Bioinformatics, 33(15), 2017, 2363–2371
doi: 10.1093/bioinformatics/btx172
Advance Access Publication Date: 24 March 2017
Original Paper

OXFORD

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/33/15/2363/3089942 by NARASENTAN KAGAKUGIJYUTSU DAIGAKUIN DAIGAKU user on 09 November 2018

Data and text mining

A transition-based joint model for disease named entity recognition and normalization

Yinxia Lou[1,3], Yue Zhang[2], Tao Qian[4], Fei Li[1], Shufeng Xiong[5] and Donghong Ji[1,*]

[1]Computer School, Wuhan University, Wuhan, 430072, China, [2]Singapore University of Technology and Design, [3]School of Computer and Information Technology, Shangqiu Normal University, Shangqiu, 476000, China, [4]College of Computer Science and Technology,Hubei University of Science and Technology, Xianning, 437000, China and [5]Pingdingshan University, Pingdingshan, 467000, China

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on July 14, 2016; revised on March 2, 2017; editorial decision on March 21, 2017; accepted on March 23, 2017

Abstract

Motivation: Disease named entities play a central role in many areas of biomedical research, and automatic recognition and normalization of such entities have received increasing attention in biomedical research communities. Existing methods typically used pipeline models with two independent phases: (i) a disease named entity recognition (DER) system is used to find the boundaries of mentions in text and (ii) a disease named entity normalization (DEN) system is used to connect the mentions recognized to concepts in a controlled vocabulary. The main problems of such models are: (i) there is error propagation from DER to DEN and (ii) DEN is useful for DER, but pipeline models cannot utilize this.

Methods: We propose a transition-based model to jointly perform disease named entity recognition and normalization, casting the output construction process into an incremental state transition process, learning sequences of transition actions globally, which correspond to joint structural outputs. Beam search and online structured learning are used, with learning being designed to guide search. Compared with the only existing method for joint DEN and DER, our method allows nonlocal features to be used, which significantly improves the accuracies.

Results: We evaluate our model on two corpora: the BioCreative V Chemical Disease Relation (CDR) corpus and the NCBI disease corpus. Experiments show that our joint framework achieves significantly higher performances compared to competitive pipeline baselines. Our method compares favourably to other state-of-the-art approaches.

Availability and Implementation: Data and code are available at https://github.com/louyinxia/jointRN

Contact: dhji@whu.edu.cn

1 Introduction

Disease named entities play central roles in many lines of biomedical research and healthcare, such as aetiology, entity summarization, knowledge discovery and medical diagnosis (Joseph *et al.*, 2015; Khan *et al.*, 2013; Wei *et al.*, 2015a). With the rapid increase of biomedical and clinical texts, automatically recognizing disease named entities and normalizing such entities to a controlled vocabulary has attracted increasing research interest (Chowdhury *et al.*, 2010; Kang *et al.*, 2013; Leaman *et al.*, 2008; Sahu and Anand, 2016).

Two subtasks are involved in the task. First, *disease named entity recognition* (DER) aims to recognize disease named entity mentions in raw biomedical texts. State-of-the-art systems for DER employ

© The Author 2017. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

conditional random fields (CRFs) (Chowdhury *et al.*, 2010; Leaman *et al.*, 2008; Wei *et al.*, 2015b). Second, because a disease entity in biomedical text has many types of naming conventions, including abbreviations and acronyms, morphological or orthographical variations, new disease names and synonyms, entity-to-concept mapping is performed as a subsequent task. *Disease named entity normalization* (DEN) maps recognized disease entity mentions to concepts in a controlled vocabulary (e.g. MEDIC vocabulary Davis *et al.*, 2015) (Chowdhury *et al.*, 2010; Kate, 2016; Leaman *et al.*, 2015; Lee *et al.*, 2016). For example, the three disease mentions *renal toxicity*, *renal damage* and *nephrotoxicity* all map to a normalized concept *kidney diseases*, which is listed in MEDIC vocabulary.

Most existing solutions address the problem in two separate steps in a pipeline, by first recognizing disease entity mentions in biomedical texts and then normalizing the recognized mentions into concepts (Chowdhury *et al.*, 2010; Ghiasvand and Kate, 2014; Kang *et al.*, 2013; Leaman *et al.*, 2013; Lee *et al.*, 2016). However, pipeline approaches face two challenges: (i) they can lead to error propagation from DER to DEN and (ii) DEN can be useful for assisting DER, but pipeline approaches cannot utilize such information. In particular, one type of typical errors of a DER system is to mistakenly predict two disease mentions as a single mention. For example, 'hemorrhagic cystitis' may be taken as a single disease mention, instead of two disease mentions 'hemorrhagic' and 'cystitis'. DEN can likely rectify the type of errors since there is no normalizable concept in a controlled vocabulary for 'hemorrhagic cystitis', but normalizable concepts for 'hemorrhagic' and 'cystitis', respectively. Another type of errors for DER systems is to leave off one or more words from a multi-word mention, which causes the normalization system to normalize the result to an incorrect concept or NULL. Here again normalization knowledge can inform DER in the reverse direction.

The above observation inspires us to investigate a joint model for DER and DEN, recognizing and normalizing entities simultaneously. A recent attempt for such joint modelling was made by Leaman and Lu (2016), who use a semi-Markov model with dynamic programming. Their model gives better results compared to a pipeline baseline. However, exact inference can be intractable if features are defined over long-range dependencies, which prevents non-local features from being used. Such non-local features can be highly informative for both DER and DEN. For example, if one disease entity has been successfully recognized in the beginning of a medical document, the normalized form can be used as a feature to guide further recognition and normalization of the same entity mention in the document, or even different mentions that can be normalized into the same form. Such dependencies can be unbounded within a sentence, and are beyond the limit of tractable dynamic programs if used as DER features. In the NLP literature, transition-based models with global optimization and beam-search (Zhang and Clark, 2011) have been exploited for joint structured prediction tasks. Such models map structured output construction into an incremental state-transition process. Free from optimality constraints, beam-search allows non-local features to be extracted from each state to guide the next transition action. Similar to CRFs, the score of a whole sequence of transition actions is trained globally in order to resolve structural ambiguities and avoids label bias. In addition, training is designed to fix heuristic search errors, which results in learning-guided linear-time search. Since our joint task has a complex integrated search space and can benefit from very non-local dependency features. We investigate the effectiveness of this framework on our joint task, designing a transition system for joint DER and DEN, and exploring a set of novel features that are far more non-local compared with those employed by Leaman and Lu (2016). For example, our model can leverage information on existing normalized entities for recognizing new entities, the dependency between which can span cross a whole sentence. In contrast, Leaman and Lu (2016)'s features are constrained to only local entities themselves.

We evaluate the performance of our model on the BC5CDR corpus released by the BioCreative V Chemical Disease Relation (BC5CDR) shared task (Wei *et al.*, 2015a), and the NCBI disease corpus. In particular, we design a pipeline baseline system that includes a transition-based DER submodel and a transition-based DEN submodel, both of which give competitive accuracies. The two submodels are then integrated by extending the transition system of the baseline DER submodel, combining the DER and DEN features. On the BC5CDR corpus, our model achieve a 86.23% *F*-score for DER and 87.61% for DEN, improving the best result in the official evaluation by 1.2%. On the NCBI disease corpus, our system achieves a *F*-score of 82.05% for DER and 82.62% for DEN, which improves the *F*-score by 2.25% for DER and 4.42% for DEN, respectively, compared with the pipeline baseline.

2 Materials and methods

2.1 Disease named entity recognition (DER)

We present a transition-based model for our baseline recognizer. Given an input sentence $x \in \chi$, the recognizer finds all mentioned disease named entities by maximizing:

$$F(x) = \underset{y \in Gen(x)}{\operatorname{argmax}} \ \Phi(y) \cdot \vec{w}, \tag{1}$$

where $Gen(x)$ represents all possible disease named entities for the input sentence, and $\Phi(y)$ denotes the feature vector that characterizes possible outputs. $\vec{w}$ is parameter vector of the model. $F(x)$ is the best output according to the global feature vector $\Phi(y)$ and $\vec{w}$.

A transition system for DER can be formalized as a quadruple $M = (C, T, c_s, C_t)$, which outputs the corresponding action sequence for constructing $F(x)$. The quadruple M is specified as follows:

- C is the set of states.
- T is the set of transition actions, each of which is a function t: C→C.
- c_s is an initial state.
- C_t is a set of terminal states.

Our model processes an input sentence from left to right, with an index being maintained for the current word. A state of the transition-based recognizer is represented by a tuple $ST = (S, W)$, where S contains partially recognized sequences from the start to the current word, and $W = (w_i, w_{i+1}, \ldots, w_n)$ is the sequence of input words that have not been processed. The initial state c_s contains an empty stack and the full input as in coming words. When the word w_i is being processed, the transition system can take one of the three actions below:

- APP(w_i), which removes w_i from W, and appends w_i to the last partial entity on S.
- SEPN(w_i), which removes w_i from W, and adds w_i as a non-DE onto S.
- SEPY(w_i), which removes w_i from W, and adds w_i as a new disease entity on S.

For example, given the sentence 'Increase of parkinson disability after fluoxetine medication.', the sequences of action 'SEPN (Increase), SEPN (of), SEPY (parkinson), APP (disability), SEPN

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/33/15/2363/3089942 by NARASENTAN KAGAKUGIJYUTSU DAIGAKUIN DAIGAKU user on 09 November 2018

Table 1. Recognition and normalization process of the sentence 'Increase of parkinson disability after fluoxetine medication'

Step	Action	S	W	controlled vocabulary
0	–	∅	Increase of parkinson disability ⋯	Epileptic—epilepsy asthmatics—asthma parkinson disability—movement disorders ⋯⋯
1	SEPN(Increase)	Increase_O	of parkinson disability after ⋯	
2	SEPN(of)	$\text{Increase}_O\ \text{of}_O$	parkinson disability after fluoxetine ⋯	
3	SEPY(parkinson)	$\text{Increase}_O\ \text{of}_O\ \text{parkinson}_B$	disability after fluoxetine medication ⋯	
4	APP(disability)	$\cdots\ \text{of}_O\ \text{parkinson}_B\ \text{disability}_I$	after fluoxetine medication. ⋯	
5	SEPNNOR(after, movement disorders)	$\cdots\ \text{of}_O\ \text{movement}_B\ \text{disorders}_I\ \text{after}_O$	fluoxetine medication.	
6	SEPN(fluoxetine)	$\cdots\ \text{after}_O\ \text{fluoxetine}_O$	medication.	
7	SEPN(medication)	$\cdots\ \text{fluoxetine}_O\ \text{medication}_O$	.	
8	SEPN(.)	$\cdots\ \text{fluoxetine}_O\ \text{medication}_O.O$	∅	

(after), SEPN (fluoxetine), SEPN (medication), SEPN (.)' can be used to analyze its structure.

2.2 Joint recognition and normalization

We propose a transition-based model for joint DER and DEN by extending the transition-based recognizer, where the input is a sentence and the outputs include recognized and normalized disease entities. In addition to the actions APP, SEPN and SEPY, the actions of the joint transition system also include:

- SEPYNOR(w_i, *concept*), which replaces the last disease entity on the stack S with its concept listed in a dictionary, and takes the action SEPY(w_i).
- SEPNNOR(w_i, *concept*), which replaces the last disease entity on S with its concept listed in a dictionary, and takes the action SEPN(w_i).

Given the sentence 'Increase of parkinson disability after fluoxetine medication.', a correct output can be derived using the action sequence 'SEPN (Increase), SEPN (of), SEPY (parkinson), APP (disability), SEPNNOR (after, movement disorders), SEPN (fluoxetine), SEPN (medication), SEPN (.)', as shown in Table 1. For extracting rich features, we also associate each word on the stack S with BIO tags, which denote the beginning, intermediate, and outside of a disease entity, respectively (Wei *et al.*, 2015b).

2.3 Decoding

We apply beam-search for decoding, using an agenda to keep the B-best state transition sequences, represented as a state items, during the incremental process. The agenda is initialized with the initial state item c_s. When a word is processed, each state in the agenda is extended by applying all possible actions, resulting in a set of new states, which are scored and ranked, with the top B being used for the agenda for the next step. The same process repeats until all input words are processed, and the highest scored state in the agenda is taken as the output.

Pseudocode for the decoder is shown in Algorithm 1. The inputs of the algorithm include a raw sentence and a controlled vocabulary, and the outputs are the best recognized and normalized entities. *LEN* returns the number of words in a sentence, and *sent[idx]* returns the i_{th} word from the sentence. The variable *cand* represents a state item, which is a pair, consisting of the partially recognized and normalized sequence and the remaining input word sequence. INITIALIZE(agenda) initializes an agenda, ADDITEM adds a new item into an agenda, *B-BEST* returns the B highest scored state items from the agenda, and BEST returns the highest scored state item from the agenda. APP, SEPN, SEPY, SEPYNOR and SEPNNOR are the transition action defined in Sections 2.1 and 2.2.

Algorithm 1: The incremental beam-search decoder

```
Input: sent, controlled vocabulary//sent: informal sentence
Output: Best recognized and normalized sentence
INITIALIZE(agenda)
for idx in [0.LEN(sent)]:
    for cand in agenda:
        new ← App(cand, sent[idx])
        ADDITEM(agenda, new)
        new ← SepN(cand, sent[idx])
        ADDITEM(agenda, new)
        new ← SEPY(cand, sent[idx])
        ADDITEM(agenda, new)
        if(cand.lastEntity.type= Disease)
        norEntities ← LookUpKConcepts(cand.lastEntity)
        for entityin norEntities
            new ← SEPYNOR(cand,sent[idx],entity)
            ADDITEM(agenda, new)
            new ← SEPNNOR(cand,sent[idx],entity)
            ADDITEM(agenda, new)
    agenda ← B-BEST(agenda)
return BEST(agenda)
```

In Algorithm 1, LookUpKConcepts returns the top-K most similar concepts of a disease mention. The similarity between a mention and a concept is measured by the Levenshtein distance (Levenshtein, 1966). However, this method sometimes cannot give the correct concept of a mention. For example, the Levenshtein distance between the mention 'dystonic' and the concept 'dystonia' is smaller than that between 'dystonic' and 'dystonic disorders', but the concept of 'dystonic' is 'dystonic disorders' instead of 'dystonia'. To overcome the issue, we adopt the following two strategies.

i. We construct an extended mention set (*EM*) of a disease mention using the following heuristic rules based on their importance.
   1. The original form of a disease mention is the first element of *EM*.
   2. Synonyms (The list of synonyms of a disease mention are obtained from the training data) of a disease mention are added to *EM*. For example, the synonyms of the 'breast cancers' are 'breast neoplasms', 'breast carcinoma', and so on.

ii. We use an improved version of Levenshtein distance to obtain the similarity between each element in *EM* and each concept in

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/33/15/2363/3089942 by NARASENTAN KAGAKUGIJYUTSU DAIGAKUIN DAIGAKU user on 09 November 2018

the MEDIC disease vocabulary. The scoring function is calculated as follows:

$$\sigma(c_j) = \max_{j \epsilon s}\left(1 - \frac{LED(m_i^r, c_j)}{\max_{m_i^r \epsilon EM}\left(LED(m_i^r, c_j)\right)}\right), \quad (r = 1,2; i = 1,2,\ldots n;) \tag{2}$$

where s denotes the number of concepts from the MEDIC disease vocabulary, $EM = \{m_1^0, m_2^1, m_3^1, \ldots, m_n^r\}$ and m_i^r denotes the i_{th} concept based on r_{th} heuristic rule, r∈ {1,2}. c_j is a concept in the MEDIC disease vocabulary, $LED(m_i^r, c_j)$ denotes the Levenshtein distance between m_i^r and c_j. $\max_{m_i^r \epsilon EM}(LED(m_i^r, c_j))$ represents the maximum Levenshtein distance between m_i^r and c_j. Finally, we use the top K candidates as candidate concepts.

2.4 Training

Algorithm 2 shows pseudocode of the training algorithm. The weight values are initialized as all zeros before training. Here, BEAMSEARCH is identical to the decoding algorithm described in Algorithm 1 except that if y', the prefix of the gold standard y, falls out of the beam after any execution of the *B-BEST* function, the top assignment z and y' are returned for parameter update.

To estimate the parameter vector $\vec{w}$ of the model given a set of training data, we follow Zhang and Clark (2011) and use the generalized perceptron algorithm (Collins, 2002) as the learning framework, together with the 'early update' mechanism of Collins and Roark (2004).

Algorithm 2: Perceptron algorithm with beam-search and early-update, y' is the prefix of the gold-standard and z is the top assignment

Input: training set $\mathcal{D} = \{(x^i, y^i)\}_{i=1}^{N}$,
maximum iteration number T
Initialization: set $\vec{w} = 0$
Algorithm:
for t = 1… T
for $(x, y) \in \mathcal{D}$
$(x, y', z) \leftarrow$ BEAMSEARCH$(x, y, \vec{w})$
if $(z \neq y')$
$\vec{w} = \vec{w} + \Phi(y') - \Phi(z)$
Output: *model parameters* $\vec{w}$

2.5 Feature templates

We design a set of feature templates for our joint model according to the characteristics of biomedical text and disease entities. The feature templates are divided into two groups, as shown in Table 2. Templates 1–6 contain only DER information, Templates 7–10 and 15 contain only DEN information, and Templates 11–14 contain both DER and DEN information. Each template is instantiated according to the current word in the decoding process. Entry in the column 'For' shows the conditions for template instantiation, where 'a', 's' and 'n' indicate that the next action is APP, SEPN|SEPY and SEPYNOR|SEPNNOR, respectively. As a result, feature templates 1–2 are instantiated when the current word is a part of an entity, and feature templates 3–15 are instantiated when the current word starts a new entity or is not an entity.

In the feature templates, e, w, t and n are used to represent an entity, a word, a POS-tag and a normal form, respectively. The label l represents the associated BIO tag of the word w. The subscripts are based on the current word, where w_{-1} and e_{-1} represent the first word and entity to the left of the current word, respectively, t_{-2} represents the POS-tag on the second word to the left of the current word, and so on. *start*(e), *end*(e) and *len*(e) represent the first word, the last word and the length of an entity e, respectively. Feature template 2 represents whether the current entity is a normalized entity in the controlled dictionary. Feature template 5 includes the 4-character prefixes and suffixes of w_{-1}, and whether w_{-1} is a common disease word. Templates 1, 3 and 7 concern partial entity information, whose role in the model is to indicate the likelihood that the partial entity including the current word will become a correct full entity. They act to guide the next action to take according to the context.

Some feature templates, including Templates 11–14, consist of the atomic features of both subtasks, thus containing DER and DEN information simultaneously. For example, $n_{-1}l_{-2}$ in Template 8 represents the combination of the first segment (either a normalized entity or common segment) and the second label to the left of the current word. Non-local features are crucial for our joint model since they are able to capture the interactions of DER and DEN subtasks, and such interactions are beneficial to both of them. For instance, the sentence 'Drug-induced *hepatotoxicity* is a common cause of acute *hepatitis*.', the first entity hepatotoxicity can help recognizing and normalizing of the second entity hepatitis. Note that the dependency range between two related entities within a sentence can be unbounded.

Compared with the features of Leaman and Lu (2016), which are confined to recognition and normalization of multiple entities, our features are highly non-local. For example, the document '*Takotsubo syndrome* (or *apical ballooning syndrome*) secondary to Zolmitriptan. *Takotsubo syndrome* (*TS*), also known as broken heart syndrome, ……', Leaman and Lu (2016) did not use the first disease entity *Takotsubo syndrome* as a feature for recognizing the subsequent entities *apical ballooning syndrome*, *Takotsubo syndrome* and *TS*, which contain repeated terms. In contrast, our model leveraged non-local records to related multiple entities, the former entity, as a feature, can help recognizing and normalizing of the latter entities, which can in turn help correct the normalization of the former one.

Template 15 represents n-gram language models. Previous work has shown that semantic information is important for text normalization (Doddington, 2002; Kaji and Kitsuregawa, 2014). We extract language model features following Sampo *et al.* (2013).

Table 2. Feature templates

	Feature templates	For
1	w_0l_0, $w_{-1}w_0$, $w_{-1}w_0l_0$, $w_{-1}l_{-1}w_0l_0$	a
2	$isCTDdisease(e_0)$	a
3	$w_{-1}w_0t_0$, w_0t_0, $l_{-1}l_0w_0$	s
4	$w_{-2}w_{-1}t_{-2}t_{-1}l_{-1}$, $e_{-1}l_{-1}$	s
5	$prefix(w_{-1})$, $suffix(w_{-1})$, $isCommonDisease(w_{-1})$	s
6	$end(e_{-2})end(e_{-1})$	s
7	$n_{-1}w_0$, $start(n_{-1})w_0$	n
8	$len(n_{-2})n_{-1}$, $end(n_{-2})n_{-1}$, $end(n_{-2})end(n_{-1})$	n
9	n_{-1}, $n_{-1}n_{-2}$, $start(n_{-1})end(n_{-1})and\ len(n_{-1}) > 1$	n
10	$end(n_{-1})len(n_{-1})$, $n_{-2}len(n_{-1})$, $end(n_{-2})len(n_{-1})$	n
11	$n_{-1}l_0$, $n_{-1}l_{-1}$, $n_{-1}l_{-2}$, $start(n_{-1})l_{-1}$	s,n
12	$end(n_{-1})l_{-1}$, $end(n_{-2})n_{-1}l_{-1}$	s, n
13	$l_{-2}l_{-1}len(n_{-1})$, $n_{-2}l_{-1}l_0$, $l_{-1}l_0len(n_{-1})$	s, n
14	$l_{-2}l_{-1}l_0len(n_{-1})$, $l_{-2}n_{-1}l_0$	s, n
15	$bigram(w_{-1}w_0)$, $trigram(w_{-2}w_{-1}w_0)$	n

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/33/15/2363/3089942 by NARASENTAN KAGAKUGIJYUTSU DAIGAKUIN DAIGAKU user on 09 November 2018

In particular, 2-gram and 3-gram features are extracted. Every type of n-grams is instantiated by dividing the probability into 10 even probability ranges. For example, if the probability of the word-bigram 'movement disorders' is in the 5th range, the feature is represented as 'bigram = 5'. In our experiments, language models are pre-trained by Sampo *et al.* (2013) (http://bio.nlplab.org) with the KenLM Language Model Toolkit (http://kheafield.com/code/kenlm). Results show that language model information not only improves the performance of disease named entity normalization, but also increases the performance of disease named entity recognition.

3 Experiments

3.1 Experimental data and evaluation metrics

We evaluated the performance of the joint model on two corpora: the BioCreative V Chemical Disease Relation (BC5CDR) task corpus (Li *et al.*, 2016) and the NCBI Disease corpus (Doğan *et al.*, 2014). The BC5CDR corpus contains 1500 PubMed abstracts, which are equally partitioned into three sections for training, development and test, respectively. A disease mention in each abstract is manually annotated with the concept identifier to which it refers in a controlled vocabulary. The NCBI Disease corpus consists of 793 PubMed abstracts, which are also separated into training (593), development (100) and test (100) subsets. The NCBI Disease corpus is annotated with disease mentions, using concept identifiers from either MeSH or OMIM. Table 3 gives the statistics of the two corpora.

To map disease mentions to MeSH/OMIM concepts (IDs), we used the Comparative Toxicogenomics Database (CTD) MEDIC disease vocabulary (Davis *et al.*, 2015), which consists of 9700 unique diseases described by more than 67 000 terms (including synonyms). We utilized the Stanford CoreNLP toolkit (http://nlp.stanford.edu/software/corenlp.shtml) for NLP preprocessing, such as part-of-speech tagging and tokenization, and the evaluation kit (http://www.biocreative.org/tasks/biocreative-v/track-3-cdr) for evaluating model performances. The results are calculated in term of the standard precision (P), recall (R) and F-score (F) to evaluate the performance, where $F = 2PR/(P + R)$.

Table 3. Overall statistics of BC5CDR and the NCBI

Corpus	Articles	Disease	
		Mention	Concept
BC5CDR corpus	1500	12 864	5818
NCBI corpus	793	6892	1049

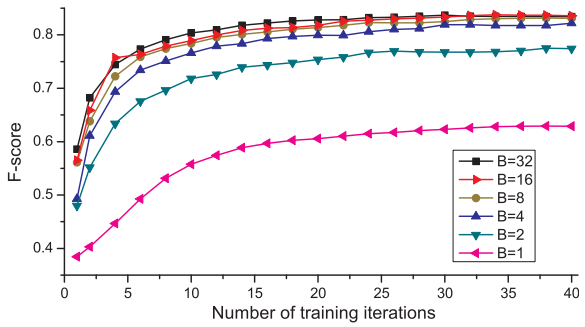

Fig. 1. Influence of beam size on the joint model, using the BC5CDR development set

3.2 Baseline

We built our pipeline baseline by using the disease named entity recognition algorithm (Algorithm 1), trained using the online structured perceptron algorithm (Algorithm 2). The same hyperparameters are used for DER and DEN. The DER model uses the feature templates 1–6, while the DEN model uses the feature templates 7–10 in Table 2.

3.3 Development experiments

3.3.1 Experimental settings

We used the BC5CDR development set to decide the size of the beam and the number of the training iterations. Figure 1 shows different F-score curves on the development set, each with a different beam size B. The X-axis represents the number of iterations, and the Y-axis denotes the F-score for disease entity normalization. With the size of the beam increasing from 1 to 32, the F-score generally increases, while the amount of the increase becomes small when the size of the beam becomes 16. After the 31th iteration, the model with a beam size of 32 does not give better accuracy compared with a beam size of 16, and we therefore chose 16 as the size of the beam for our system. Figure 1 also shows that the F-score increases with an increasing number of training iterations, but the amount of increase becomes small after the 36th iteration. We chose 36 as the number of the iterations to train our system.

On the BC5CDR development set, we further decided the value of K for the top-K most similar concepts. Figure 2 shows the F-score curves for different values of K in the range between 1 and 13. The X-axis represents the value of K, and the Y-axis denotes the F-score. With K increasing from 1 to 10, the F-score generally increases. However, the F-score starts to decrease when the value of K reaches beyond 11. Intuitively, a smaller K value means less possibility for the concept set to contain the gold concept. However, the larger K is, the more noise there can be in the concept set. Thus, we set K as 10 in the remaining experiments.

3.3.2 Development results

Comparison with baseline. Table 4 shows the results on the BC5CDR development set, where R;N denotes the pipeline baseline model, RN$^{\sim}$ denotes the joint model without non-local features, RN denotes the joint model which includes features 1–14 in Table 2. The RN model performs better on DER compared to the pipeline R;N model, which demonstrates the effectiveness of joint model in leveraging DEN information to improve DER accuracies.

Effect of non-local features. Table 4 shows that RN model performs better compared to the RN$^{\sim}$, which demonstrates the effectiveness of joint model using non-local features.

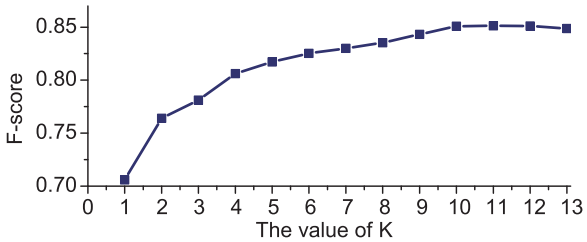

Fig. 2. Influence of the value of K on the BC5CDR development set

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/33/15/2363/3089942 by NARASENTAN KAGAKUGIJYUTSU DAIGAKUIN DAIGAKU user on 09 November 2018

Effect of n-gram language model features. Table 4 shows that the performance increases when using n-gram language model features. In particular, the normalization *F*-score is improved by 3.31%, which indicates that statistical language model plays a useful role.

3.4 Overall performance

In addition to the pipeline baseline, we compared our models with four state-of-the-art methods on the BC5CDR test set, which include the two best results in the original BioCreative V CDR task.

- **Dnorm**: Dnorm (Leaman *et al.*, 2013) is the baseline system from the organizer of the BioCreative V CDR (BC5CDR) task.
- **Avg**: Avg is the average DEN *F*-score of all the results, provided by participants of the BC5CDR task.
- **AuDis**: AuDis (Lee *et al.*, 2016) is a pipeline architecture, which performs automatic CRF-enhanced disease normalization.
- **LeadMine**: LeadMine (Lowe *et al.*, 2015) is also a pipeline framework, which uses CRFs for disease identification and Wikipedia for concept mapping.
- **TaggerOne**: TaggerOne (Leaman and Lu, 2016) is a joint framework for entity recognition and normalization, which uses a semi-Markov structured linear classifier.

Table 5 shows the results, where * corresponds to a *P* value <0.05 compared with our model using pairwise *t*-test. Similar to the development set, our joint model achieves significantly improved results for DER and DEN compared with the pipeline baseline.

Table 4. Recognition and normalization results on the BC5CDR development set

Method	Recognition			Normalization		
	Rec-P	Rec-R	Rec-F	Nor-P	Nor-R	Nor-F
R;N	0.8458	0.6746	0.7506	0.6530	0.5857	0.6175
RN$^{\sim}$	0.8631	0.7122	0.7804	0.8579	0.7642	0.8083
RN	0.8778	0.7479	0.8075	0.8683	0.7714	0.8170
RN+lm	0.8894	0.7865	0.8348	0.8883	0.8149	0.8501

R;N, pipeline model; RN$^{\sim}$, joint model which does not include non-local features; RN, joint model, which includes features 1–14 in Table 2; lm, n-gram language model features; Rec-F, *F*-score of DER; Nor-F, *F*-score of DEN.

Table 5. Recognition and normalization results on the BC5CDR test set

Method	Recognition			Normalization		
	Rec-P	Rec-R	Rec-F	Nor-P	Nor-R	Nor-F
Our joint models						
RN	0.8786	0.7801	0.8264	0.8648	0.8053	0.8339
RN+lm	0.8904	0.7918	0.8382	0.8864	0.8280	0.8562
RN+lm[a]	0.8961	0.8309	0.8623	0.8956	0.8571	0.8761
Other joint models						
TaggerOne[a]	0.8520*	0.8020*	0.8260*	0.8460*	0.8270*	0.8370*
Pipeline models						
Dnorm	–	–	–	0.8115*	0.8013*	0.8064*
Avg	–	–	–	0.7899*	0.7481*	0.7603*
R;N	0.8301*	0.6847*	0.7504*	0.7327*	0.6549*	0.6916*
AuDis[a]	–	–	–	0.896	0.8350*	0.8646
LeadMine	–	–	–	0.8608*	0.8617	0.8612

[a]Systems that are trained on both the training and the development sets.

The *F*-score is boosted from 75.04% to 82.64% for DER and from 69.16% to 83.39% for DEN, respectively. It demonstrates that DEN can enhance DER and DER can also help DEN. Furthermore the *F*-score increases by 1.18% for DER and 2.23% for DEN using n-gram language model features.

As shown in Table 5, AuDis (Lee *et al.*, 2016), which gives the best result on the BC5CDR task, achieves a DEN score of 86.46%. Lee *et al.* (2016) trained their model on both the training and the development sets, and performed several post-processing steps. To normalize disease mentions to specific concepts in an existing repository, they developed a dictionary-lookup method based on the collection of the MEDIC and NCBI disease corpora, the CDR task corpus (training set and development set) and their own extension. LeadMine (Lowe *et al.*, 2015) gives better results (*F*-score 86.12%), which is a dictionary/grammar-based entity recognizer that recognizes and normalizes both chemicals and diseases to Medical Subject Headings (MeSH) IDs. The disease lexicon was obtained from three sources: MeSH, the Disease Ontology and Wikipedia. We also train our models on both the training set and the development set, but use only MEDIC as the disease lexicon. Our joint model gives highly competitive results (*F*-score 86.23% for DER, *F*-score 87.61% for DEN) on this task.

We compare our model with the joint model of Leaman and Lu (2016), namely TaggerOne, on two corpora: the NCBI Disease corpus (Doğan *et al.*, 2014) and the BC5CDR task corpus (Li *et al.*, 2016). Tables 5 and 6 show the results on the two corpora, respectively. On the BC5CDR task, compared with TaggerOner, our transition-based model improves the *F*-score by 3.61% for DER and 3.91% for DEN, respectively. On the NCBI corpus, our method outperforms TaggerOne on DEN *F*-score by 1.92%. Note that we used the same parameter setting tuned on the BC5CDR corpus, and did not fine-tune our model for NCBI.

4 Discussion

In this section, we analyze the results on the BC5CDR test set to show the main reasons that the joint model is better than the pipeline model. We characterize the main errors generated by the joint model. Table 7 shows the number of mention instances for correct/

Table 6. Recognition and normalization results on the NCBI disease curpus

Method	Recognition			Normalization		
	Rec-P	Rec-R	Rec-F	Nor-P	Nor-R	Nor-F
Dnorm	0.8220	0.7750	0.7980	0.8030	0.7630	0.7820
TaggerOne	0.8510	0.8080	0.8290	0.8220	0.7920	0.8070
RN+lm	0.9072	0.7489	0.8205	0.8873	0.7730	0.8262

Table 7. Comparisons between the pipeline and joint models on the BC5CDR test set

Model		Recognition	Normalization
R;N	RN		
Correct	Correct	2970 (67.1%)	2800 (63.3%)
Correct	Wrong	59 (1.3%)	97 (2.2%)
Wrong	Correct	481 (10.9%)	762 (17.2)
Wrong	Wrong	914 (20.7%)	862 (17.3%)

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/33/15/2363/3089942 by NARASENTAN KAGAKUGIJYUTSU DAIGAKUIN DAIGAKU user on 09 November 2018

incorrect recognition and normalization, respectively. For both DER and DEN, the number of instances that were addressed correctly by the RN model but incorrectly by the R;N model is over eight times compared to those addressed by the R;N model correctly but by the RN model incorrectly (59 versus 481; 97 versus 762). Moreover, among the 481 instances that were addressed correctly by RN model but incorrectly by R;N, there were 461 that were correctly normalized by the RN model, but none were correctly normalized by the R;N model. This indicates that the joint model helps to capture DEN information to improve DER.

We compare the recognition capabilities of the two models modelling words with different lengths on the BC5CDR test set. Figure 3 shows the results. The recognition precision of RN is slightly higher than that of R;N for words with length 1, and the precision of RN is significantly higher than that of R;N for words with lengths over 2. This demonstrates the advantage of the joint model in capturing longer range joint context for DER.

4.1 Case study

Table 8 gives two specific examples, where italic strings denote disease mentions. The results of both the R;N and RN models are shown, where *NIL* denotes that there is no corresponding concept in the MEDIC vocabulary for a disease mention, italic strings denote recognized mentions, and italic strings in square brackets give the normalized concept of each mention.

For 1), the pipeline model incorrectly recognized 'failure' as a named entity in the DER phase, resulting in a normalization error in the DEN phase. In contrast, the joint model gave the correct results. This is because the joint model can reduce such error propagation by enabling feedback from successor phases to their predecessors. In particular, in the joint model, failure of normalizing the entity 'failure' in some hypotheses triggers reordering of all the candidate hypotheses in the beam. As a result, the correct prediction 'Acute renal failure' ranks higher after reordering.

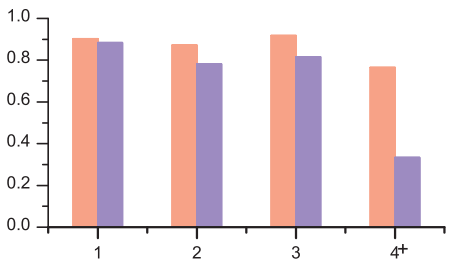

Fig. 3. Recognition precision against word length, where black boxes denote the performances of the RN model, and the gray boxes denote the R;N model

Table 8. Example outputs of the pipeline model and joint model on the BC5CDR test set

Method	Result
R;N	1) Acute renal *failure*[*NIL*] due to rifampicin. 2) Drug-induced *hepatotoxicity*[*Liver Diseases*] is a common cause of acute hepatitis.
RN	1) *Acute renal failure*[*Acute Kidney Injury*] due to rifampicin. 2) Drug-induced *hepatotoxicity*[*Drug-Induced Liver Injury*] is a common cause of acute *hepatitis*[*Drug-Induced Liver Injury*].

1) *Acute renal failure* due to rifampicin. 2) Drug-induced *hepatotoxicity* is a common cause of acute *hepatitis*.

For 2), the pipeline model correctly recognizes the first entity 'hepatotoxicity', but misses the second one 'hepatitis' in the DER phase. Furthermore, the first entity is incorrectly normalized as the concept 'Liver Diseases' in the DEN phase. In contrast, the joint model correctly recognizes and normalizes both entities. This is mainly due to interaction between the recognition of different entities in the sentence, which is introduced by the joint model. Specifically, the first entity, as a feature, can help recognizing and normalizing of the second entity, which in turn helps correct the normalization of the first one. Note that such non-local dependencies are infeasible for the model of Leaman and Lu (2016).

4.2 Error analysis

We randomly selected 100 incorrect outputs of our model RN on the BC5CDR test set for analysis, and classified them into three types, namely false negatives in DEN (DEN FNs), false positives in DEN but true positives in DER (DEN FPs), and false positives in both DEN and DER (DEN&DER FPs). Figure 4 gives the distribution.

DEN&DER FPs account for 48% of the errors, which fall in two sub-categories. One sub-category is a boundary error, which involves a true named entity mention whose boundaries are not identified correctly. For example, in (1) below, the recognized entity is 'cerebral ischemic stroke', which is longer than the correct one 'ischemic stroke'. In (2) and (3), the recognized entities are 'personality disorder' and 'auditory toxicity', which are shorter than the correct versions: 'antisocial personality disorder' and 'Ocular and auditory toxicity', respectively. The ratio of these errors against the DEN&DER FPs is 63.3%. A possible reason for such errors is that the current features fail to capture the structural difference in the entities, which may require more detailed lexical or syntactic knowledge.

The other sub-category involves those words and phrases that look like named entities according to their context, but are noise, For example, 'psychiatric' in (4) is not a DE. The main reason can be that the modelling of the context by the current model fails to rule out such noise. To address such mistakes, the model may need to capture details of the context or leverage different negative examples.

1. A man presented with cerebral *ischemic stroke* after intravenous heroin.
2. *Antisocial personality disorder* was the only factors associated with psychosis.
3. *Ocular and auditory toxicity* in hemodialyzed patients receiving desferrioxamine.
4. Antidepressants have previously been associated with *paranoid* reactions in psychiatric patients.

DEN FNs account for 38% of the errors. Such errors are generally related to abbreviated disease names (e.g. 'MDP', 'CAD'), ambiguous adjectives (e.g. 'chill', 'flushing') and nouns (e.g. 'weakness', 'drowsiness'). A possible reason is that such ambiguous

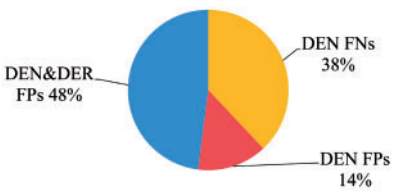

Fig. 4. Error distribution

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/33/15/2363/3089942 by NARASENTAN KAGAKUGIJYUTSU DAIGAKUIN DAIGAKU user on 09 November 2018

named entities generally correspond to multiple concept candidates, which may lead to recall loss in the DEN phase.

DEN FPs account for 14% of the errors, which are mainly caused by confusion with similar hypernyms or hyponyms of the correct concepts in the DEN phase. For example, the entity mention 'liver damage' was normalized as 'Liver Diseases', while the correct concept is 'Drug-Induced Liver Injury'. The entity mention 'brain damage' was normalized as 'Brain Damage, Chronic', while the correct concept is 'Brain Injuries'.

5 Related work

Disease Named Entity Recognition. DER has been widely studied in the literature (Chowdhury *et al.*, 2010; Leaman *et al.*, 2008; Sahu and Anand, 2016). Most previous studies use standard sequence labeling models with Viterbi search, which gives state-of-the-art performances (Chowdhury *et al.*, 2010; Leaman *et al.*, 2008; Lee *et al.*, 2016). For example, Chowdhury *et al.* (2010) present a conditional random fields (CRFs) classifier using a feature set tailored for DER, achieving the best performance approaches on the Arizona Disease Corpus.

Disease Named Entity Normalization. DEN has been highlighted as subtasks of SemEval-2014 (Pradhan *et al.*, 2014) and BioCreative V (Wei *et al.*, 2015a). A variety of approaches, including dictionary-based, machine learning and heuristic rules, are described for the systems participating in these tasks (Ghiasvand and Kate, 2014; Kang *et al.*, 2013; Kate, 2016; Leaman *et al.*, 2013; Lee *et al.*, 2016). Most studies assume that the named entities are pre-detected by a separated DER model, and focus on developing techniques to improve the normalization accuracy. Kang *et al.* (2013) use rule-based natural language processing to improve disease normalization in biomedical text. Leaman *et al.* (2013) propose the DNorm system for the NCBI task, which is based on CRFs. Lee *et al.* (2016) leverage CRFs for DER and a dictionary-lookup method for DEN. Such methods process DER and DEN as a two separate step pipeline. The main limitations of such methods include error propagation from DER to DEN and lack of feedback from DEN to DER. In contrast, we investigate a joint model to address the problems.

Joint DER and DEN. Joint models have been studied for many NLP tasks and recently for DER and DEN. For example, semi-CRF has been used for joint entity recognition and disambiguation (Luo *et al.*, 2015), where Viterbi decoding is used for assigning POS tags and normalizing non-standard tokens simultaneously (Li and Liu, 2015). Semi-Markov models are also used for joint disease entity recognition and normalization. Leaman and Lu (2016) leverage a joint scoring function for DER and DEN. The joint models use exact inference with dynamic programming, which forbids non-local features. In this paper, we use a joint model based on a transition-based frameworks. Our model leverages beam-search to address the search challenge, but also employs non-local combined features of the two tasks.

Transition-based Models. Transition-based models transform the output construction process into a state-transition sequence. Stating from a start state, the outputs are constructed incrementally, by applying a sequence of transition actions. Zhang and Clark (2011) establish a framework for modelling sequences of transition actions as a whole, using linear-time beam-search decoding and learning-to-search (Collins and Roark, 2004) for fixing search errors. The advantages of the framework as compared with dynamic programming models include low runtime cost and the capability of integrating arbitrary non-local feature thanks to heuristic-decoding. It has been applied to joint segmentation and POS-tagging (Qian *et al.*, 2015; Stern *et al.*, 2012; Zhang and Clark, 2008), dependency parsing (Andor *et al.*, 2016; SøGaard and Haulrich, 2011; Zhang and Nivre, 2011; –>Zhou *et al.*, 2015), constituent parsing (Watanabe and Sumita, 2015; Zhu *et al.*, 2013) and joint lexical and syntactic systems (–>Bohnet and Nivre, 2012; Constant and Nivre, 2016; Hatori *et al.*, 2011). We investigate the effectiveness of this model, and in particular non-local feature for joint disease entity recognition and normalization. Similar to Qian *et al.* (2015) and Lyu *et al.* (2016), our method can be regarded as the investigation of a transition-based joint chunking system for a new task, with designing of a set of task-specific actions and features.

6 Conclusion

We proposed a transition-based model for joint disease entity recognition and normalization, based on the transition-based structured prediction framework of Zhang and Clark (2011) using structured perceptron with early-update training and beam-search decoding. Results on the BC5CDR corpus and the NCBI Disease corpus demonstrated that our joint model improves the performance of disease entity recognition and normalization significantly compared with its pipeline baseline. Moreover, we found that language model features are very helpful for this task. Our model gives competitive results to the best method in the literature thanks to the use of non-local features in the transition-based method.

Funding

This work was supported by the National Natural Science Foundation of China [grant number 61373108], Major Projects of the National Social Science Foundation of China [grant number 11&ZD189], Humanities and Social Science Foundation of Ministry of Education of China [grant number 16YJCZH004] and Educational Commission of Henan Province, China [grant number 17A520050]

Conflict of Interest: none declared.

References

Andor,D. *et al.* (2016) Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the ACL (Volume 1: Long Papers)*, pp. 2442–2452.

Bohnet,B. and Nivre,J. (2012) A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the EMNLP-CONLL*, pp. 1455–1465.

Chowdhury,M. *et al.* (2010) Disease mention recognition with specific features. In *Proceedings of the 2010 workshop on biomedical NLP*, pp. 83–90.

Collins,M. (2002) Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of ACL*, pp. 1–8.

Collins,M. and Roark,B. (2004) Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, p. 111. ACL.

Constant,M. and Nivre,J. (2016) A transition-based system for joint lexical and syntactic analysis. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 161–171, Berlin, Germany.

Davis,A.P. *et al.* (2015) The comparative toxicogenomics database's 10th year anniversary: update 2015. *Nucleic Acids Res.*, **43**, 914–920.

Doddington,G. (2002) Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pp. 138–145.

Doğan,R.I. *et al.* (2014) Ncbi disease corpus: a resource for disease name recognition and concept normalization. *J. Biomed. Informatics*, **47**, 1–10.

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/33/15/2363/3089942 by NARASENTAN KAGAKUGIJYUTSU DAIGAKUIN DAIGAKU user on 09 November 2018

Ghiasvand,O. and Kate,R.J. (2014) Uwm: disorder mention extraction from clinical text using crfs and normalization using learned edit distance patterns. In *Proceeding of the Eight International Workshop on Semantic Evaluations*, pp. 828–832.

Hatori,J. *et al*. (2011) Incremental joint pos tagging and dependency parsing in chinese. In *In IJCNLP*, pp. 1216–1224.

Joseph,S. *et al*. (2015) Pcoskb: a knowledgebase on genes, diseases, ontology terms and biochemical pathways associated with polycystic ovary syndrome. *Nucleic Acids Res*., **44**, D1032–D1035.

Kaji,N. and Kitsuregawa,M. (2014) Accurate word segmentation and pos tagging for japanese microblogs: corpus annotation and joint modeling with lexical normalization. In *Proceedings of EMNLP*, pp. 99–109.

Kang,N. *et al*. (2013) Using rule-based natural language processing to improve disease normalization in biomedical text. *J. Am. Med. Informatics Assoc*., **20**, 876–881.

Kate,R.J. (2016) Normalizing clinical terms using learned edit distance patterns. *J. Am. Med. Informatics Assoc*., **23**, 380–386.

Khan,I.Y. *et al*. (2013) Importance of artificial neural network in medical diagnosis disease like acute nephritis disease and heart disease. *Int. J. Eng. Sci. Innovative Technol*., **2**, 210–217.

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

Leaman,R. *et al*. (2008) Banner: an executable survey of advances in biomedical named entity recognition. In *Pacific Symposium on Biocomputing*, pp. 652–663.

Leaman,R. *et al*. (2013) Dnorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, **29**, 2909–2917.

Leaman,R. *et al*. (2015) Challenges in clinical natural language processing for automated disorder normalization. *J. Biomed. Informatics*, **57**, 28–37.

Lee,H.-C. *et al*. (2016) Audis: an automatic crf-enhanced disease normalization in biomedical text. *Database*, **baw091**, 1–11.

Levenshtein,V. (1966) Binary codes capable of correcting deletions, insertions and reversals. *Soviet Phys. Doklady*, **10**, 707–710.

Li,C. and Liu,Y. (2015) Joint pos tagging and text normalization for informal text. *IJCAI*, 1263–1269.

Li,J. *et al*. (2016) Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, **baw068**, 1–10.

Lowe,D.M. *et al*. (2015) Leadmine: disease identification and concept mapping using wikipedia. In *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*, pp. 240–246.

Luo,G. *et al*. (2015) Joint named entity recognition and disambiguation. In *Proc. EMNLP*., pp. 879–880.

Lyu,C. *et al*. (2016) Joint word segmentation, pos-tagging and syntactic chunking. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 3007–3014.

Pradhan,S. *et al*. (2014) Semeval-2014 task 7: analysis of clinical text. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. 2014, **199**: 54–62.

Qian,T. *et al*. (2015) A transition-based model for joint segmentation, pos-tagging and normalization. In *Proceedings of the 2015 Conference on EMNLP*, pp. 1837–1846.

Sahu,S. and Anand,A. (2016) Recurrent neural network models for disease name recognition using domain invariant features. In *Proceedings of ACL(1)*, pp. 2216–2225.

Sampo,P. *et al*. (2013) Distributional semantics resources for biomedical text processing. In *Proceedings of Languages in Biology and Medicine*, pp. 1–4.

SøGaard,A. and Haulrich,M. (2011) Sentence-level instance-weighting for graph-based and transition-based dependency parsing. In *Proceedings of the 12th International Conference on Parsing Technologies*, pp. 43–47.

Stern,R. *et al*. (2012) A joint named entity recognition and entity linking system. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, chapter, pp. 52–60. ACL.

Watanabe,T. and Sumita,E. (2015) Transition-based neural constituent parsing. In *Proceedings of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on NLP(Volume 1: Long Papers)*, pp. 1169–1179.

Wei,C.-H. *et al*. (2015a) Overview of the biocreative v chemical disease relation (cdr) task. In *Proceedings of BioCreative Workshop, Sevilla, Spain*, pp. 154–166.

Wei,Q. *et al*. (2015b) Disease named entity recognition and normalization using conditional random fields and levenshtein distance. In *Proceedings of BioCreative Workshop, Sevilla, Spain*, pp. 327–334.

Zhang,Y. and Clark,S. (2008) Joint word segmentation and pos tagging using a single perceptron. In *ACL*, pp. 888–896.

Zhang,Y. and Clark,S. (2011) Syntactic processing using the generalized perceptron and beam search. *Comput. Linguist*., **37**, 1–47.

Zhang,Y. and Nivre,J. (2011) Transition-based dependency parsing with rich non-local features. In *Proceedings of ACL*, pp. 188–193.

Zhou,H. *et al*. (2015) A neural probabilistic structured-prediction model for transition-based dependency parsing. In *Proceedings of ACL*, pp. 1213–1222.

Zhu,M. *et al*. (2013) Fast and accurate shift-reduce constituent parsing. In *Proceedings of the 51st Annual Meeting of the ACL (Volume 1: Long Papers)*, pp. 434–443.

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/33/15/2363/3089942 by NARASENTAN KAGAKUGIJYUTSU DAIGAKUIN DAIGAKU user on 09 November 2018