Investigating of Disease Name Normalization Using Neural Network and Pre-Training

YINXIA LOU1,4, TAO QIAN2, FEI LI3, JUNXIANG ZHOU4, DONGHONG JI1, AND MING CHENG5

1Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430074, China
2School of Computer Science and Technology, Hubei University of Science and Technology, Xianning 437000, China
3Department of Computer Science, University of Massachusetts Lowell, Lowell, MA 01854, USA
4School of Medical Information, The First Affiliated Hospital of Zhengzhou University, Zhengzhou 450052, China
5Department of Medical Information, The First Affiliated Hospital of Zhengzhou University, Zhengzhou 450052, China

Corresponding author: Tao Qian (taoqian@whu.edu.cn)

This work was supported in part by the Major Project of the National Social Science Foundation of China under Grant 11&ZD189, in part by the National Key Research and Development Program of China under Grant 2017YFC1200500, in part by the Natural Science Foundation Project of Hubei under Grant 2018CFB690, in part by the Doctoral Initiation Foundation of Hubei University of Science and Technology under Grant 2016-19XB012, in part by the Humanities and Society Scientific Research Program of Hubei Province Education Department under Grant 18Y139, and in part by the Science and Technology Project of Hubei Province Education Department under Grant B2018173.

ABSTRACT Normalizing disease names is a crucial task for biomedical and healthcare domains. Previous work explored various approaches, including rules, machine learning and deep learning, which focused on only one approach or one model. In this study, we systematically investigated the performances of various neural models and the effects of different features. Our investigation was performed on two benchmark datasets, namely the NCBI disease corpus and the BioCreative V Chemical Disease Relation (BC5CDR) corpus. The convolutional neural network (CNN) performed the best (F1 90.11%) in the NCBI disease corpus and the attention neural network (Attention) performed the best (F1 90.78%) in the BC5CDR corpus. Compared with the state-of-the-art system, DNorm, our models improved the F1s by 1.74% and 0.86% respectively. In terms of features, character information could improve the F1 by about 0.5-1.0% while sentence information worsened the F1 by about 3-4%. Moreover, we proposed a novel approach for pre-training models, which improved the F1 by up to 9%. The CNN and Attention models are comparable in the task of disease name normalization while the recurrent neural network performs much worse. In addition, character information and pre-training techniques are helpful for this task while sentence information hurts the performance. Our proposed models and pre-training approach can be easily adapted to the normalization task for any other type of entities. Our source code is available at: https://github.com/yx100/EntityNorm.

INDEX TERMS Deep learning, disease name normalization, text mining, natural language processing.

I. INTRODUCTION

The task of disease name normalization is to map a disease mention in raw text to a unique concept in a controlled vocabulary. Take the mentions “dyskinesias” and “Parkinson’s disease” in Figure 1 for example. They can be normalized to “D004409: Dyskinesia, Drug-Induced” and “D010300: Parkinson Disease” of the MEDIC vocabulary [1], respectively. Disease name normalization has attracted increasing attention in biomedical text mining and natural language processing (NLP) research [2]–[10]. It is a fundamental task that can facilitate downstream tasks such as knowledge base construction [11] and relation extraction [12]–[15].

FIGURE 1. Examples of disease name normalization. Two disease names are normalized into D004409 and D010300 of the MEDIC vocabulary [1].
Disease name normalization is a challenging task due to the fact that disease names can be expressed flexibly in raw text, due to numerous naming conventions, synonyms and morphological variations. Abbreviations and acronyms of disease names also significantly increase the difficulty of the task. In addition, the continuous emergence of new disease names brings constant challenges to the task.

Previous work explored various approaches for disease name normalization. Typical methods include dictionary lookup and pattern matching through manually defined rules [5], [16]. However, it can be infeasible for such methods to cover all the situations as the mentions of disease names are numerous and keep changing. In addition, the rules that are designed for certain types of names (e.g., disease) may not be applicable for other types of names (e.g., chemical). To address these issues, some studies utilized machine learning [6], [17] and deep learning [5], [18], [19] for disease name normalization. However, none of these performed systematic analysis and comparison between different types of models using the same dataset, thus making it difficult to choose a model in practice. In addition, previous work only employed word-level information but neglected other related information such as characters and sentence contexts.

In this paper, we systemize compare three typical neural network models, namely the convolutional neural network (CNN) [20], recurrent neural network (RNN) [21] and Attention network [22] for disease name normalization. The reasons why we select these models are mainly because: (1) previous work for deep-learning-based normalization usually used CNN or RNN models [19], [23]. (2) the main models for deep-learning-based NLP are CNN, RNN and Attention models.

Besides neural models, we also incorporated three widely-used neural features, i.e., words, characters and sentences into our models. To the best of our knowledge, this is the first work to systematically analyze and compare the effect of different neural networks and features using a standard benchmark. Moreover, we proposed a novel approach to pre-train models using vocabularies of normalization tasks, which is very effective for alleviating the problem of insufficient training data, and can also be transferred to other normalization tasks.

Our investigation was carried out on two benchmark datasets, namely the NCBI disease corpus [24] and BioCreative V Chemical Disease Relation (BC5CDR) corpus [25]. Results show that the performances of the CNN and Attention network are comparable for the task of disease name normalization, but RNN did not perform well, which is likely due to that the lengths of mentions are generally short. Moreover, character information is helpful for our task but sentence information hurts the performance. In addition, our pre-training approach improved the performances of our models significantly. Compared with the state-of-the-art model DNorm [17], our best models outperformed it by 1.74% and 0.86% in F1 on two datasets, respectively, giving the highest score in the literature.

This paper mainly includes the following contributions:
- We explored the impact of different features (word, character and sentence) on disease mention normalization. We found that word features play a major role for this task, and character features could improve the performance to some extent, while sentence features are not significantly helpful.
- We showed the effectiveness of pre-training models using the vocabularies of disease mention normalization. Such approach could alleviate the problem of insufficient training data, which could be easily transferred to other normalization tasks.
- The performance of our models is comparable with several strong systems of disease mention normalization. To facilitate related research and reproduce our results, we published our source code and models.

II. RELATED WORK

Disease entity normalization has been proposed as subtasks of SemEval-2014 [3] and BioCreative V [13]. Early methods mainly focused on dictionary lookup and heuristic rules [2], [5], [16], [17]. Dogan and Lu [16] proposed an inference method, which was mainly based on dictionary lookup and pattern matching for disease entity normalization. D’Souza and Ng [5] leveraged a multi-pass sieve system based on manual rules and dictionary lookup. They defined ten rules with different priorities to measure morphological similarities between disease mentions and the concepts in the vocabulary. However, it is impossible to design complete rules to cover all the situations since entity mentions and concept names of the vocabulary are various and volatile.

To address these issues, there has been prior work that explores machine learning techniques for disease name normalization [6], [17]. For example, Leaman et al. [17] proposed DNorm, a strong normalization model based on the learning-to-rank approach. It achieved competitive results on the NCBI disease corpus [24]. Leaman and Lu [6] further proposed TaggerOne, a joint model for named entity recognition and normalization, which obtained the state-of-the-art performance on several benchmark corpora [24], [25]. In this paper, we take DNorm as one of our baselines, since TaggerOne is a joint model using different sources of information.

Recently, researchers have investigated disease name normalization using deep learning [26]. Deng et al. [27] presented a two-step ensemble CNN method that normalizes microbiology-related entities, and achieved reasonable performance in the BioNLP-19 task Bacteria Biotope. Karadeniz and Özgür [28] proposed an unsupervised method for entity linking tasks using word embeddings and a syntactic parser. Niu et al. [29] presented a multi-task character-level attentional network model for medical entity normalization. Cho et al. [18] utilized word embeddings to generate the representations of disease names and used these representations to calculate the similarities between mentions and concepts. Liu and Xu [19] also employed the similar method but the
difference is that they used recurrent neural networks (RNNs) instead of directly using word embeddings. Li et al. [23] handled disease name normalization using a hybrid method, i.e., utilizing the rule-based method [5] to generate several candidates and then using convolutional neural networks (CNNs) to select the final answer in the candidates. Different from prior work which focused on building competitive models, our work focuses on comparing various neural models and features.

III. METHODS
In this section, we introduce a unified framework for model comparison and three instance normalization models that utilize words, characters and sentence contexts respectively. For each model, we employed three popular neural networks, namely CNN [20], LSTM [21] and Attention [30].

A. WORD-BASED NORMALIZATION MODEL (WNM)
WNM only leverages the words in a mention as input. Its architecture is illustrated in Figure 2(a). Below we describe the details of the WNM from bottom to top.

1) INPUT LAYER
Given a disease mention \( m = (w_1, \ldots, w_{|m|}) \) with \(|m|\) words (e.g., renal damage), we used \( e_{w_i} \in \mathbb{R}^d \) to denote the \( d \)-dimensional word embedding of the \( i \)-th word \( w_i \) (e.g., renal). A word embedding [31] is actually a vector to denote the word. It has been widely used in the NLP community and many word embedding tools have been proposed such as GloVe [32], Word2vec [31] and FastText [33]. In this paper, we utilized FastText\(^1\) since it is better than the other two by capturing sub-word information.

\(^1\)https://github.com/facebookresearch/fastText
In particular, we collected 0.8 million PubMed abstracts by searching the names in the MEDIC vocabulary. Then the Natural Language Toolkit (NLTK)\textsuperscript{2} toolkit was used to split each abstract into sentences and tokenize each sentence into words. All words were converted into lowercase. After that, the preprocessed text was fed into FastText to generate 200-dimensional word embeddings. Since not all the names in the MEDIC vocabulary can be found in PubMed, there are still about 20\% out-of-vocabulary (OOV) words that are not in our pre-trained embeddings. During training, the pre-trained embeddings were fine-tuned with our models.

2) HIDDEN LAYER
The hidden layer of the WNM calculates a representation \( r_m \) for the input mention \( m \), given the embeddings \((e_{w_1}, \ldots, e_{w_{|m|}})\) of its words. Since the input mention \( m \) can be variable-length, it is natural to adopt the CNN, LSTM or Attention to transform the list of embeddings into a fixed-length representation.

\textbf{a: CNN-BASED WNM}

Figure 3(a) shows the architecture of the CNN-based WNM. Given the embeddings \((e_{w_1}, \ldots, e_{w_{|m|}})\) of a mention \( m \) (e.g., acute liver failure), we can consider them to be a sequence \( M \). The convolutional layer of the CNN can extract different features using the filters with different sizes [20]. In this paper, we selected 2, 3 and 4 as our filter sizes and we also use a special token, “PAD”, to facilitate the convolution operation.

For example, if the input is “acute liver failure” and the filter size is 3, convolution is performed 3 times on 3 word windows, namely “PAD acute liver”, “acute liver failure” and “liver failure PAD”, respectively. If these word windows can be written as \( M_{[0:2]}, M_{[1:3]} \) and \( M_{[2:4]} \), one convolutional operation can be formalized as:

\[
c_i = \text{ReLU}(W_1M_{[i+i-k+1]} + b_1),
\]

where \( W_1 \) and \( b_1 \) indicate the weight and bias of this filter, and \( k \) indicates the filter size. Therefore, we obtain several outputs after all the convolution of this filter finishes. To integrate all these outputs into one vector \( p \), a max-pooling operation is used:

\[
p_j = \max_{1 \leq i \leq N} c_{ij},
\]

where \( j \) denotes the \( j \)-th dimension of the filter output \( p \) and \( N \) denotes the convolutional number.

The outputs of the three filters \( P_{filter_2}, P_{filter_3}, P_{filter_4} \) are concatenated to form the final representation \( r_m \) of the mention \( m \), given by:

\[
r_m = P_{filter_2} \oplus P_{filter_3} \oplus P_{filter_4},
\]

where \( \oplus \) represents the concatenation operation.

\textbf{b: LSTM-BASED WNM}

Figure 3(b) shows the architecture of the LSTM-based WNM. Given the word embeddings \((e_{w_1}, \ldots, e_{w_{|m|}})\) of a mention \( m \) (e.g., acute liver failure), a standard LSTM unit [21] at time step \( i \) is calculated by the following equations:

\[
i_i = \sigma(W_i e_{w_i} + U_i h_{i-1} + b_i), \]

\[
f_i = \sigma(W_f e_{w_i} + U_f h_{i-1} + b_f), \]

\[
\tilde{c}_i = \tanh(W_c e_{w_i} + U_c h_{i-1} + b_c), \]

\[
c_e_i = f_i \odot \tilde{c}_i + i_i \odot \tilde{c}_i, \]

\[
o_i = \sigma(W_o e_{w_i} + U_o h_{i-1} + b_o), \]

\[
h_i = o_i \odot \tanh(c_e_i),
\]

where \( e_{w_i} \) denotes the current input, \( i_i, f_i \) and \( o_i \) represent the values of an input gate, forget gate and output gate at time step \( i \), respectively. \( W_i, W_f, W_c, W_o, U_i, U_f, U_c \) and \( U_o \) are weight matrices. \( b_i, b_f, b_c \) and \( b_o \) are bias vectors. \( \tilde{c}_i \) is the candidate cell value, and \( c_e_i \) is the cell value. \( h_i \) is hidden state vector. \( \sigma \) and \( \odot \) are the element-wise sigmoid and the element-wise product, respectively.

In our study, we use bi-directional LSTM. This architecture has been empirically demonstrated effective in previous work [34]. In this paper, we use \( \tilde{h}_i \) to indicate the hidden state at step \( i \) in the left-to-right direction and \( \tilde{h}_i \) to indicate the one in the right-to-left direction. To obtain the vector

\[...\]
representation $r_m$ of the mention $m$, we concatenate the last
hidden state $\overrightarrow{h}_{[m]}$ in the left-to-right direction and the last
hidden state $\overleftarrow{h}_{1}$ in the right-to-left direction:

$$r_m = \overrightarrow{h}_{1} \oplus \overleftarrow{h}_{[m]}, \quad (5)$$

where $\oplus$ represents the concatenation operation. For example,
given a disease mention “acute liver failure”, the hidden
state of “acute” in the right-to-left direction and the hidden
state of “failure” in the left-to-right direction are concatenated.

**C: ATTENTION-BASED WNM**

Figure 3(c) shows the architecture of the Attention-based
WNM. The attention mechanism can be used to learn a weight
for each word of a mention to reflect its importance for the
final representation of the mention [30]. We adopt the
widely-used attention mechanism [22], which employs the
dot-product and summation to generate an output vector.

Given the embeddings $(e_{w_1}, \cdots, e_{w_{|m|}})$ of a mention $m$,
the attention network first computes a weight $\alpha_i$ for the $i$-th word:

$$u_i = v^T e_{w_i}, \quad \alpha_i = \frac{\exp(u_i)}{\sum_{i=1}^{|m|} \exp(u_i)}, \quad (6)$$

where $v^T$ is the transpose of the attention weight $v$ and $u_i$
denotes a scalar score for the $i$-th word. Then the attention
mechanism utilizes the softmax operation to normalize these
scores. After that, we compute the mention representation $r_m$
by weightedly summing all the embeddings of the mention:

$$r_m = \sum_{i=1}^{|m|} \alpha_i e_{w_i}. \quad (7)$$

**3) OUTPUT LAYER**

After the mention representation $r_m$ is generated by the hidden
layer, a linear transformation on $r_m$ is preformed as follow:

$$h = W_h r_m + b_h, \quad (8)$$

where $W_h \in \mathbb{R}^{d_h \times d}$ and $b_h \in \mathbb{R}^{d_h}$ are the weight and bias
in the linear transformation layer. The purpose of adding this
layer is to generate a unified input for the output layer and
reduce the dimension of the mention representation $r_m$.

In the output layer, we treat the disease name normalization
as a classification problem. The softmax classifier is
employed to predict a concept identifier (ID) $y$ for the mention
$m$. This process can be formalized as:

$$s_c = W_i h + b_i, \quad p(i) = \frac{\exp(s_c)}{\sum_{i=1}^{|V|} \exp(s_c)} , \quad y = \arg \max_i p(i), \quad (9)$$

where $W_i \in \mathbb{R}^{|V| \times d_h}$ and $b_i \in \mathbb{R}^{|V|}$ are the weight and bias
in the output layer, respectively. $|V|$ is the concept number in the
vocabulary. After the probability distribution $p$ is generated
from the output layer, we select the one with the maximal
probability as the prediction.

During training, the cross-entropy loss is used to optimize
our models:

$$L = - \sum_m \log p(y|m), \quad (10)$$

where $p(y|m)$ indicates the probability of the gold answer $y$ of
a mention $m$. We use the Adam algorithm [35] to control the
training process and L2 normalization to prevent overfitting.
The batch size is set as 16 for all our experiments.

**B. CHARACTER-BASED NORMALIZATION MODEL (CNM)**

CNM does not only leverages the words in a mention as input, but also the characters of the words. Its architecture is illustrated in Figure 2(b). Character features have been demonstrated to be effective in prior work [14], since they are able to capture morphological information (e.g., prefixes and suffixes) and deal with the out-of-vocabulary problem [36]. Therefore, we investigated the effect of character information on mention normalization.

As shown in Figure 2(b), CNM takes the embeddings of the characters of a word as input, and generates the character-level representation of the word using the CNN, LSTM or Attention network. Since the process is similar to that of WNM, the details are not described in this section for conciseness.

After obtaining the character-level representation $r_c$ of the
word $w_i$, we concatenated it with the word embedding $e_{w_i}$:

$$r_{w_i} = e_{w_i} \oplus r_c. \quad (11)$$

where $r_{w_i}$ denotes the new representation of the word $w_i$. Then
$r_{w_i}$ is used for the upper layer instead of the original word
embedding $e_{w_i}$.

In our experiments, we randomly initialize the character
embeddings with a uniform distribution in the range of [-0.01, 0.01]. The dimension of character embeddings was set as to 20. During training, character embeddings are fine-tuned
with our models.

**C. SENTENCE-BASED NORMALIZATION MODEL (SNM)**

To investigate the effect of contextual information on mention
normalization, we propose sentence-based normalization model. Besides word and character features, SNM also leverages the words in the sentence that contains the target
mention. Its architecture is illustrated in Figure 2(c).

Given a sentence $s$, SNM first takes the embeddings of
the words in the sentence as input and then uses CNN, LSTM or Attention network, respectively, to encode the
embedding sequence into a fixed-length representation $r_s$.
After that, we concatenate the sentence representation $r_s$ with
the mention representation $r_m$:

$$\tilde{r}_m = r_s \oplus r_m, \quad (12)$$
where $\tilde{r}_m$ denotes the new representation of the mention $m$ with the information of the sentence context. Finally, $\tilde{r}_m$ is fed into the upper layer.

### D. PRE-TRAINING

Despite that deep learning models can be more accuracy to traditional statistical machine learning models, they demand more training data. However, labeled data are expensive in most tasks or domains. Recently, it has become popular to use automatically-labeled data to pre-train deep learning models before using them in various tasks [37], [38]. Motivated by this, we propose a novel semi-supervised method for the entity normalization task, utilizing all the names in the MEDIC vocabulary as extra training data. Figure 4 shows the process of pre-training and training.

Since the names in the vocabulary has already been categorized, they can be considered as automatically-labeled data. For example, the following list shows a real term (ID:D007680) in the MEDIC vocabulary:

- Disease Name: Kidney Neoplasms
- Synonym 1: Cancer of Kidney
- Synonym 2: Cancer, Renal
- Synonym 3: Kidney Cancers
- Synonym 4: Renal Neoplasms

In our study, we used the disease name and synonyms as the training data to pre-train our models. After pre-training, we initialized the parameters of our models with the pre-trained values and continued training our models on the corpus data. From this example, we can see that the pre-training method has several advantages.

First, such methods can theoretically be employed in any normalization task since each of them must have a vocabulary. Second, such methods can generate a large amount of high-quality training data, since vocabularies are usually very large and well-curated by human experts. Last but not least, the generated data from a vocabulary can improve the generalization of models. For instance, they include multiple expressions for a concept (e.g., “Kidney” and “Renal”). They permute the orders of disease names (e.g., “Cancer of Kidney” and “Kidney Cancers”), which is important for normalization tasks since the orders are usually not important.

Moreover, they add some trivial words (e.g., “of”) and plurals (e.g., “Cancers”), which makes models more robust by facing various situations.

### E. DATASETS

We validate the performances of our models on two datasets: the NCBI disease corpus [24] and the BioCreative V Chemical Disease Relation (BC5CDR) task corpus [25]. The NCBI disease corpus consists of 793 PubMed abstracts, which were separated into training (593), development (100) and test (100) sets. The NCBI corpus was annotated with disease mentions and their concept identifiers from either MeSH [39] or OMIM [40].

The BC5CDR corpus contains 1,500 PubMed abstracts with annotated disease mentions and their MeSH identifiers, which were equally partitioned into three sections for training, development and test, respectively. Table 1 gives the statistics of both corpora.

In our experiments, we use the MEDIC vocabulary of the Comparative Toxicogenomics Database (CTD) project. Following Dnorm [17], we used the MEDIC version published on April, 2012 (about 9,661 concepts and 67,000 names) for the experiments of the NCBI disease corpus, and used the version published on June, 2015 (about 11,885 concepts and 76,685 names) for the experiments of the BC5CDR corpus.

### F. HYPERPARAMETER SETTINGS

Table 2 shows the hyper-parameters used in our experiments. Some of the hyper-parameters were determined based on the prior work [14], and the others were tuned based on the development sets. The details of parameter tuning are as

---

**TABLE 1.** Statistics of the NCBI and BC5CDR corpora. “Mention” and “ID” refer to the numbers of disease mentions and unique concept identifiers.

| Corpus               | Abstracts | Mention | ID |
|----------------------|-----------|---------|----|
| NCBI training set    | 592       | 5145    | 670 |
| NCBI development set | 100       | 787     | 176 |
| NCBI test set        | 100       | 960     | 203 |
| BC5CDR training set  | 500       | 4182    | 1965 |
| BC5CDR development set| 500       | 4244    | 1865 |
| BC5CDR test set      | 500       | 4424    | 1988 |

**TABLE 2.** Hyper-parameter settings.

| Model | Hyper-parameter  | Value |
|-------|------------------|-------|
| WNM & SNM | Embeding size | 200   |
|        | LSTM hidden size | 100   |
|        | CNN filter size  | 2.34  |
|        | Output layer size| 100   |
| CNM   | Embeding size    | 20    |
|        | hidden size      | 10    |
|        | CNN filter size  | 3     |
|        | Learning rate    | 0.0001|
|        | Dropout           | 0.5 (NCBI) |
|        | Dropout           | 0.3 (BC5CDR) |
|        | Batch size       | 16    |

---

3http://ctdbase.org
4http://www.omim.org
5https://www.nlm.nih.gov/mesh
follows: for the CNN filter size, we tried [2,3,4] and [3,4,5]. The F1s using the former are 91.53% and 91.05% in the NCBI disease corpus, and the F1s using the latter are 91.76% and 91.46% in the BC5CDR corpus. For the learning rate, we selected 0.001, 0.0001 and 0.0005. In the NCBI disease corpus, the F1s are 90.02%, 91.38% and 90.57%, and in the BC5CDR corpus, the F1s are 91.13%, 91.53% and 91.08% respectively. For the dropout rate, we tried 0.1, 0.3 and 0.5. The results using 0.1 are 90.91% for the NCBI disease corpus and 90.56% for the BC5CDR corpus. The results are 91.07% and 91.32% using 0.3, and 91.46% and 90.87% using 0.5.

In addition, following the prior work [14], the dimension of word embeddings was set to 200 and that of character embeddings was set to 20. The dimension of character hidden layer was set to 10. For CNN, the character filter size was set to 3 and the word filter sizes were set to [2,3,4]. For LSTM, the dimension of LSTM hidden states was set to 100. The size of the output layer was set to 100. The learning rate was set to 0.0001 and the mini-batch size was set to 16.

G. EVALUATION METRICS
Following DNorm [17], our models were evaluated based on the abstract-level ID comparison. Assuming there are two ID sets for a abstract, one is called the gold ID set and the other is called the predicted ID set. If a predicted ID matches a gold ID, it is counted as true positives (TP). If not, it is counted as false positives (FP). All the non-matched gold IDs are counted as false negatives (FN). The precision \( P \), recall \( R \) and F-score \( F1 \) for this abstract were computed based on the following equations:

\[
P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F1 = \frac{2 \times P \times R}{P + R}. \tag{13}
\]

Finally, the \( P \), \( R \) and \( F1 \) for a corpus is the macro-averaged \( P \), \( R \) and \( F1 \) of all the abstracts.

H. PREPROCESSING
Given an abstract, we used the NLTK [41] to split it into sentences and tokenize these sentences into words. All the words were transformed into lowercase and punctuations were removed from disease mentions or names in the vocabulary. Composite disease mentions, such as “bladder and liver tumours: D001749|D008113” were separated into individual parts such as “bladder tumours: D001749” and “liver tumours: D008113”. We handled with the abbreviations in each abstract using Ab3p [42]. All abbreviations were replaced with their full names.

IV. RESULTS
A. RESULTS ON THE NCBI DISEASE CORPUS
Table 3 shows the results of our models and the baselines on the test set of the NCBI disease corpus. DNorm [17] gives the state-of-the-art results for disease name normalization. It learns the similarities between mentions and concept names through pairwise ranking, which has been widely used in information retrieval [43], [44]. The best version of DNorm was trained using both the training and development sets of the corpus. In addition, we also compared with MultiSieve [5], which utilizes manually-designed rules and the dictionary lookup method. To make fair comparison, we followed their setting. In addition, we run each model three times via randomly initialized model parameters. The results shown in the table are the means of these runs.

Since there are a number of models in this paper, it is exhausted to fully evaluate all the combinations of various neural networks. Therefore, after we tested the best architecture of WNM at the first step, we fixed the WNM and tried various networks for the character part to generate the CNNM. For SNM, the process is similar.

According to Table 3, we can see that the best architecture for WNM is using the convolutional neural network (WCNN), which gives the F1-score 89.11%. Surprisingly, the recurrent neural network (WLSTM) performs much worse than the other two kinds of neural networks. The reasons are discussed in the “Comparing the CNN, LSTM and Attention network” section (page 9). By adding character-level features (WCNN+CCNN), the precision, recall and F1-score of WCNN are further improved by 1.13%, 1.89% and 1.0% respectively. Moreover, we can also observe that the model is able to achieve better performances when both word and character features are represented with the CNN. In contrast, there are considerable decreases (about 4%) when sentence information is added into our models (SNM). Overall, the best architecture is CNN with both word and character features (WCNN+CCNN), which improves the recall and F1-score by 2.7% and 1.74% compared with DNorm, respectively.

B. RESULTS ON THE BCSCDR CORPUS
Table 4 shows the results of our models and the baselines on the test set of the BC5CDR corpus. First, the best architecture for WNM is the attention-based neural network (WATT).
TABLE 4. Results (%) on the BC5CDR corpus.

| Model | P     | R     | F1    |
|-------|-------|-------|-------|
| MultiSieve [5] | 84.43 | 85.98 | 84.78 |
| DNorm [17]  | 91.34 | 89.71 | 89.92 |
| WNM        | WCNN   | 89.28 | 91.53 | 90.09 |
|           | WLNSTM | 74.05 | 76.96 | 74.97 |
|           | WATT   | 89.46 | 91.66 | 90.23 |
| CNM        | WATT+CCNN | 89.97 | 91.95 | 90.69 |
|           | WATT+CLSTM | 89.83 | 91.90 | 90.58 |
|           | WATT+CATT | 90.32 | 92.12 | 90.78 |
| SNM        | WATT+CATT+SCNN | 86.83 | 88.92 | 87.57 |
|           | WATT+CATT+SLSTM | 86.44 | 88.79 | 87.28 |
|           | WATT+CATT+SATT | 86.77 | 88.61 | 87.37 |

“W”, “C”, and “S” denote word, character and sentence respectively. The bold number denotes the highest value in each column. Our best F1 (90.78% in WATT+CATT) is significantly higher than the F1 of DNorm (89.92%). The result with mark * demonstrates that the p value is below 0.05 using t-test compared with DNorm.

TABLE 5. Comparison of our model with other neural network models on both corpora.

| Model          | NCBI-disease Acc (%) | BC5CDR Acc (%) |
|----------------|----------------------|-----------------|
| Liu and Xu [19]| 85.30                | 86.27           |
| Li [23]        | 86.10                | 86.42           |
| our models     | 89.92                | 90.05           |

It achieves the F1-score 90.23%, which is slightly higher than the baseline. After the character features are used, the performances are improved no matter which kind of neural networks is employed. Thereinto, the attention-based architecture with word and character features (WATT+CATT) improves the F1-score most (0.55%). After adding the sentence context, all the F1-scores of SNM drop dramatically (about 3%). On the BC5CDR corpus, the best architecture is based on the word-level and character-level attention (WATT+CATT), which improves the recall and F1-score by 1.45% and 0.86% compared with DNorm. This is consistent with the results in Table 3.

DNorm achieves higher precision compared with our neural network models. One important reason is that its features consist of discrete word patterns, which fire only when exact matches occur. However, compared to neural models, this may lead to lower recall as shown in Table 3 and 4. One of the reasons for the lower false-negatives of the neural models can be that it can normalize more complex and ambiguous entities.

C. COMPARING OUR MODEL WITH OTHER MODELS

Table 5 shows the accuracy obtained by our models and other neural network models on both corpora. Liu and Xu [19] employed the RNN to model mention and concept representations, and computed similarities between them. Li et al. [23] first utilized heuristic rules to generate can didates and then used the CNN to select the final answer. Since both of them used the NCBI disease corpus and reported the accuracy of their models, we followed their settings and used our best-performed model (WCNN+CCNN) in the NCBI disease corpus to compare with theirs. The method of computing the accuracy is to use the number of total entity mentions to divide the number of correctly normalized mentions. As shown in Table 5, the accuracy of our model (WCNN+CCNN) is 4.62% and 3.82% higher than those of their models, respectively. In addition, CNN-based models (our model and Li et al. [23]) performed better than RNN-based models (Liu and Xu [19]), which is consistent with the results in Tables 3 and 4. For BC5CDR corpus, we re-implemented their models on BC5CDR corpus, and used our best-performed model (WATT+CATT) to compare with theirs. Comparison results are shown in Table 5. We can see our model (WATT+CATT) achieves better accuracy compared with their models.

V. DISCUSSION

A. EFFECT OF CHARACTER INFORMATION

Biomedical text is characterized by its richness and complexity of special vocabulary. Our pre-trained word embeddings cover about 80% of all the experimental data including the corpora and the vocabulary. To analyse the effect of character information, we performed a quantitative analysis for the WCNN and WCNN+CCNN models on the NCBI corpus. By calculating the precision of the datapoints that contain out-of-vocabulary (OOV) words, we found that the WCNN+CCNN model improved the precision for such datapoints by nearly 10% compared with the WCNN model. For example, the mention “adrenocortical” cannot be correctly normalized by the WCNN model, but it can be correctly normalized by the WCNN+CCNN model.

B. EFFECT OF SENTENCE-CONTEXT INFORMATION

As shown in Table 3, the best F1 for the SNM (WCNN+CCNN+SCNN) is 86.31%, which is 3.7% lower than the F1 of the WCNN+CCNN model. Consistently, the best F1 for the SNM (WATT+CATT+SCNN) in Table 4 is 87.57%, which is also 3% lower than the F1 of the WATT+CATT model. We can conclude that sentence-context information is not significantly helpful for disease mention normalization. This is not consistent with the discovery in previous work, in which sentence information has been shown useful for other NLP tasks [45], [46].

To gain insights into the effect of sentence-context information, we compare the F1s of the model without sentences,
FIGURE 5. Effect of pre-training. The pink bars denote the models without any pre-training. The blue bars denote the models using only pre-trained word embeddings. The red bars denote the models pretrained using vocabularies. The green bars denote the models using both embedding and vocabulary pre-training.

with those of the model with sentences containing different numbers of entities. Table 6 shows comparison results. We see that the F1s of the models with sentence information is lower than those of the models without sentence information, regardless of whether a sentence contains an entity, two, or three or more entities. The experiment results show that disease mention normalization is different from general NLP tasks and disease entities are not strong context dependent. For example, a sentence “A common human skin tumour is caused by activating mutations in beta-catenin.”, we can replace “skin tumour” with a more general disease name such as “tumour”. As a result, the two mentions have the same contexts but they should belong to different concepts in the vocabulary.

Table 6 also shows that when the sentence contains multiple entities, the F1s of models with sentence information decrease more compared with those of models without sentence information. This may be because the context information is similar for the entities, and the representation of one entity contains more noise information about other entities in the same sentence. Therefore, it is difficult for models to distinguish their representations under such context.

C. EFFECT OF PRE-TRAINING WORD EMBEDDINGS

In order to test the effect of pre-training word embeddings, we performed experiments using different word embeddings. We randomly initialized word embeddings first and initialized word embeddings using pre-trained ones. Random embeddings are initialized with a uniform distribution ranging between -0.01 and 0.01. Their dimensions are the same as those of pre-trained embeddings. We selected our best model in two corpora namely WCNN+CCNN and WATT+CATT in these comparison experiments. As shown in Figure 5, comparing the green bars with the red bars, pre-training word embeddings can improve the F1s on both corpora by 2.1% and 2.46%, respectively. Comparing the blue bars with the pink bars, pre-training word embeddings achieve significant improvements on both corpora. These show that pre-training word embeddings are useful for disease mention normalization.

TABLE 7. Comparison of different feature ablation (%) on NCBI disease corpus.

| Model         | P    | R    | F1   |
|---------------|------|------|------|
| WCNN+CCNN+SCNN| 85.77| 87.94| 86.31|
| WCNN+CCNN     | 89.97| 91.37| 90.11|
| WCNN+SCNN     | 84.59| 88.58| 85.85|
| CCNN+SCNN     | 66.78| 72.56| 68.44|
| WCNN          | 88.68| 90.68| 89.11|
| CCNN          | 73.76| 80.01| 75.81|
| SCNN          | 66.23| 50.04| 54.10|

D. EFFECT OF PRE-TRAINING MODELS USING THE VOCABULARY

To evaluate the effect of pre-training models using the vocabulary, we carried out comparison experiments between with and without pre-training models. We chose our best-performing models, namely WCNN+CCNN and WATT+CATT, to perform these experiments on both corpora. As shown in Figure 5(b), comparing the green bars with the blue bars, there are dramatic increases (about 9% and 6%) using pre-training models on both corpora. We can also observe that the growth of pre-training models is much bigger than that of pre-training word embeddings.

E. COMPARING CNN, LSTM AND ATTENTION NETWORK

As illustrated in Table 3 and Table 4, we observe that the performance of the LSTM network is the lowest in all settings, and CNN yields generally better results than those of LSTM on both corpora. This is consistent with the discovery in prior work [47]–[49], where the CNN may slightly perform better in the text classification task while the LSTM may be good at some sequence prediction tasks, such as NER. CNN and Attention networks are capable of performing better because they are better at capturing local patterns or key points. This has been also demonstrated in the experimental results, where the CNN network performs the best on the NCBI disease corpus and the Attention Network performs the best on the CDR corpus. Another reason may be that the normalization task are not as sensitive to the order of the words in a name as the words themselves. Thus, the CNN and Attention networks perform better since they are order-independent models.

F. ABLATION STUDY

To evaluate the effects of different features, we carried out ablation study experiments with word, character and sentence features on both corpora. We selected the best-performed
neural network in both corpora to conduct the experiments. As shown in Table 7, the results among WCNN, CCNN and SCNN show that word features are the most effective. In addition, the combination of word and character features (WCNN+CCNN) makes the model perform the best, while adding sentence features (WCNN+CCNN+SCNN) leads to a decrease of the performance, which was verified in Subsection B on Page 8. From Table 8, we observed similar results, which demonstrates the conclusions are consistent in both corpora.

VI. CONCLUSION
We performed a comprehensive investigation of disease name normalization based on a neural network framework, analyzing the effect of word-level, character-level and sentence-level information on the normalization task by employing three widely-used architectures of neural networks, i.e., CNN, LSTM or Attention. We draw three conclusions from our experimental results on two corpora. (i) Word-level feature plays a key role in the disease normalization task, character-level information is beneficial, but sentence-level information do not work well in this task. (ii) CNN and Attention network is superior to the LSTM network in this task. (iii) Both pre-training word embeddings and pre-training models can improve the performance significantly. Our findings can potentially facilitate the research of biomedical text mining and natural language processing.

ACKNOWLEDGMENT
The authors greatly appreciate Yue Zhang for editing the manuscript.

REFERENCES
[1] A. P. Davis, T. C. Wiegers, M. C. Rosenstein, and C. J. Mattingly, “MEDIC: A practical disease vocabulary used at the comparative toxicogenomics database,” Database, vol. 2012, p. bar065, Mar. 2012.
[2] N. Kang, B. Singh, Z. Afzal, E. M. van Mulligen, and J. A. Kors, “Using rule-based natural language processing to improve disease normalization in biomedical text,” J. Amer. Med. Inform. Assoc., vol. 20, no. 5, pp. 876–881, Sep. 2013.
[3] S. Pradhan, N. Elhadad, W. Chapman, S. Manandhar, and G. Savova, “SemEval-2014 task 7: Analysis of clinical text,” in Proc. 8th Int. Workshop Semantic Eval. (SemEval), 2014, pp. 54–62.
[4] R. J. Kate, “Normalizing clinical terms using learned edit distance patterns,” J. Amer. Med. Inform. Assoc., vol. 23, no. 2, pp. 380–386, Mar. 2016.
[5] J. D’Souza and V. Ng, “Sieve-based entity linking for the biomedical domain,” in Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process., vol. 2, 2015, pp. 297–302.
[6] R. Leaman and Z. Lu, “TaggerOne: Joint named entity recognition and normalization with semi-Markov models,” Bioinformatics, vol. 32, no. 18, pp. 2839–2846, Sep. 2016.
[7] Y. Ren, Y. Zhang, M. Zhang, and D. Ji, “Context-sensitive Twitter sentiment classification using neural network,” in Proc. 30th AAAI Conf. Artif. Intell., 2016, pp. 1–7.
[8] Y. Lou, Y. Zhang, T. Qian, F. Li, S. Xiong, and D. Ji, “A transition-based joint model for disease named entity recognition and normalization,” Bioinformatics, vol. 33, no. 15, pp. 2363–2371, Aug. 2017.
[9] Y. Ren, H. Fei, X. Liang, D. Ji, and M. Cheng, “A hybrid neural network model for predicting kidney disease in hypertension patients based on electronic health records,” BMC Med. Inform. Decis. Making, vol. 19, no. S2, p. 51, Apr. 2019.
[10] M. Cheng, L. Li, Y. Ren, Y. Lou, and J. Gao, “A hybrid method to extract clinical information from Chinese electronic medical records,” IEEE Access, vol. 7, pp. 70624–70633, 2019.
[11] J. Shin, S. Wu, F. Wang, C. De Sa, C. Zhang, and C. Ré, “Incremental knowledge base construction using deepdeepe,” Proc. VLDB Endowment, vol. 8, no. 11, pp. 1310–1321, 2015.
[12] H.-W. Chun, Y. Tsuruoka, I.-D. Kim, R. Shiba, N. Nagata, T. Hishiki, and J. I. Tsujii, “Extraction of gene-disease relations from medline using domain dictionaries and machine learning,” in Biocomputing. Singapore: World Scientific, 2006, pp. 4–15.
[13] C.-H. Wei, Y. Peng, R. Leaman, A. P. Davis, C. J. Mattingly, J. Li, T. C. Wiegers, and Z. Lu, “Overview of the BioCreative V disease relation (CDR) task,” in Proc BioCreative Workshop, Seville, Spain, 2015, pp. 154–166.
[14] F. Li, M. Zhang, G. Fu, and D. Ji, “A neural joint model for entity and relation extraction from biomedical text,” BMC Bioinf., vol. 18, no. 1, p. 198, Dec. 2017.
[15] Y. Ren, H. Fei, and D. Ji, “Drug-drug interaction extraction using a span-based neural network model,” in Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM), Nov. 2019, pp. 1237–1239.
[16] R. I. Dogan and Z. Lu, “An inference method for disease name normalization,” in Proc. AAAI Fall Symp., Inf. Retr. Knowl. Discovery Biomed. Text, 2012, pp. 1–6.
[17] R. Leaman, R. I. Dogan, and Z. Lu, “DNorm: Disease name normalization with pairwise learning to rank,” Bioinformatics, vol. 29, no. 22, pp. 2909–2917, Nov. 2013.
[18] H. Cho, W. Choi, and H. Lee, “A method for named entity normalization in biomedical articles: Application to diseases and plants,” BMC Bioinf., vol. 18, no. 1, p. 451, Dec. 2017.
[19] H. Liu and Y. Xu, “A deep learning way for disease name representation and normalization,” in Proc. Nat. CCF Conf. Natural Lang. Process. Chinese Comput. Cham, Switzerland: Springer, 2017, pp. 151–157.
[20] Y. Kim, “Convolutional neural networks for sentence classification,” 2014, arXiv:1408.5882. [Online]. Available: http://arxiv.org/abs/1408.5882.
[21] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.
[22] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” 2015, arXiv:1508.04025. [Online]. Available: http://arxiv.org/abs/1508.04025.
[23] H. Li, Q. Chen, B. Tang, X. Wang, H. Xu, B. Wang, and D. Huang, “CNN-based ranking for biomedical entity normalization,” BMC Bioinf., vol. 18, no. 11, p. 385, Jan. 2017.
[24] R. I. Dogan, R. Leaman, and Z. Lu, “NCBI disease corpus: A resource for disease name recognition and concept normalization,” J. Biomed. Informat., vol. 47, pp. 1–10, Feb. 2014.
[25] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C.-H. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, and Z. Lu, “BioCreative V CDR task corpus: A resource for chemical disease relation extraction,” Database, vol. 2016, p. baw068, Jan. 2016.
[26] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, p. 436, 2015.
[27] P. Deng, H. Chen, M. Huang, X. Ruan, and L. Xu, “An ensemble CNN method for biomedical entity normalization,” in Proc. 5th Workshop BioNLP Open Shared Tasks, 2019, pp. 143–149.
[28] I. Karadeniz and A. Özgür, “Linking entities through an ontology using word embeddings and syntactic re-ranking,” BMC Bioinf., vol. 20, no. 1, p. 1555, Dec. 2019.
[29] J. Niu, Y. Yang, S. Zhang, Z. Sun, and W. Zhang, “Multi-task character-level attentional networks for medical concept normalization,” Neural Process. Lett., vol. 49, no. 3, pp. 1239–1256, Jun. 2019.
[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5998–6008.
[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013, arXiv:1301.3781. [Online]. Available: http://arxiv.org/abs/1301.3781.
[32] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1532–1543.
[33] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” Trans. Assoc. Comput. Linguistics, vol. 5, pp. 135–146, Dec. 2017.
[34] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” Neural Netw., vol. 18, nos. 5–6, pp. 602–610, Jul. 2005.

[35] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980

[36] F. Dernoncourt, J. Y. Lee, O. Uzuner, and P. Szolovits, “De-identification of patient notes with recurrent neural networks,” J. Amer. Med. Inform. Assoc., vol. 24, no. 3, pp. 596–606, 2017.

[37] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., vol. 1, 2018, pp. 2227–2237.

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2019, arXiv:1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805

[39] C. E. Lipscomb, “Medical subject headings (MeSH),” Bull. Med. Library Assoc., vol. 88, no. 3, p. 265, 2000.

[40] A. Hamosh, A. F. Scott, J. S. Amberger, C. A. Bocchini, and V. A. McKusick, “Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders,” Nucleic Acids Res., vol. 33, no. 1, pp. D514–D517, 2005.

[41] E. Loper and S. Bird, “NLTK: The natural language toolkit,” 2002, http://www.nltk.org. [Online]. Available: https://www.nltk.org

[42] S. Sohn, D. C. Comeau, W. Kim, and W. J. Wilbur, “Abbreviation definition identification based on automatic precision estimates,” BMC Bioinf., vol. 9, no. 1, p. 402, Dec. 2008.

[43] T.-Y. Liu, “Learning to rank for information retrieval,” Found. Trends Inf. Retr., vol. 3, no. 3, pp. 225–331, 2007.

[44] Y. Song, Y. He, Q. Hu, and L. He, “ECNU at 2015 CDS track: Two re-ranking methods in medical information retrieval,” in Proc. Text Retr. Conf., 2015, pp. 1–6.

[45] P. Von Däniken and M. Cieliebak, “Transfer learning and sentence level features for named entity recognition on tweets,” in Proc. 3rd Workshop W-NUT, Copenhagen, Denmark, vol. 3, Sep. 2017, pp. 166–171.

[46] P. Liu, S. Chang, X. Huang, J. Tang, and J. C. K. Cheung, “Contextualized non-local neural networks for sequence learning,” in Proc. AAAI Conf. Artif. Intell., vol. 33, Jul. 2019, pp. 6762–6769.

[47] Y. Y. Lee and F. Dernoncourt, “Sequential short-text classification with recurrent and convolutional neural networks,” 2016, arXiv:1603.03827. [Online]. Available: http://arxiv.org/abs/1603.03827

[48] J. Wang, Z. Wang, D. Zhang, and J. Yan, “Combining knowledge with deep convolutional neural networks for short text classification,” in Proc. 26th Int. Joint Conf. Artif. Intell., Aug. 2017, pp. 2915–2921.

[49] J. Zeng, J. Li, Y. Song, C. Gao, M. R. Lyu, and J. King, “Topic memory networks for short text classification,” 2018, arXiv:1809.03664. [Online]. Available: http://arxiv.org/abs/1809.03664

YINXIA LOU received the M.S. degree from Hunan Normal University, Changsha, China. She is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Wuhan University, China. Her research interests include deep learning, biomedical text mining, and natural language processing.

FEI LI received the B.S. degree in computer science and technology from Jilin University, Changchun, in 2005, the M.S. degree in computer application technology from Ningbo University, Ningbo, in 2008, and the Ph.D. degree in computer software and theory from Wuhan University, Wuhan, in 2017. He is currently a Postdoctoral Research Associate with the University of Massachusetts. His research interests include natural language processing, deep learning, and their application in the biomedical domain.

MING CHENG received the Ph.D. degree from the Computer School, Wuhan University, China, in 2016. He is currently doing Postdoctoral Research at The First Affiliated Hospital of Zhengzhou University. His research interests include biomedical natural language processing (NLPI) and healthcare data mining.

TAO QIAN received the Ph.D. degree from the Computer School, Wuhan University, China, in 2016. He is currently an Associate Professor with the Hubei University of Science and Technology, China. His current research interests include natural language processing and sentiment analysis.

JUNXIANG ZHOU received the B.S. degree from Zhengzhou University, Zhengzhou, China, and the M.S. degree from the China University of Mining and Technology. He is currently a Lecturer with Shangqiu Normal University, China. His research interests include data mining and machine learning.

DONGHONG JI received the B.S., M.S., and Ph.D. degrees from the Computer School, Wuhan University, China, in 1988, 1991, and 1995, respectively. He was a Postdoctoral Research Fellow with Tsinghua University, from 1995 to 1998. From 1998 to 2008, he was a Research Scientist with the Institute for Infocomm Research, Singapore. He is currently a Professor and the Ph.D. Supervisor with the School of Cyber Science and Engineering, Wuhan University. His research interests include natural language processing, machine learning, and data mining.

***