

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/319404236>

# A Bidirectional LSTM and Conditional Random Fields Approach to Medical Named Entity Recognition

Conference Paper · January 2018

DOI: 10.1007/978-3-319-64861-3\_33

---

CITATIONS

18

---

READS

3,410

4 authors, including:



Tianyong Hao

South China Normal University

102 PUBLICATIONS 625 CITATIONS

SEE PROFILE

# A Bidirectional LSTM and Conditional Random Fields Approach to Medical Named Entity Recognition

Kai Xu<sup>1</sup>, Zhanfan Zhou<sup>2</sup>, Tianyong Hao<sup>2(✉)</sup>, and Wenyin Liu<sup>1(✉)</sup>

<sup>1</sup> School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China

504087493@qq.com, liuwy@gdut.edu.cn

<sup>2</sup> School of Informatics, Guangdong Universities of Foreign Studies, Guangzhou, China  
a\_fanky@163.com, haoty@126.com

**Abstract.** Medical named entity recognition is a fundamental and essential research for medical natural language processing, aiming to identifying medical concepts or terminology such as diseases, drugs, treatments, procedures, etc. from unstructured medical text. A model based on a bidirectional LSTM and conditional random fields (Bi-LSTM-CRF) is proposed for medical named entity recognition. Our model contains three layers and relies on character-based word representations learned from the supervised corpus. BiLSTM-CRF model can learn the information characteristics of a given dataset. Experiments on a publicly available NCBI Disease Corpus as an evaluation standard dataset shows our approach achieves a 0.8022 F1 measure, which outperforms a number of widely used baseline methods.

**Keywords:** Medical NER · LSTM · CRF · Deep learning

## 1 Introduction

Natural Language Processing (NLP) has been widely applied in the medical domain and become an active research area in medical informatics [1, 7]. Many NLP studies have successfully demonstrated its applications in clinical practice and research [2, 5]. As a fundamental research in NLP, Named Entity Recognition (NER) has become a hot research area in recent years. NER is the first step in text processing since a free-text document usually consists of a number of named entities. Differencing from other domains, medical text frequently contains a large amount of named entities (NE) such as medicines, diseases, treatments, symptoms, etc. [3]. An NE may be represented in different forms due to various reasons, e.g., the usage habits differ from clinicians to clinicians. Even more, term abbreviations are commonly used in the medical domain and the same abbreviation may refer to different mentions [3]. Consequently, the medical NER is still a challenging research topic in medical text processing.

There are a number of existing researches on NER over medical texts. DNORM [2, 4] was the first claimed machine learning method to normalize disease mentions in biomedical text. It directly calculated the similarity between medical mentions and concepts

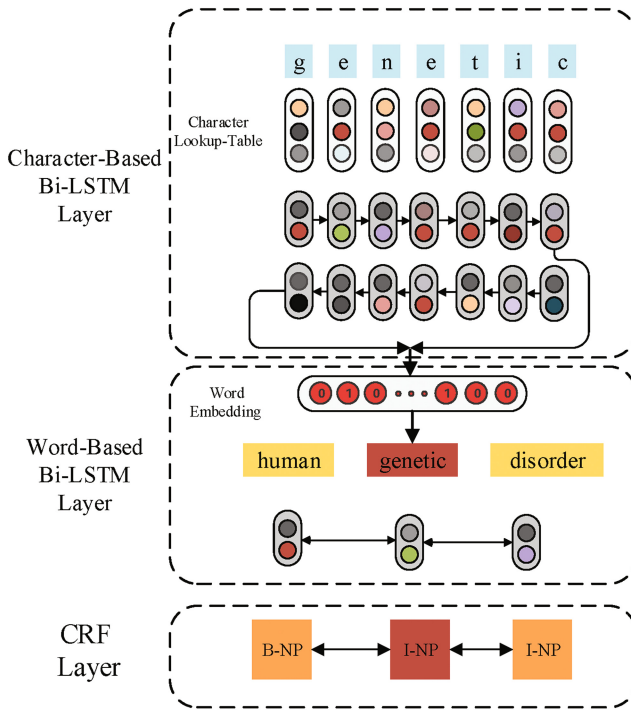
defined in medical dictionaries, e.g., UMLS. It also tried to learn arbitrary mappings between mentions and concepts, including synonymy, polysemy and non one-to-one relationships. Jain [6] presented a supervised system for clinical named entity recognition based on CRF method. Neveol [8] developed a method to extract ten types of medical named entities, e.g., disorders, from biomedical text in French. The method applied the semantic network defined in UMLS to identify the semantic types of extracted entities. Chen et al. [9] conducted a study to simulate an active learning experiment using different algorithms to illustrate performance of the NER model against the annotation cost. However, the reported NER performance in terms of precision, recall, and F1 were not high enough for practical usage. For example, the highest reported F1 was 86.76% by HITSZ\_CDR [10].

On the other hand, there are medical dictionaries containing predefined formal concepts, which are widely applied for medical named entity recognition. Among the dictionaries, Unified Medical Language System (UMLS) integrates and distributes key terminology, classification and coding standards, and associated resources to promote creation of more effective and interoperable biomedical information systems and services. The UMLS has three Knowledge Sources: (1) Metathesaurus: Terms and codes from many vocabularies, (2) Semantic Network, and (3) SPECIALIST Lexicon and Lexical Tools for Natural language processing [11].

Targeting at improving the performance of medical NER, we propose a method named as Bi-LSTM-CRF, which is a bidirectional Long Short-Term Memory Network (LSTM) with a sequential Conditional Random Field (CRF). The method contains three layers: a character-based Bi-LSTM layer, a word-based Bi-LSTM layer, and a CRF layer. In order to capture word sensitivity and distributional sensitivity, a character-based word representation model is applied. To evaluate the performance of the propose method, we use a publically available NCBI Disease Corpus as our dataset, which is a standard dataset for NER tasks and has been used in some existing works such as [4]. The NCBI Disease corpus consists of 793 PubMed medical literature abstracts and has been split into three subsets: Training subset, Test Subset, and Development Subset. We conduct three experiments and compared with a list of baseline methods including CRF, BiLSTM, NCBI, UMLS, CMT, and MeSH. The results present that our method outperforms the baseline methods, indicating its effectiveness of the method.

## 2 The Bi-LSTM-CRF Model

In this section, we present a bidirectional LSTM with a sequential conditional random field method as Bi-LSTM-CRF. With respect to the architecture defined in [12, 13], our method consists of three layers. The first layer is a character-based Bi-LSTM layer designed to learn character-level expressions of the words in a given text. The second layer is a word-based Bi-LSTM layer which combines a word lookup table into multiple word-based Bi-LSTM networks in order to obtain word embeddings. The third layer is a CRF layer which captures the relations among labels extracted from a CRF model, e.g., label *B-LOC* cannot follow label *I-PER* in a sentence. The overall architecture of the Bi-LSTM-CRF method is shown in Fig. 1, where the data flow is from top to bottom.



**Fig. 1.** The overall architecture of the Bi-LSTM-CRF Model for medical NER tasks

### (1) The Character-Based Bi-LSTM Layer

Given a text consisting of a list of words, the prefix and suffix information for each word can be extracted as character-level word elements [14, 18, 19, 21, 23]. Since medical texts commonly contain medical words and phrases, the identification of word prefix and suffix information is not easy. Considering the existing evidences by using the deep representation learning [15], we apply a deep learning model to obtain the characteristics of medical texts by constructing word embeddings.

These character-level representations are then concatenated together to be a word-level representation according to a word lookup-table. The advantage of learning character-level embedding is that it can capture content-specific features, which are valuable for morphologically rich languages and to deal with the out-of-vocabulary issue for certain tasks such as grammatical labeling and language demonstration [16].

Since bidirectional LSTM can learn complex non-nearby sequence conditions, we use a character to word (C2W) model composed of bidirectional LSTMs. The input of C2W is a word  $w$  for a given medical text. The word is then expressed in a  $d$ -dimensional vector. The C2W model uses the word lookup-table for processing input and output words, thus the model can be easily replaced in other types of networks.

A word is split into a characters set  $C$  containing a list of characters  $c_1, c_2, \dots, c_m$ . Each character  $c_i$  is then converted into a one-hot vector, with each character  $c_i$  is generated from a complete character set  $M$  containing letters, numbers and punctuations. The

$M$  is generated on the whole training dataset and is used as character lookup tables for catching similarity among characters in the linguistics, e.g., vowels, consonants, affix, prefix, etc. We initialize a character lookup table randomly which contains an embedding for each character. The character embeddings associated with each character are sorted in the direct and reverse orders, and then are put into two forward and backward LSTM networks. The forward and backward representations from a bidirectional LSTM constitute a word embedding. Afterwards, the character-level representations are matched with the lookup table of the current word. If it is not in the lookup table, the character is mapped to an unknown word token (UNK) embedding. We empirically set UNK embedding with a probability value 0.5 during the UNK embedding training. A LSTM recurrent model is fit for encoding particularly for long sequences, while, it have a representation one-sided towards their latest inputs [12]. Therefore, we make the last representation of the forward and the backward LSTMs to be an exact representation of word suffix, and a better representation of the word prefix [16].

In this manner, given a character expressions  $e_{c_1}^C, \dots, e_{c_m}^C$  as input, the forward LSTM and the backward LSTM generates sequences  $s_0^f, \dots, s_m^f$  and  $s_0^r, \dots, s_m^r$ , respectively. The expression of the word  $w$  thus is acquired by joining the sequences as  $\{(s_0^f, \dots, s_m^f), (s_0^r, \dots, s_m^r)\}$ .

## (2) The Word-Based Bi-LSTM Layer

Recurrent Neural Network (RNN) is commonly applied on sequential text annotation tasks. A RNN network takes a group of vectors  $(x_1, x_2, \dots, x_n)$  as input and return a group of vectors  $(y_1, y_2, \dots, y_n)$  as output. Theoretically, RNN can learn the content of long text but it may has the issue that the long-distance information is easy to be forgot. Long Short-Term Memory Networks (LSTMs) enhance the capability of information preservation. It can capture long-distance information according to a few gates, which control the contribution of each memory unit. In this paper, we utilize the following equation, where  $\sigma$  and  $\odot$  are the element-wise sigmoid function product calculation and element-wise product calculation, respectively.

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\ c_t &= (1 - i_t) \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o), \quad h_t = o_t \odot \tanh(c_t) \end{aligned} \quad (1)$$

A sentence  $X$  contain a number of words, which can be represented as a group of  $d$ -dimensional vectors  $(x_1, x_2, \dots, x_n)$ .  $\vec{y}_t$  can be calculated by capturing the left-side of the word  $t$  from LSTMs. After the calculation, the right-side of the word  $t$  is acquired and represented as  $\bar{y}_t$ . The representation procedure can be accomplished by generating the same backward sequences in LSTM. We thus define the first as forward LSTM and the second as the backward LSTM. The two types of LSTM networks has two sets of parameters. This pair of forward and backward LSTM is regarded as a bidirectional LSTM, which can combine the representations of left-side and right-side calculations as  $y_t = [\vec{y}_t, \bar{y}_t]$ . The representations contain the context information of the word  $t$  thus is informative for further label classification usage.

### (3) The CRF Layer

An easy labeling model is to utilize  $\vec{y}_i$  as elements to deal with independent label output problem. The language structure of medical NER expressions can be learned using Part-of-Speech (POS) tagging. For example, a label *B-LOC* is not followed by *I-PER*. We use a Conditional Random Fields model to label the POS tags.

For an input sentence  $X = (x_1, x_2, \dots, x_n)$ ,  $P$  denotes an output matrix from a bidirectional LSTM network. The size of  $P$  is  $n \times k$  and the number of labels is  $k$ . The probability of the  $i^{th}$  word being assigned with the label  $j^{th}$  is calculated as  $P_{ij}$ . We therefore define the sequence of expectations as  $Y = (y_1, y_2, \dots, y_n)$ . We then set  $A$  as the transition matrix of probability  $P$  and  $A_{ij}$  is the matrix of transition probability from label  $i$  to  $j$ . In the sentence,  $y_0$  as start label and  $y_n$  as end label are added into the sentence for marking the location of the label sequence. The label probability matrix thus can be calculated using the Eq. (2).

$$s(X, y) = \sum_{i=0}^n A_{y_i, y_{i+1}} + \sum_{i=1}^n P_{i, y_i} \quad (2)$$

For each  $y$ , we use a softmax function to calculate the probability over all possible label sequences as follows:

$$p(y|X) = \frac{e^{s(X, y)}}{\sum e^{s(X, y)}} \quad (3)$$

After that, the maximal log probability of correct label sequence is obtained using Eq. (4), where  $p(y|X)$  denotes all possible label sequences in the sentence  $X$ .

$$\begin{aligned} \log(p(y|X)) &= s(X, y) - \log(\sum e^{s(X, y)}) \\ &= s(X, y) - \log \text{add}(s(X, y)) \end{aligned} \quad (4)$$

In Eq. (4),  $\log \text{add}$  represents logarithmic summation, while  $p = p_1 + p_2$  calculates  $\ln p$  using Eq. (5).

$$\ln p = \ln(p_1 + p_2) = \ln(e^{\ln p_1} \times (1 + e^{\ln p_2 - \ln p_1})) = \ln p_1 + \ln(1 + e^{\ln p_2 - \ln p_1}) \quad (5)$$

Finally, we maximize the probability as  $y^*$  to predict potential label sequences, as shown in Eq. (6). Equations (4), (6) are computed by dynamic programming.

$$y^* = \arg \max s(X, y) \quad (6)$$

## 3 Experiments

### (1) Data Sets

A publicly available dataset NCBI Disease Corpus [17, 23] is used. It was annotated and developed by a research group in National Center for Biotechnology Information (NCBI) and National Institute of Health (NIH) and has been used in many medical NER

tasks or publications, such as in [19–24]. The corpus contains 793 PubMed abstracts and annotated PubMed citations, constituting more than 6000 sentences and 2136 unique disease mentions. The corpus provides manually annotated texts to enable the development of NLP tools to address disease recognition and normalization. As a benchmark dataset, the corpus consists of three sub-datasets: a training dataset (593 texts), a development dataset (100 texts), and a test dataset (100 texts). The details of the NCBI Disease Corpus are listed in Table 1.

**Table 1.** The Overview of NCBI Corpus NCBI

Types	Training data	Development data	Testing data	Total
PubMed Citations	593	100	100	793
Total Disease Mentions	5145	787	960	6892
Unique Disease Mentions	1710	368	427	2136

### (2) Baseline Methods

We compared our method against several baseline methods. The first baseline is a look-up method using medical dictionary [14], which uses terms and vocabularies to perform exact matching. The second baseline is MetaMap, which is the state-of-the-art natural language processing tool for identifying UMLS Metathesaurus concepts in biomedical text. MetaMap identifies possible mappings based on lexical look-up to identify all UMLS concepts [25]. There are two types of identifiers for filtering: semantic type and MEDIC, where the former defines that only concepts associated with certain semantic types are kept and latter defines that only concepts in MEDIC are kept [26]. The third base is an inference method designed and refined on a manually annotated set of PubMed abstract sentences, which reflected the consensus annotation agreement of the AZDC disease corpus [11] and the EBI disease corpus [14]. The other baselines are the CRF combined with certain medical vocabularies, e.g., Medical Subject Headings (MeSH).

### (3) Parameter Tuning

We use Precision, Recall, and F1 score as our evaluation metrics, which are commonly used in NER tasks. During CRF model training, two parameters are used:  $c$  stands for the hyper-parameter related to fitting rate and  $f$  denotes the cut-off threshold in feature selection. In order to find best  $c$  and  $f$ , we first keep  $f$  as 1 to test NER performance changes using different  $c$  values ranging from 1 to 4. The result is shown in Table 2, where the performance in term of F1 is highest as 0.744 when  $c$  value equals to 4. Similarly, we fix the  $c$  to 4 to find the optimized value of  $f$ . The final optimized values are  $c = 1.5$  and  $f = 3$ , respectively.

During LSTM network training, a list of parameters needs to be set. As shown in the Table 3, all the parameters and their settings with descriptions are presented. By adjusting the parameters to observe network output, the optimized values for the parameters are obtained.

**Table 2.** The parameter optimization for CRF training

Method	Precision	Recall	F1
CRF( $c = 1, f = 1$ )	0.869	0.622	0.726
CRF( $c = 1.5, f = 1$ )	0.871	0.634	0.734
CRF( $c = 2.5, f = 1$ )	0.872	0.642	0.739
CRF( $c = 3.5, f = 1$ )	0.874	0.648	0.744
CRF( $c = 4, f = 1$ )	0.873	0.649	0.744
CRF( $c = 4.5, f = 1$ )	0.870	0.649	0.743
CRF( $c = 1.5, f = 2$ )	0.861	0.649	0.740
CRF( $c = 1.5, f = 3$ )	0.855	0.6596	0.745
CRF( $c = 1.5, f = 4$ )	0.838	0.659	0.738

**Table 3.** LSTM network parameters and their optimized settings

Parameter	Setting	Description
Char_dim	25	Character embedding dimension
Char_LSTM_dim	25	Character LSTM hidden layer size
Char_bidirect	TRUE	Use bidirectional LSTMs for Characters
Word_dim	100	Token embedding dimension
Word_LSTM_dim	100	Token LSTM hidden layer size
Word_bidirectional	TRUE	Use a bidirectional LSTM for words
CRF	TRUE	Use CRF
Dropout	1	Input droupout
Lr_method	SGD	Learning method (SGD Adadelta Adam)

(4) Result

To test the effectiveness of our Bi-LSTM-CRF method, we firstly try to view the performance of our method with optimized CRF using different LSTM settings. From the result as shown in Table 4, the performance in terms of both precision and recall are decreased without the use of character-based vector representation, indicating the effectiveness of the first layer. In addition, the performance is dramatically decreased with F1 value dropping from 80.22 to 56.88 without CRF layer. The F1 is decreased from 80.22 to 77.74 without using Bi-LSTM.

**Table 4.** The effect of different parameter settings and the final optimized result

Parameter	Value	Precision	Recall	F1
Character	Disable	81.56	72.36	76.69
CRF	Disable	55.91	57.88	56.88
Bi-LSTM	Disable	82.45	73.53	77.74
All	Optimized	84.80	76.12	80.22

Moreover, we test the performance with different word embedding dimensions, as shown in Table 5. By setting dimensions from 50 to 200, the maximum performance

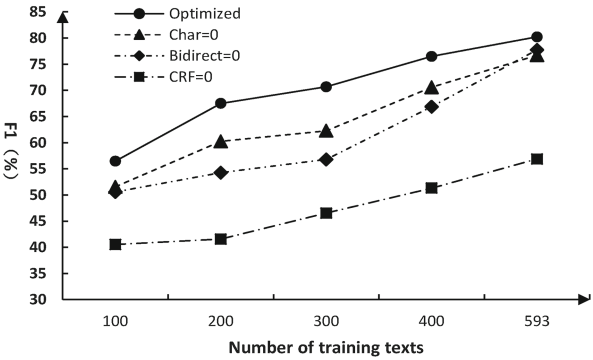


value is 78.95 when the dimension equals to 100. After that, we test the Bi-LSTM network dimension using different dimensions. The result presents that the optimized Bi-LSTM network dimension is 100, where its F1 value is 80.22 in the experiment.

**Table 5.** The effect of parameter settings of word embedding and Bi-LSTM dimensions

	Dimensions	Precision	Recall	F1
Word embedding	50	81.62	73.65	77.43
	100	83.40	74.95	78.95
	150	85.86	68.59	76.26
	200	82.38	74.82	78.42
Bi-LSTM	50	80.18	73.76	76.84
	100	84.80	76.12	80.22
	150	83.84	68.35	75.31
	200	84.80	70.24	76.83

In order to verify the robustness of our Bi-LSTM-CRF method, we use different numbers of texts from the training dataset. The performance in terms of F1 value is evaluated and presented in Fig. 2. With different number of texts, the performances using four parameters are compared with each other, where the parameters are from Table 3 with “*Char* = 0” denotes character embedding disabled, “*Bidirect* = 0” denotes bidirectional LSTM disabled, and “*CRF* = 0” denotes CRF layer disabled. From the result, the performance in terms of F1 value is increasing when using more training texts and the “*Bidirect* = 0” (bidirectional LSTM disabled) is more sensitive to training dataset size, indicting the potential importance of bidirectional LSTM.



**Fig. 2.** The performance of Bi-LSTM-CRF using different numbers of training texts

Finally, we compare the proposed method with the widely used NER methods as baselines, i.e., CRF, Bi-LSTM, NCBI, UMLS, CMT, and MeSH. The comparison result is shown in Table 6. From the result, dictionary look-up obtains the worst F1 as 31.60, while MetaMap with two filtering strategies are much better, with F1 increased to 54.10 and 55.90. CRF with UMLS, CMT, and MeSH has similar performance with F1 ranging

from 73.47 to 75.62. Our Bi-LSTM-CRF achieves the best performance on Recall and F1, with the overall F1 as 80.22. Our Bi-LSTM-CRF exceeds MetaMap (semantic type filtering) with more than 30.27% F1 increase and MetaMap (MEDIC filtering) with more than 16.53% F1 increase. The result demonstrates that our Bi-LSTM-CRF method outperforms all the compared baseline methods, indicating that it is applicable for medical named entity recognition.

**Table 6.** The performance comparison with the baseline methods

Methods	Precision	Recall	F1
Dictionary look-up	21.30	71.80	31.60
MetaMap (semantic type filtering)	49.50	67.90	54.10
MetaMap (MEDIC filtering)	51.00	70.20	55.90
Inference method	59.70	73.10	63.70
CRF + UMLS	83.90	68.80	75.62
CRF + CMT	79.50	68.30	73.47
CRF + MeSH	85.50	66.00	74.55
Our Bi-LSTM-CRF	84.80	76.12	80.22

## 4 Conclusion

This paper presents a bidirectional LSTM with a sequential conditional random field method as Bi-LSTM-CRF. The method has a new architecture containing three layers. With a pre-defined CRF layer, the method can capture character-based representations for improving medical named entity recognition. Based on a standard evaluation dataset NCBI Disease Corpus, we test the performance of the proposed Bi-LSTM-CRF method by comparing with the commonly used baseline methods including CRF, BiLSTM, NCBI, UMLS, CMT, and MeSH. The experimental results shows that our method outperforms all the baselines, demonstrating its effectiveness.

**Acknowledgement.** This work was supported by National Natural Science Foundation of China (No. 61403088), the programs of Personalized Health Service Public Platform based on Open and Big Data (No. 2014B010118005), Ancient Literature Knowledge base Platform for the Inheritance and Development of Traditional Chinese Medicine (No. 2014A020221039) and Innovative School Project in Higher Education of Guangdong (No. YQ2015062).

## References

1. Lei, J., Tang, B., Lu, X.: A comprehensive study of named entity recognition in Chinese clinical text. *J. Am. Med. Inform. Assoc.* **21**(5), 808–814 (2013)
2. Leaman, R., Islamaj, D.R., Lu, Z.: DNorm: disease name normalization with pairwise learning to rank. *Bioinformatics* **29**(22), 2909–2917 (2013)

3. Islamaj, D.R., Lu, Z.: An improved corpus of disease mentions in PubMed citations. In: Proceedings of the ACL 2012 Workshop on BioNLP, pp. 91–99 (2012)
4. Jain, D.: Supervised named entity recognition for clinical data. In: CLEF 2015 Online Working Notes, CEUR-WS 2015, vol. 28 (2015)
5. Alanazi, S., Sharp, B., Stanier, C.: A named entity recognition system applied to Arabic text in the medical domain. *Int. J. Comput. Sci.* **12**(3), 109–117 (2015)
6. Chen, Y., Lasko, T.A., Mei, Q., Denny, J.C., Xu, H.: A study of active learning methods for named entity recognition in clinical text. *J. Biomed. Inform.* **58**(C), 11–18 (2015)
7. Leaman, R., Gonzalez, G.: Banner: an executable survey of advances in biomedical named entity recognition. *Pac. Symp. Biocomput.* **13**, 652–663 (2008)
8. Wang, H., Zhang, W., Zeng, Q., Li, Z., Feng, K., Liu, L.: Extracting important information from Chinese operation notes with natural language processing methods. *J. Biomed. Inform.* **48**(2), 130–136 (2014)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
10. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **18**(5–6), 602–610 (2005)
11. Unified Medical Language System (UMLS), NIH. <https://www.nlm.nih.gov/research/umls/quickstart.html>. Accessed 2017
12. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), pp. 260–270 (2016)
13. Leaman, R., Lu, Z.: Taggerone: joint named entity recognition and normalization with semi-markov models. *Bioinformatics* **32**(18), 343 (2016)
14. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM. *Neural Comput.* **12**(10), 2451–2471 (2000)
15. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(8), 1798–1828 (2013)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *Computer Science* (2013)
17. Islamaj, D.R., Leaman, R., Lu, Z.Y.: NCBI disease corpus: a resource for disease name recognition and concept normalization. *J. Biomed. Inform.* **47**(2), 1–10 (2014)
18. Ling, W., Tsvetkov, Y., Amir, S., Fernandez, R., Dyer, C., Black, A.W., et al.: Not all contexts are created equal: better word representations with variable attention. In: The Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1367–1372 (2015)
19. Campos, D., Matos, S., Oliveira, J.L.: A modular framework for biomedical concept recognition. *BMC Bioinform.* **14**, 281 (2013)
20. Ling, W., Luís, T., Marujo, L., Astudillo, R.F., Amir, S., Dyer, C.: Finding function in form: compositional character models for open vocabulary word representation. *Computer Science*, 1899–1907 (2015)
21. Song, M., Kim, W.C., Lee, D., Heo, G.E., Kang, K.Y.: PKDE4J: entity and relation extraction for public knowledge discovery. *J. Biomed. Inform.* **57**, 320–332 (2015)
22. Good, B.M., Nanis, M., Wu, C., Su, A.I.: Microtask crowdsourcing for disease mention annotation in pubmed abstracts. *Pac. Symp. Biocomput.* **509–510**, 282 (2015)
23. Comeau, D.C., Liu, H., Doğan, R.I., Wilbur, W.J.: Natural language processing pipelines to annotate BioC collections with an application to the NCBI disease corpus. *Databases* **23**, 3790–3795 (2014)

24. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. *Computer Science*, [arXiv:1508.01991](#) (2015)
25. Kim, S., Yeganova, L., Wilbur, W.J.: Meshable: searching pubmed abstracts by utilizing mesh and mesh-derived topical terms. *Bioinformatics* **19**, 331 (2016)
26. Wei, Q., Tao, C., Xu, R., He, Y., Lin, G.: Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks. *Databases* **2016**, 1–8 (2016)