Guohong Xiao, Qing Jiang, Hua Xu, and Others. 2020. A study of entity-linking methods for normalizing Chinese diagnosis and procedure terms to ICD codes. *JBI*, page 103418.

Shaolei Wang, Wanxiang Che, Yue Zhang, Meishan Zhang, and Ting Liu. 2017. Transition-Based Disfluency Detection using LSTMs. In *EMNLP*, pages 2785–2794, Copenhagen, Denmark. Association for Computational Linguistics.

Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, Curtis Langlotz, and Jiawei Han. 2019. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, 35(10):1745–1752.

Yuxuan Wang, Wanxiang Che, Jiang Guo, and Ting Liu. 2018. A Neural Transition-Based Approach for Semantic Dependency Graph Parsing. In *AAAI*.

Taro Watanabe and Eiichiro Sumita. 2015. Transition-based Neural Constituent Parsing. In *ACL*, pages 1169–1179, Beijing, China. Association for Computational Linguistics.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Jiao Li, Thomas C Wiegers, and Zhiyong Lu. 2015. Overview of the BioCreative V chemical disease relation (CDR) task. In *Proceedings of the fifth BioCreative challenge evaluation workshop*, pages 154–166.

Sam Wiseman and Alexander M Rush. 2016. Sequence-to-Sequence Learning as Beam-Search Optimization. In *EMNLP*, pages 1296–1306, Austin, Texas. Association for Computational Linguistics.

Jun Xu, Hee-Jin Lee, Zongcheng Ji, Jingqi Wang, Qiang Wei, and Hua Xu. 2017. UTH_CCB System for Adverse Drug Reaction Extraction from Drug Labels at TAC-ADR 2017. In *Proceedings of Text Analysis Conference*.

Jun Xu, Yonghui Wu, Yaoyun Zhang, Jingqi Wang, Hee-Jin Lee, and Hua Xu. 2016. CD-REST: a system for extracting chemical-induced disease relation in literature. *Database*.

Jun Xu, Yaoyun Zhang, Jingqi Wang, Yonghui Wu, Min Jiang, Ergin Soysal, and Hua Xu. 2015. UTH-CCB: The Participation of the SemEval 2015 Challenge - Task 14. In *SemEval*, pages 311–314.

Liner Yang, Meishan Zhang, Yang Liu, Maosong Sun, Nan Yu, and Guohong Fu. 2018. Joint POS Tagging and Dependence Parsing With Transition-Based Neural Networks. *TASLP*, 26(8):1352–1358.

Sendong Zhao, Ting Liu, Sicheng Zhao, and Fei Wang. 2019. A Neural Multi-Task Learning Framework to Jointly Model Medical Named Entity Recognition and Normalization. In *AAAI*, pages 817–824.