Bioinformatics, 34(20), 2018, 3539–3546
doi: 10.1093/bioinformatics/bty356
Advance Access Publication Date: 30 April 2018
Original Paper

Data and text mining

D3NER: biomedical named entity recognition using CRF-biLSTM improved with fine-tuned embeddings of various linguistic information

Thanh Hai Dang[1,*,†], Hoang-Quynh Le[2,†], Trang M. Nguyen[1,‡] and Sinh T. Vu[1,‡]

[1]Department of Computational Science and Engineering, Faculty of Information Technology, VNU University of Engineering and Technology, Hanoi 100000, Vietnam and [2]Knowledge Technology Laboratory (KTLab), Faculty of Information Technology, VNU University of Engineering and Technology, Hanoi 100000, Vietnam

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.
‡The authors wish it to be known that these authors contributed equally.

Associate Editor: Jonathan Wren

Received on January 1, 2018; revised on April 22, 2018; editorial decision on April 25, 2018; accepted on April 27, 2018

Abstract

Motivation: Recognition of biomedical named entities in the textual literature is a highly challenging research topic with great interest, playing as the prerequisite for extracting huge amount of high-valued biomedical knowledge deposited in unstructured text and transforming them into well-structured formats. Long Short-Term Memory (LSTM) networks have recently been employed in various biomedical named entity recognition (NER) models with great success. They, however, often did not take advantages of all useful linguistic information and still have many aspects to be further improved for better performance.

Results: We propose D3NER, a novel biomedical named entity recognition (NER) model using conditional random fields and bidirectional long short-term memory improved with fine-tuned embeddings of various linguistic information. D3NER is thoroughly compared with seven very recent state-of-the-art NER models, of which two are even joint models with named entity normalization (NEN), which was proven to bring performance improvements to NER. Experimental results on benchmark datasets, i.e. the BioCreative V Chemical Disease Relation (BC5 CDR), the NCBI Disease and the FSU-PRGE gene/protein corpus, demonstrate the out-performance and stability of D3NER over all compared models for chemical, gene/protein NER and over all models (without NEN jointed, as D3NER) for disease NER, in almost all cases. On the BC5 CDR corpus, D3NER achieves *F*1 of 93.14 and 84.68% for the chemical and disease NER, respectively; while on the NCBI Disease corpus, its *F*1 for the disease NER is 84.41%. Its *F*1 for the gene/protein NER on FSU-PRGE is 87.62%.

Availability and implementation: Data and source code are available at: https://github.com/aidantee/D3NER.

Contact: hai.dang@vnu.edu.vn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Named Entity Recognition (NER) in textual documents is an essential phase for more complex downstream text mining analyses, being a difficult and challenging topic of interest among research community for a long time (Kim *et al.*, 2009; Krallinger *et al.*, 2015; Wei *et al.*, 2016a). In the domain of bio-medicine, entities can be chemicals, diseases, anatomies, pathways and genes/proteins, etc. which are named in bio-medical literature, which has been growing at an unprecedented speed (PubMed is a typical example). Resolving biomedical NER successfully is prerequisite for extracting huge amount of biomedical knowledge deposited in the unstructured textual literature, transforming them into well-structured formats. Biomedical entities have their own diversities and characteristics of being named, causing the recognition of them in the literature more difficult (Leaman and Gonzalez, 2008; Zhou *et al.*, 2004).

Traditionally, to perform well and efficiently, NER models require a set of informative features (i.e. linguistic patterns) that are well engineered and carefully selected, heuristically based on domain knowledge (Campos *et al.*, 2012). Typical examples of such models for biomedical domain include DNorm (Leaman *et al.*, 2013), TmChem (Leaman *et al.*, 2015), TaggerOne (Leaman and Lu, 2016), UET-CAM (Le *et al.*, 2015, 2016) and the model of Lou *et al.* (2017). Feature engineering, however, is very time-consuming, very often yields incomplete non-satisfactory sets. Moreover, resulting feature sets are both domain and model-specific.

In the past few years, the advent of deep neural networks with the capability of automatic feature engineering even from noisy data has leveraged the development of NER models (Jagannatha and Yu, 2016; Lample *et al.*, 2016). Very recently, an advanced deep neural network type called bidirectional Long Short-Term Memory (biLSTM) has increasingly been employed for biomedical NER, yielding state-of-the-art performance at the time of their publication, such as Limsopatham and Collier (2016), Ma and Hovy (2016), Wei *et al.* (2016b), Luo *et al.* (2018) and Habibi *et al.* (2017). Within these models, the biLSTM is used to learn optimal contextual vector representations of every linguistic unit (i.e. word/token) in a sentence to be taken as input to a state-of-the-art advanced sequence labeling model called the Conditional Random Fields (CRFs) (Lafferty *et al.*, 2001). Linguistic units are initialized with low dimensional continuous vector representations (embeddings) that are pretrained from extremely huge amount of unlabeled text. Some models also take as input character-level embeddings of words to their biLSTM models, bringing the further out-performance to their biomedical NER models (Habibi *et al.*, 2017; Luo *et al.*, 2018; Verwimp *et al.*, 2017). Apart from such two embeddings, other important features related to linguistic information (such as POS and chunking) and to domain useful resources/dictionaries have also been demonstrated to improve the biomedical NER performance (Luo *et al.*, 2018). These features, however, still need to be manually designed to take their effect.

Biomedical literature uses a lot of abbreviations, of which many do not follow a standard convention and are only used locally within the scope of authors' articles. For example, the BC 5 CDR and NCBI corpora contain more than 6000 and 4000 abbreviations for chemicals and diseases, respectively. This ambiguous abbreviation usage causes some system's errors. It, however, has not been addressed thoroughly in other existing NER systems. Leaman *et al.* (2015), Leaman and Lu (2016) and Wei *et al.* (2016b) resolve this issue by merely replacing abbreviations with their full form. Nevertheless, because the full form of an abbreviation is often longer and much more complex than the abbreviation itself, such the replacement introduces more syntactical complexity to sentences, possibly causing more errors when the model has to label separated tokens (within the full form) rather than the syntactically concrete abbreviation as a whole.

To this regards, this paper presents D3NER, a novel biomedical named entities recognition model using CRFs and a well-designed biLSTM network architect improved with embeddings of various informative linguistic information. Apart from pretrained word/token embeddings and character-level word embeddings, D3NER incorporates abbreviation embeddings and Part-of-speech (POS) embeddings. D3NER is validated on three benchmark datasets, i.e. the BioCreative V Chemical/Disease relation corpus (Li *et al.*, 2015), the NCBI Disease corpus (Doğan *et al.*, 2014) and FSU-PRGE (Hahn *et al.*, 2010), which were used by other existing biomedical NER models. Experimental results on such three demonstrate the out-performance and stability of D3NER over other state-of-the-art related models.

2 Materials and methods

2.1 Datasets

We evaluate D3NER on three benchmark corpora of disease, chemical and gene/protein annotations: the BioCreative V Chemical Disease Relation (BC5 CDR) corpus (Li *et al.*, 2015), the NCBI Disease corpus (Doğan *et al.*, 2014) (see Table 1) and FSU-PRGE (Hahn *et al.*, 2010). These three were used for evaluating various state-of-the-art biomedical NER models. In our study, D3NER is fine-tuned using training and development sets while the test set is kept totally untouched for reporting the system performance.

We found that the average chemical mention length in the BC5 CDR corpus is quite short, of 1.19 tokens. Short chemical entities, however, are often abbreviations, which are notoriously ambiguous. The longest chemical mention has 22 tokens. The average disease mention length in the BC5 corpus is longer, of 1.62 tokens. Disease mentions, nevertheless, seem to be less complex, with the longest term of 15 tokens. The average disease mention length in the NCBI Disease corpus is 2.051 tokens, with the longest being 13 tokens.

Regarding gene/protein NER, D3NER is evaluated on FSU-PRGE (Hahn *et al.*, 2010), which is also used in Habibi *et al.* (2017) to which D3NER is compared. This corpus is the largest one among all of gene/protein entities used in Habibi *et al.* (2017). FSU-PRGE contains 3309 abstracts with 59365 mentions of 16 683 unique entities.

2.2 Data pre-processing

The spaCy (Spacy: Industrial-Strength Natural Language Processing in Python: https://spacy.io) open source library is used for segmentation, tokenization and POS tagging.

We also applied two additional pre-processing rules: (i) Removing hyphen (–) as it often brings bad effects to the tokenization and makes

Table 1. Information about the corpora used for training and evaluating D3NER

Corpus	Subset	Articles	Disease		Chemical	
			Mentions	Uniques	Mentions	Uniques
BC5 CDR	Training	500	4182	1965	5203	1038
	Development	500	4244	1865	5347	1012
	Test	500	4424	1988	5385	1066
NCBI Disease	Training	593	5145	1710		
	Development	100	787	368		
	Test	100	960	427		

the model perform inconsistently. To this end, we replace each hyphen with a space. (ii) Normalizing numbers by replacing all numbers with zeros because numbers are highly specific and prone to over-fitting.

2.3 Model architecture

D3NER comprises of four layers, namely TPAC embeddings, context representing biLSTM, project and NER layer, being structured in an architect as depicted in Figure 1.

2.3.1 TPAC embeddings layer

The embedding layer takes as input a pre-processed sentence of n tokens $t_1 t_2 \dots t_n$ each coupled with a POS tag, and output an embedding $\vec{e}_i$ for each token t_i $(1 \leq i \leq n)$, which encodes several important linguistic information rather than only the token itself. To this end, each token is presented with a continuous vector being concatenation of the embedding of the Token itself, its POS, information about its Abbreviation status and Character (TPAC) (see Equation (1)). The overall architect of this layer is described in Figure 2. Except for the token and abbreviation embeddings, the two others are fine-tuned during the D3NER training by back-propagating gradients as the D3NER layers are stacked on top of each others, allowing the NER loss during the supervised training be used to make update to these embeddings.

$$\vec{e}_i = \vec{e}_i^{t} \oplus \vec{e}_i^{p} \oplus \vec{e}_i^{a} \oplus \vec{e}_i^{c} \tag{1}$$

Token-level embedding (*TE*) $\vec{e}^{t}$ captures semantic (dis)similarities between tokens that are not visible from their morphological surface (e.g. '*Grippe*' and '*Influenza*'). *TEs* can be pre-trained using a given word embedding model on an extremely huge amount of unlabeled text. In this study, *TEs* are obtained from a freely online corpus of pre-trained word embeddings of 200 dimensions provided by Pyysalo *et al.* (2013), in which they employed the word2vec skip-gram model (Mikolov *et al.*, 2013) on PubMed abstracts and PMC full texts (6 millions distinct words).

POS embedding (*PE*) $\vec{e}^{p}$ captures (dis)similarities between grammatical properties of words and their syntactic structural roles within a sentence. Words that bear the same POS tag in a sentence often exhibit similar roles within it. On the other hand, words with the same morphological surface but being marked with different POS may have different properties. We use 56 POS tags in the OntoNotes version 5.0 of the Penn Treebank tag set, which is used by other existing POS taggers. In D3NER, *PE* are randomly initialized according to the Glorot uniform (Bengio and Glorot, 2010).

Abbreviation embedding (*AE*) $\vec{e}^{a}$ encodes whether a token is an abbreviation or not and further the information about maximum similarities between its full mention form with names of every entity type (e.g. chemical, disease or gene/protein, etc.) in biomedical lexicon databases [e.g. FSU-PRGE (training set only) for genes/proteins, MeSH for chemicals and diseases in our study].

All local abbreviations in an abstract are first identified using Ab3P (Sohn *et al.*, 2008). Then, the character-level $n-$gram TF-IDF vector for each abbreviation's full form and every concept name in MeSH and FSU-PRGE (training set) are generated for measuring the pair-wise cosine similarity scores. Initially, $\vec{e}^{a}$ is a 3-dimensional vector (a_d, a_c, a_{gp}), in which, if a word is an abbreviation then a_d, a_c and a_{gp} are the maximum similarity between its full form with the MeSH disease, chemical and FSU-PRGE gene/protein name, respectively. To scale this vector to be suitable with other embeddings, we put it through a fully connected layer to generate the final 5-dimensional vector $\vec{e}^{a}$.

Character-level embedding (*CE*) $\vec{e}^{c}$ represents a token's meaning in sense of its morphological surface (e.g. '*effective*' and '*effectiveness*'). A compositional *CE* model, which is similar to the character to word embedding model by Ling *et al.* (2015), is built upon another biLSTM (called CE-BiLSTM). We denote the character set by $\mathbb{C}$, containing 76 entries for 26 letters in uppercase and lowercase forms, punctuation and numbers. Each character $c_j \in \mathbb{C}$ is represented with a vector $\vec{ce}_j$ that is initialized by looking up a *C* lookup table, which was randomly constructed according to the Glorot uniform (Bengio and Glorot, 2010).

The CE-biLSTM takes as input a single token t_i, which is an ordered sequence of m characters $(c_1 c_2 \dots c_m)$, with $c_j \in \mathbb{C}$, and outputs the character-level embedding $\vec{e}_i^{c}$ of token t_i by concatenating the last output's vector of the forward and backward LSTMs, namely $\overrightarrow{fwe}_i^{c}$ and $\overleftarrow{bwe}_i^{c}$ (Equation (2)).

$$\vec{e}_i^{c} = \overrightarrow{fwe}_i^{c} \oplus \overleftarrow{bwe}_i^{c} \tag{2}$$

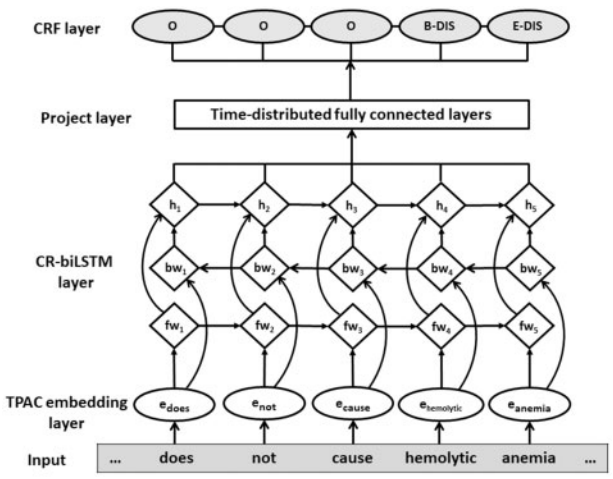

Fig. 1. The D3NER layer architecture. Example comes from the BioCreative V Chemical Disease Relation task corpus (PMC3425586)

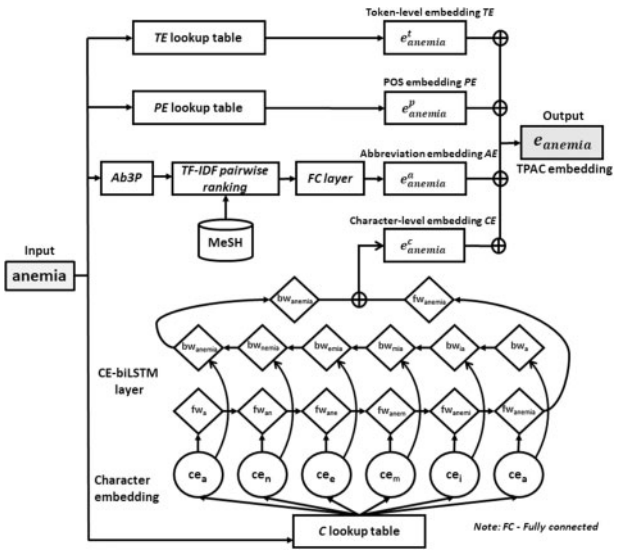

Fig. 2. The TPAC embedding architecture of D3NER. Token 'anemia' is given as an example

2.3.2 Context representating BiLSTM layer

This layer takes as input a sequence of embedding vectors $\vec{e_1}, \vec{e_2}, \ldots, \vec{e_n}$ produced by the TPAC layer for a sentence of n tokens $t_1 t_2 \ldots t_n$. It is for modeling the context information of each token/word within the sentence. To distinguish it from the other biLSTM in the TPAC layer, we call it the 'context representating BiLSTM' layer (CR-biLSTM).

For each token t_i, the forward LSTM processes from the beginning of the sentence to compute the forward context vector representation $\overrightarrow{fw_i}$; and vice versa, the backward LSTM, which processes from the end of the sentence, computing the backward context vector representation $\overrightarrow{bw_i}$. Concatenation of the forward and backward representations gives the final embedding $\vec{h_i}$ of token t_i (Equation (3)).

$$\vec{h_i} = \overrightarrow{fw_i} \oplus \overrightarrow{bw_i} \tag{3}$$

2.3.3 Project layer

The project layer aims to encode the output of the CR-biLSTM layer into a sequence of d-dimensional vectors which is compatible with the number of labels pre-defined in the CRF layer. It consists of two time-distributed fully connected layers. The first is used to encode the CR-biLSTM output of 550-dimensional vectors into 275-dimensional vectors while the second continues to decrease the dimensions from 275 to 9 (for chemical/disease NER) or 5 (for gene/protein NER), which correspond to the number of pre-defined NER labels (see Section 2.3.4). Such two can learn features at various levels of abstraction and are experimentally demonstrated to be much better at generalizing.

Before each fully connected layer, we apply the batch normalization technique (Ioffe and Szegedy, 2015), which helps improving the performance noticeably. Batch normalization is quite effective at accelerating and enhancing the training of deep models. It transforms the output of a hidden layer into the standard normal distribution (zero mean and unit variance) before activation.

2.3.4 NER layer

The on top layer uses linear chain Conditional Random Fields (CRFs) (Lafferty *et al.*, 2001) to label the whole sentence by performing the Viterbi algorithm (LeCun *et al.*, 1998). We adopt the expressive variant of *IOB* tagging scheme called *IOBES*, in which *I* stands for Inside, *O* for Outside, *B* for Beginning, *E* for Ending and *S* for Singleton. *IOBES* tagging scheme was shown to improve labeling models' performance marginally (Ratinov and Roth, 2009) and has been used in several NER studies (Lample *et al.*, 2016; Wei *et al.*, 2016b). As D3NER focuses on recognizing diseases, chemicals and genes/proteins, we thus extend the *IOBES* scheme into 13 labels: one set of *I*−, *B*−, *E*− and *S*− for diseases, two others for chemicals, genes/proteins and the label *O* is for tokens that are neither diseases, chemicals nor genes/proteins.

2.3.5 Hyper parameters and model training

We implement the neural networks using the TensorFlow library (an Open Source Software Library for Machine Intelligence: https://www.tensorflow.org). Both LSTMs in D3NER employ the RMSProp optimizer with the learning rate and the momentum value being 0.0005 and 0.9, respectively, which are chosen based on experiments on the development set. They both use the Glorot random uniform based initializer. The *tanh* activation function is applied to the output of all LSTM units. Batch-padding is applied to pad the length of all tokens to be equal to the maximum length in each batch. The mini batch training size is set to 128. Pretrained token embeddings (*TE*) have 200 dimensions while each character is initialized with a 50-dimensional vector. These numbers for *PE*, *CE*, *AE* are 25, 100 and 5, respectively. We set 100 and 275 as the dimension numbers for the hidden states of CE-biLSTM and CR-biLSTM, respectively.

To avoid overfitting, we apply dropout (Srivastava *et al.*, 2014), with 0.5 and 0.15 respectively for the final hidden layer of CE-biLSTM and CR-biLSTM, and 0.5 for the first fully connected layer of the project layer. Early stopping (Caruana *et al.*, 2001) is applied based on the D3NER performance on the validation sets (the model often stops at around 27, 13 and 25 epochs on the BC5 CDR, the NCBI Disease and the FSU-PRGE corpus, respectively).

2.4 Long Short Term Memory (LSTM) network

A Long Short Term Memory network (LSTM for short), originally introduced by Hochreiter and Schmidhuber (1997), is a specific variant of Recurrent Neural Networks (RNNs) that, unlike RNNs, is not suffered from the problem of vanishing and exploding gradients (Hochreiter and Schmidhuber, 1997). LTSM therefore can model long-term dependencies within sequences. This advancement is a result of equipping LSTMs with the memory capability through a memory-cell c_t with an adaptive gating mechanism. A standard LSTM consists of three specific gates (i.e. forget gate f_t, input gate i_t and output gate o_t) to control the degree that the LSTM previous state is kept and the extracted features of the current data input are memorized. The LSTM hidden state at time t is calculated using equations as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{5}$$

$$c_t = i_t \otimes \tanh(W_c x_t + U_c h_{t-1} + b_c) + f_t \otimes c_{t-1} \tag{6}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{7}$$

$$h_t = o_t \otimes \tanh(c_t) \tag{8}$$

In the Equation (4) to Equation (8), σ denotes the sigmoid function, and $\otimes$ denotes the element-wise multiplication.

2.5 Conditional Random Fields

Conditional Random Fields (CRFs) (Lafferty *et al.*, 2001) is a discriminative undirected probabilistic graphical model, which bring together all advantages of the most two well-known typical methods for sequence labeling, i.e. Hidden Markov Models (HMMs) (Rabiner and Juang, 1986) and Maximum Entropy Markov Models (MEMMs) (McCallum *et al.*, 2000).

Given a training dataset $D = \{(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^N, \mathbf{y}^N)\}$ of N data observation sequences $\mathbf{x}^i$ and corresponding label sequences $\mathbf{y}^i$, CRFs are learned to maximize the log-likelihood of the conditional probability of the label sequences given the observation sequences, that is:

$$L = \sum_{i=1}^{N} \log\left(p(\mathbf{y}^i|\mathbf{x}^i)\right) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2} \tag{9}$$

3 Model evaluation

D3NER performance for disease and chemical NER is reported on the test sets (see Table 1), as done in other existing state-of-the-art NER models to which D3NER is compared. For gene/protein NER, the performance is evaluated using 4-fold cross-validation. A correct match is recorded when the text span predicted as an entity by

D3NER entirely overlaps with one of the same entity type in the golden standard corpus. Three well-known evaluation metrics are used to score the model performance, including F1 score, precision (P), recall (R). Mehryary *et al.* (2016) noted that using random initialization can cause a significant impact on the model performance. We, therefore, run the model for 20 times with different random initialization for each time and the averages of resulting aforementioned performance scores are reported for evaluation and comparison.

3.1 Compared existing models

D3NER is compared with seven very recent state-of-the-art biomedical NER models, including:

- **Dnorm** (Leaman *et al.*, 2013) and **tmChem** (Leaman *et al.*, 2015) for disease and chemical NER, respectively. They are both based on CRFs with carefully-engineered rich feature sets. Dnorm performance was ranked first in the ShARe/CLEF eHealth 2013 shared task (Pradhan *et al.*, 2015) while tmChem in the BioCreative IV CHEMDNER task (Krallinger *et al.*, 2015).
- **Wei *et al.*, (2016b)**'s and **Habibi *et al.*, (2017)'s models**, both based on biLSTMs and CRFs. The former is for only disease NER by first developing two separate models (i.e. one based on CRFs with a rich feature set and another based on a bidirectional recurrent neural network) and then combining the outputs of these two models as input to a support vector machine (SVM). The latter is for recognition of six different biomedical named entities, including diseases and chemicals. It stacks a CRFs layer on top of two biLSTM networks with character-level and pretrained word embeddings to form a single model that can be trained by backpropagation. This model has been experimentally demonstrated to out-perform the baseline model (i.e. CRFs with a generic NER feature set plus pretrained word embeddings) and the best entity-specific NER tool (i.e. Dnorm for diseases and tmChem for chemicals).
- **Att-ChemdNER** (Luo *et al.*, 2018) for chemical NER at the document level. It is a CRF-biLSTM based model that is additionally equipped with an attention layer. Apart from character-level and pretrained word embeddings, Att-ChemdNER also incorporates two kinds of traditional manually designed features (i.e. POS and chunking embeddings as linguistic features and chemical dictionary embedding as domain resource feature). The model has been experimentally shown to yield performance better than TaggerOne (Leaman and Lu, 2016) and tmChem (Leaman *et al.*, 2015).
- **TaggerOne** (Leaman and Lu, 2016) and **Lou *et al.*, (2017)'s model**. These two models perform NER in a joint manner with named entity normalization (NEN), in which the feedback from the NEN model can be used to reduce NER errors, boosting its performance. The former is for both disease and chemical NER while the latter is only for diseases. TaggerOne employs a semi-Markov structured linear classifier with a rich linguistic feature set for NER, jointly learned with a supervised semantic indexing model for NEN. The model of Lou *et al.* (2017) is based on a finite state machine, in which, given an input sentence, its output (i.e. a sequence of recognized and normalized disease entities) is incrementally constructed by a sequence of transition actions. Like TaggerOne, it also requires a subset of informative features manually designed. Lou *et al.* (2017) have demonstrated the out-performance of their model for disease NER over Dnorm, LeadMine (Lowe *et al.*, 2015) and TaggerOne.

4 Results and discussion

4.1 D3NER performance and comparisons

Experimental results on two benchmark datasets show that D3NER could yield excellent performance for chemical NER and very good for disease NER. The chemical NER performance of D3NER has *F*1 of 93.14% ($std = 0.32\%$, $95\%CI = [92.89, 93.29]$) when evaluating on the BC5 CDR corpus. The performance of D3NER for disease NER has *F*1 of 84.68% ($std = 0.33\%$, $95\%CI = [84.52, 84.84]$) and 84.41% (std = 0.65%, $95\%CI = [84.09, 84.72]$) when assessing on the BC5 CRD corpus and NCBI Disease corpus, respectively. For gene/protein NER, these numbers are 87.62% (0.45%, [87.52%, 87.72%]) on FSU-PRGE corpus (see Supplementary Table S1). We note that D3NER performs very stably over 20 runs. The model performance results on BC5 CDR at token-level are shown in Supplementary Table S2.

Compared with 7 very recent state-of-the-art related models, D3NER could yield the best chemical NER performance in terms of *F*1 and Recall (*R*) as well. Interestingly, *F*1 of D3NER for chemicals is even 1.74% higher than that of the TaggerOne joint model (see Table 2 for more details). Surprisingly, D3NER could always yield the both chemical and disease NER performance that are of the highest recall scores among those of such seven models, when being evaluated on two benchmark datasets.

For the disease NER, when being evaluated on the BC5 CDR corpus, D3NER out-performs 6 among 7 compared state-of-the-art models; and only performs worse than the joint model by Lou *et al.* (2017) (1.55% lower in *F*1). It is reasonable since the joint model can use the feedback from NEN to reduce errors in NER. However, note that D3NER still yields performance with recall remarkably higher than that of Lou *et al.* (2017), i.e. 2.31% higher. In contrast, when being assessed on the NCBI Disease corpus, D3NER could yield the performance with *F*1 of 2.36% better than that of this joint model of Lou *et al.* (2017). There is a significant decrease in recall (from 83.09% down to 74.89%) for the model of Lou *et al.* (2017) when being evaluated on this NCBI Disease dataset. D3NER, however, could retain its predictive power on this dataset, yielding the performance with *F*1 only 0.27% worse than that on the BC5 CDR corpus.

On the NCBI Disease corpus, we also observe the same phenomenon. In terms of F1, D3NER could yield the disease NER performance that are better than those from 6 among 7 compared state-of-the-art models. The exception is from the model of Habibi *et al.* (2017), which

Table 2. Performance of D3NER and compared state-of-the-art models on two benchmark corpora for Disease and Chemical NER

Model	BC5 CDR chemical			BC5 CDR disease			NCBI disease		
	P	R	F	P	R	F	P	R	F
Dnorm	–	–	–	82.00	79.50	80.70	82.20	77.50	79.80
tmChem	93.20	84.00	88.40	–	–	–	–	–	–
TaggerOne[a]	92.40	84.70	88.40	83.10	76.40	79.60	83.50	79.60	81.50
Habibi *et al.*	92.18	89.94	91.05	84.19	82.79	83.49	86.43	82.92	**84.64**
Wei *et al.*	–	–	–	85.28	83.30	84.28	–	–	–
Luo *et al.*	93.49	91.68	92.57	–	–	–	–	–	–
Our model	93.73	**92.56**	**93.14**	83.98	**85.40**	84.68	85.03	**83.80**	84.41
TaggerOne[b]	**94.20**	88.80	91.40	85.20	80.20	82.60	85.10	80.80	82.90
Lou *et al.*	–	–	–	**89.61**	83.09	**86.23**	**90.72**	74.89	82.05

Note: The highest values for each metric of each entity type are highlighted in bold.

[a]TaggerOne NER only.

[b]TaggerOne joint model.

is only 0.23% better than ours. Nevertheless, it is worth noting that the performance scoring metrics of their model are reported on different test sets (Habibi *et al.*, 2017), which is not the same test sets as used in D3NER. For each corpus, Habibi *et al.* (2017) first merged three subsets (training, development and test set, as described in Table 1) and then randomly divided the resulted set into three subsets again, what they called the training, development and test set, upon which their model is trained, fined-tuned and evaluated, respectively. We did evaluate D3NER on the original test set provided in each corpus as, to the best of our knowledge, this set has been carefully prepared by the corpus' creators as a 'real' test set in reality, which contains data different enough from those in the training set. This guarantees that the performance scoring metrics reported for D3NER are not biased, reflecting its true NER power. Interestingly, when being evaluated on the BC5 CDR Chemical and the FSU-PRGE gene/protein corpus, D3NER could yield the performance with *F*1 of 1.19 and 0.37% better than that of the model of Habibi *et al.* (2017), respectively.

Among the models that are evaluated on both the BC5 CDR and the NCBI Disease corpora, D3NER exhibits itself as the most stable model in disease NER. D3NER has the smallest *F*1 difference over such two corpora, i.e. 0.27%. The TaggerOne joint model comes in second, with this *F*1 difference of 0.30%. However, it does not hold true for the original TaggerOne (without jointing NER and NEN) when this different is up to 1.9%. This demonstrates the capability of D3NER to efficiently handle the out-of-vocabulary (OOV) issue, which happens very often in reality. To show more evidences, we note that D3NER with a randomly initialized token embedding performs much worse than that with the pre-trained embedding for gene/protein NER on FSU-PRGE, both disease and chemical NER on BC5 CDR. In particular, the former could only reach the averaged F1 of 80.39, 89.49 and 85.19% for disease, chemical and gene/protein NER, respectively. These numbers are respectively 84.68, 93.14 and 87.62% in case of the latter. A statistical analysis points out that the OOV issue occurs at different degrees in two benchmark corpora, upon which D3NER is assessed. On the BC5 CDR corpus, the training, development and test sets are quite different from each others, i.e. the development set contains 3887 new tokens that do not appear in the training set, and the test set contains such 3208 tokens. On the NCBI Disease corpus, these differences are at a less degree, i.e. the development set has only 653 new tokens, and the test set has 651.

4.2 Impact of different embeddings

We study the contribution of each embedding to the D3NER performance by removing each of them in turn from D3NER and then evaluating the model on the FSU-PRGE corpus (for gene/protein NER) and BC5 CDR corpus (for chemical and disease NER). In this regard, we also evaluate D3NER when using only token embeddings and only character-level embeddings.

The experimental results show that all embeddings help D3NER to boost its performance (in terms of the increments in *F*1) for disease, chemical and gene/protein NER (see Table 3 and Supplementary Table S4). The contribution, however, varies among them, e.g. *TE* contributes most, *CE* comes in second for disease and gene/protein NER (this place for chemical NER is *AE*), whereas *PE* just brings the least improvements. *TE* contributes up to 97.92 and 98.17% of the power of D3NER in recognition of chemicals and diseases, respectively. These proportions are only 94.51 and 94.32% for *PE*. Especially, *TE* is even more powerful than the rest embeddings altogether in the chemical, disease and gene/protein NER with D3NER.

Table 3. Impact of different embeddings on the chemical and disease NER performance of D3NER

Model	Chemical			Disease		
	P	R	F1	P	R	F1
D3NER	93.73	92.56	93.14	83.98	85.40	84.68
Without AE	92.13	91.14	91.63	83.49	84.36	83.92
Without CE	92.57	91.55	92.06	82.41	85.28	83.81
Without PE	93.64	92.18	92.90	83.79	85.29	84.53
Without TE	90.46	87.73	89.06	80.50	80.28	80.38
CE only	89.02	86.89	88.03	80.39	79.37	79.87
TE only	91.16	91.25	91.20	81.92	84.40	83.13

Note: Results reported on the BC5 CDR corpus.

Table 4. Impact of fine-tunning embeddings as the D3NER's hyper-parameters

Embeddings	Chemical			Disease		
	P	R	F1	P	R	F1
Fixed	92.68	92.12	92.39	83.28	85.42	84.33
Fine-tuned	93.73	92.56	93.14	83.98	85.40	84.68

Note: Results reported on BC5 CDR corpus.

The contribution of embeddings to recognition of each named entity type is also different. *AE* has more effect in recognition of chemical named entities than of diseases and genes/proteins. This order is NER of genes/proteins, followed by chemicals and then diseases for *CE*. Adding *AE* to D3NER would bring the performance improvement of up to 1.51% increment significantly to *F*1 for the chemical NER, only of 0.76 and 0.26% for the disease and gene/protein NER, respectively. These numbers are 1.08, 0.87 and 1.26% for the addition of *CE* to D3NER.

4.3 Impact of fine-tunning embeddings

We examine the impact of fine-tunning embeddings in chemical and disease NER by comparing the performance of D3NER with that of an variant of it, in which all embeddings (including *C*, *PE* and *AE*) are not fine-tuned (kept fixed) during the model training. The comparative results of two models on the FSU-PRGE corpus (for gene/protein NER) (Supplementary Table S5) and BC5 CDR corpus (for chemical and disease NER) (Table 4) demonstrate that fine-tuning embeddings has a certain effect on the performance of D3NER. An statistically significant increment of *F*1 is observed for chemical, disease and gene/protein NER when D3NER uses fine-tunned embeddings, i.e. 0.75, 0.35 and 0.14%, respectively. Without this improvement, D3NER would have been ranked second in chemical NER among seven compared models.

5 Error analysis

Table 5 shows some chemical and disease NER examples that D3NER disagreed with the golden annotation in BC5 CDR. The confusion matrix of D3NER on the BC5 CDR corpus can be found in the Supplementary Table S3. Some typical gene/protein NER errors on FSU-PRGE are showed in Supplementary Table S6. The error examples are chosen as reasonably good indications of certain disadvantages of D3NER, which is possibly useful for further improving the model in the follow-up.

Table 5. Examples for errors caused by D3NER on the BC5 CDR corpus

#	PMID	Golden annotation	Predicted label	Errors counter	Cause of errors
1	12119460	Chemical '*5-FU*' and chemical '*FA*'	Chemical '*5-FU/FA*'	2 FN, 1 FP	Conjunctions create boundary errors
2	9578276	Disease '*hyper- or hypotension*'	Diseases '*hypotension*'	1 FN, 1 FP	
3	7176945	Chemical '*scoline*' and disease '*pain*'	Disease '*scoline pain*'	2 FN, 1 FP	Coordinations of chemical and disease create boundary errors
4	24451297	Disease '*Drug-Induced Acute Liver Injury*'	Disease '*Liver Injury*'	1 FN, 1 FP	
5	24341598	Chemical '*sodium chloride*'	Chemical '*isotonic sodium chloride*'	1 FN, 1 FP	Boundary errors due to adjectives
6	24897009	Disease '*peripheral neuropathy*'	Disease '*Optochiasmatic and peripheral neuropathy*'	1 FN, 1 FP	
7	9125676	Chemical '*d, l-sotalol*'	Chemical '*sotalol*'	1 FN, 1 FP	Boundary errors due to term's complex structure (i.e. special character, long length, etc.)
8	8829135	Disease '*learning and post-training consolidation deficits*'	Disease '*consolidation deficits*'	1 FN, 1 FP	
9	24341598	Chemical '*Contrast*'	–	1 FN	Semantic sensitive, i.e, term (or part of term) looks like generic term
10	25951420	Disease '*particulate matter*'	–	1 FN	
11	16323982	Chemical '*C*'	–	1 FN	Abbreviations bring confusion
12	11708428	Chemical '*PDN*'	Disease '*PDN*'	1 FN, 1 FP	
13	23892921	Disease '*RRMM*'	Chemical '*RRMM*'	1 FN, 1 FP	
14	6631522	Chemical '*COX-2 inhibitors*'	–	1 FN	Our model's limitations
15	24675088	Disease '*subcellular degeneration*'	–	1 FN	
16	24341598	–	Disease '*CIN*'	1 FP	Human imperfect annotation
17	24618873	–	Disease '*Cerebellar and oculomotor dysfunction*'	1 FP	
18	17879217	–	Disease '*staphylococcal endocarditis*'	1 FP	

We here present some error analyses on BC5 CDR as the same is also occurred in FSU-PRGE. Multiple-token named entities are more likely to cause errors at the borders, which are severely punished in the exact matching based evaluation (both *FN* and *FP*) (see examples 1–10). Conjoined (examples 1–2) and nested terms (examples 3–4) may cause errors since D3NER disagreed with the golden annotation about labeling them as one or several entities and determining their boundaries. In examples 5–6, D3NER also included adjectives into the recognized terms while the golden annotation excluded. Vice versa, for some long and complex entities, D3NER just recognized the main parts, missing the remaining parts of them (examples 7–8). Some errors come from misleading words (examples 9–10), which are context-sensitive (a generic term in almost contexts can be a part of biomedical entity in a few other contexts in the corpus). Examples 11–13 show errors caused by abbreviations; some abbreviations are not annotated in the golden annotation, whilst others are too confused to be recognized exactly; abbreviations also cause most of the confusions between chemicals and diseases. Examples 14–15 are shown up because of our model limitations. Examples 16–18 are errors caused by the disagreement between our tagging schema and the golden annotation since we considered them as entities but the golden annotation did not.

6 Conclusions

In conclusion, this paper presents D3NER, a novel biomedical named entity recognition model using conditional random fields and bidirectional long short-term memory improved with jointly fine-tuned embeddings of various linguistic information. We evaluate D3NER on three benchmark datasets, i.e. the BioCreative V Chemical Disease Relation (BC5 CDR) corpus, the NCBI Disease corpus and the gene/protein FSU-PRGE corpus, which have also been used for performance evaluation in seven very recent state-of-the-art related models to which D3NER is compared. Experimental results demonstrate the power of D3NER in recognition of chemical, disease and gene/protein named entities. D3NER could yield excellent performance for chemical NER and very good for disease and gene/protein NER in terms of three popular performance scoring metrics.

We compared D3NER with seven very recent state-of-the-art biomedical NER models, of which five are of the same type as D3NER and two are even of the joint model for NER and NEN. Joint learning has been proven to bring performance improvements to NER (Leaman and Lu, 2016; Lou *et al.*, 2017). To this end, D3NER outperforms all seven models in chemical NER. For disease NER, D3NER outperforms six (including 1 joint model) on both evaluation corpora. It comes behind the remaining joint model of NER and NEN only on one corpus; and vice versa on the other corpus. For gene/protein NER, D3NER outperforms the very recently state-of-the-art model of Habibi *et al.* (2017). Interestingly, D3NER exhibits itself as the most stable model among all compared ones. It always could yield the performance with the highest recall for both chemical and disease NER on two benchmark corpora. Each of the fine-tuned embeddings used in D3NER has been demonstrated to contribute at a different certain degree to the model performance. Combining them altogether, however, could bring the best possible performance to D3NER, allowing it to retain its favorability over compared models.

We anticipate that D3NER shipped with a standalone executable program with the easy-to-use command interface can be integrated as a better alternative NER model into biomedical knowledge

extracting systems. To utilize practical uses of D3NER tool, we provide its computational run time related information as a reference in the Supplementary Tables S7–S9.

Acknowledgements

We are grateful to the Vietnam National Foundation for Science and Technology Development (NAFOSTED) for its financial support through the project [102.05 – 2016.14]. We would like to thank the Department of Computational Science and Engineering, and KTLab both at the VNU UET for the support. We also thank the anonymous reviewers for their comments and suggestions.

Funding

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2016.14.

Conflict of Interest: none declared.

References

Bengio,Y. and Glorot,X. (2010) Understanding the difficulty of training deep feedforward neural networks. In: *proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pp. 249–256.

Campos,D. *et al*. (2012) Biomedical named entity recognition: a survey of machine-learning tools. In *Theory and Applications for Advanced Text Mining, IntechOpen*. doi: 10.5772/51066.

Caruana,R. *et al*. (2001). Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping. In: *Advances in Neural Information Processing Systems*, pp. 402–408.

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

Habibi,M. *et al*. (2017) Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, **33**, i37–i48.

Hahn,U. *et al*. (2010) A proposal for a configurable silver standard. In: *Proceedings of the Fourth Linguistic Annotation Workshop*. pp. 235–242.

Hochreiter,S. and Schmidhuber,J. (1997) Long short-term memory. *Neural Comput*., **9**, 1735–1780.

Ioffe,S. and Szegedy,C. (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *International Conference on Machine Learning*, pp. 448–456.

Jagannatha,A.N. and Yu,H. (2016) Structured prediction models for RNN based sequence labeling in clinical text. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 856–865

Krallinger,M. *et al*. (2015) CHEMDNER: the drugs and chemical names extraction challenge. *J. Cheminform*., **7**, S1.

Kim,J.D. *et al*. (2009) Overview of BioNLP'09 shared task on event extraction. In: *BioNLPWorkshop*, pp. 1–9.

Lafferty,J. *et al*. (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: *Proceedings of the 18th International Conference on Machine Learning*, pp. 282–289.

Lample,G. *et al*. (2016) Neural architectures for named entity recognition. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 260–270.

Leaman,R. and Gonzalez,G. (2008). BANNER: an executable survey of advances in biomedical named entity recognition. *Pac. Symposium Biocomput*., **13**, 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) tmChem: a high performance approach for chemical named entity recognition and normalization. *J. Cheminform*., **7**, S3.

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

LeCun,Y. *et al*. (1998) Gradient based learning applied to document recognition. *Proc. IEEE*, **86**, 2278–2324.

Le,H.Q. *et al*. (2016) Sieve-based coreference resolution enhances semi-supervised learning model for chemical-induced disease relation extraction. *Database*, **2016**, baw102.

Le,H.Q. *et al*. (2015) The UET-CAM system in the BioCreAtIvE V CDR task. In: *Fifth BioCreative Challenge Evaluation Workshop*, pp. 208–213.

Li,J. *et al*. (2015) Annotating chemicals, diseases, and their interactions in biomedical literature. In: *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*, pp. 173–182.

Limsopatham,N. and Collier,N. (2016) Learning orthographic features in bi-directional lstm for biomedical named entity recognition. In: *BioTxtM 2016*, pp. 10.

Ling,W. *et al*. (2015) Finding function in form: Compositional character models for open vocabulary word representation. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1520–1530.

Luo,L. *et al*. (2018) An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. *Bioinformatics*, **34**, 1381–1388.

Lou,Y. *et al*. (2017) A transition-based joint model for disease named entity recognition and normalization. *Bioinformatics*, **33**, 2363–2371.

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.

Ma,X. and Hovy,E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1064–1074.

McCallum,A. *et al*. (2000) Maximum entropy markov models for information extraction and segmentation. *ICML*, **17**, 591–598.

Mehryary,F. *et al*. (2016) Deep learning with minimal training data: Turkunlp entry in the bionlp shared task 2016. In: *Proceedings of the 4th BioNLP Shared Task Workshop*, Association for Computational Linguistics, pp. 73–81.

Mikolov,T. *et al*. (2013) Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst*., **26**, 3111–3119.

Pradhan,S. *et al*. (2015) Evaluating the state of the art in disorder recognition and normalization of the clinical narrative. *J. Am. Med. Inform. Assoc*., **22**, 143–154.

Pyysalo,S. *et al*. (2013) Distributional semantics resources for biomedical text processing. *LBM*, **2013**, 39–44.

Rabiner,L. and Juang,B. (1986) An introduction to hidden Markov models. *IEEE ASSP Mag*., **3**, 4–16.

Ratinov,L. and Roth,D. (2009) Design challenges and misconceptions in named entity recognition. In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pp. 147–155.

Sohn,S. *et al*. (2008) Abbreviation definition identification based on automatic precision estimates. *BMC Bioinformatics*, **9**, 402.

Srivastava,N. *et al*. (2014) Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res*., **15**, 1929–1958.

Verwimp,L. *et al*. (2017) Character-word LSTM language models. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, long paper, Vol. 1, pp. 417–427.

Wei,C.H. *et al*. (2016a) Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task. *Database (Oxford)*, 2016, baw032.

Wei,Q. *et al*. (2016b) Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks. *Database*, **2016**, baw140.

Zhou,G. *et al*. (2004) Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, **20**, 1178–1190.