BMC Bioinformatics

RESEARCH Open Access

A hybrid deep learning framework for bacterial named entity recognition with domain features

Xusheng Li[1,2], Chengcheng Fu[1,2], Ran Zhong[3], Duo Zhong[1,2], Tingting He[1,2] and Xingpeng Jiang[1,2*]

From IEEE International Conference on Bioinformatics and Biomedicine 2018
Madrid, Spain. 3-6 December 2018

Abstract

Background: Microbes have been shown to play a crucial role in various ecosystems. Many human diseases have been proved to be associated with bacteria, so it is essential to extract the interaction between bacteria for medical research and application. At the same time, many bacterial interactions with certain experimental evidences have been reported in biomedical literature. Integrating this knowledge into a database or knowledge graph could accelerate the progress of biomedical research. A crucial and necessary step in interaction extraction (IE) is named entity recognition (NER). However, due to the specificity of bacterial naming, there are still challenges in bacterial named entity recognition.

Results: In this paper, we propose a novel method for bacterial named entity recognition, which integrates domain features into a deep learning framework combining bidirectional long short-term memory network and convolutional neural network. When domain features are not added, F1-measure of the model achieves 89.14%. After part-of-speech (POS) features and dictionary features are added, F1-measure of the model achieves 89.7%. Hence, our model achieves an advanced performance in bacterial NER with the domain features.

Conclusions: We propose an efficient method for bacterial named entity recognition which combines domain features and deep learning models. Compared with the previous methods, the effect of our model has been improved. At the same time, the process of complex manual extraction and feature design are significantly reduced.

Keywords: Named entity recognition, Biomedical text mining, Conditional random field, Deep learning

Background

Microorganisms are ubiquitous in nature. Human beings are exposed to microorganisms from birth to death and are associated with microorganisms during all stages of life. The human body together with its microbiome constitutes a super-species, forming our own exclusive microbial community [1]. Studies have shown that microbial diversity is associated with various human diseases, including allergy, diabetes, obesity, arthritis, inflammatory bowel disease, and even neuropsychiatric diseases [2–4]. Therefore, the diversity of microbial communities and the interaction between microorganisms and the host immune system play crucial role in guaranteeing human healthy. Microorganisms in microbial communities interact with other members actively which ensures the stability and diversity of microbial communities [5]. Thus it is important to explore the microbial interaction for understanding the structure of microbial community and applying these results to the biomedical field. In the past, the method of extracting microbial relationships traditionally is to culture bacteria separately in biological laboratory. However, most microbes cannot be cultured experimentally as well as it is time-consuming and expensive. Recently, computational approaches can alleviate above problems to some

* Correspondence: xpjiang@mail.ccnu.edu.cn
[1]School of Computer, Central China Normal University, Wuhan, Hubei, China
[2]Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning, Central China Normal University, Wuhan, Hubei, China
Full list of author information is available at the end of the article

extent thanks to the development of high-throughput sequencing technologies. At present, there are several kinds of computational methods for this task including exploring microbial interactions from metagenomic data, inferring microbial interaction from genomic information and mining microbial interaction from biomedical literature [5]. The two former computational approaches are widely explored; however, extracting the microbial interaction from the biomedical literature is less popular. There are rich relevant researches published in the literature confirming certain microbial interactions through direct experiments. It will be a valuable resource to explore the microbial interaction by mining biomedical literatures and integrate these knowledge into a database or knowledge graph. Nevertheless, the rapid growth in the volume of biomedical literature and the variety of microorganisms make manual interaction extraction barely possible.

In previous work, Freilich [6] proposed a microbial interaction extraction method based on the co-occurrence model. They first extracted the species names from the intestinal microbial abundance data. Then, they retrieved articles with the two species in PubMed and calculated the co-occurrence probability of the species. Finally, a microbial co-occurrence network was constructed to predict microbial interaction. Similarly, Lim [7] obtained the data in the same way and put forward an automated microbial interaction extraction method based on support vector machine (SVM). What they had in common was the process to get the species from microbial abundance data of the human gut, which might result in the omission of certain potential interactions due to the different standards of spelling species names.

In recent years, with the development of natural language processing (NLP), text mining strategy makes it possible to extract microbial interaction from unstructured texts. Furthermore, named entity recognition (NER) is the core task of interaction extraction (IE). The purpose of NER is to extract words with special meaning from the text, such as *Person*, *Location*. Various methods about NER have been proposed as the advancement of computer technology, which are mainly based on following three categories: (1) rule-based method [8]; (2) machine learning-based method [9], 3) neural network-based method [10]. It is not portable and universal that rule-based way needs to design rules in specific domain with experts. The second approach based on statistical machine learning has strong portability and excellent performance, but it requires complex feature engineering and large-scale labeling. Furthermore, neural network based method has the highlighting performance without cumbersome process of feature design as well as large-scale tagging data. Although the method of NER in the general domain has fully developed, it is a challenging task in the domain of bacterial name identification on account of complexity of microbial names.

Wang [11, 12] proposed a method of bacterial named entity recognition based on conditional random fields (CRF) and dictionary, which contains more than 40 features (word features, prefixes, suffixes, POS, etc.). The model effect was optimized after selecting the best combinations of 35 features, in the meanwhile, the computing efficiency of this model was greatly improved by deploying the model on Spark platform. Unfortunately, CRF and dictionary-based method need manually design features and additionally dictionary resources, and the result of the model depend on the quality of the annotated data and the rationality of the feature design.

In the last few years, deep learning has been widely utilized and has achieved great performance in many fields, such as image [13]; speech recognition [14]; machine translation [15]; reading comprehension [16] and so on. Similarly, the method based on deep learning has attracted extensive attention in the field of NER. Lample [17] first adopted Bi-LSTM -CRF for NER, Ma [18] introduced Bi-LSTM-CNN- CRF for NER, in which CNN was used to extract character-level features. Since then, more and more deep learning algorithms are used for NER. Also, the biomedical text mining contest was organized to accelerate the research on biomedical [19, 20], and many of top participating systems utilized deep learning in biomedical text [10, 21]. Li [22] shown that deep learning-based method could acquire well performance in bacterial NER. However, his work did not take advantage of the existing biological resources and incorporate them as features into the model.

In this paper, we propose a method combining domain features and deep learning for bacterial NER, which achieves excellent performance in dataset. When adopting POS features only, the F1-measure of the model reaches 89.4%. With POS features and dictionary features are both added, the F1-measure is up to 89.7%. The experimental results demonstrate that external resources can contribute to the improvement of the result of the model.

Materials and methods

As shown in Fig.1, we build a model mainly divided into the following three layers: embedding layer, encoding layer and decoding layer. Firstly, we concatenate pretrained word embedding, character-level embedding extracted by convolution neural network, POS embedding and dictionary embedding and input it into the encoding layer. Then the encoding layer is used for parameter learning. In the end, we can predict the best output path of sentence through the decoding layer.

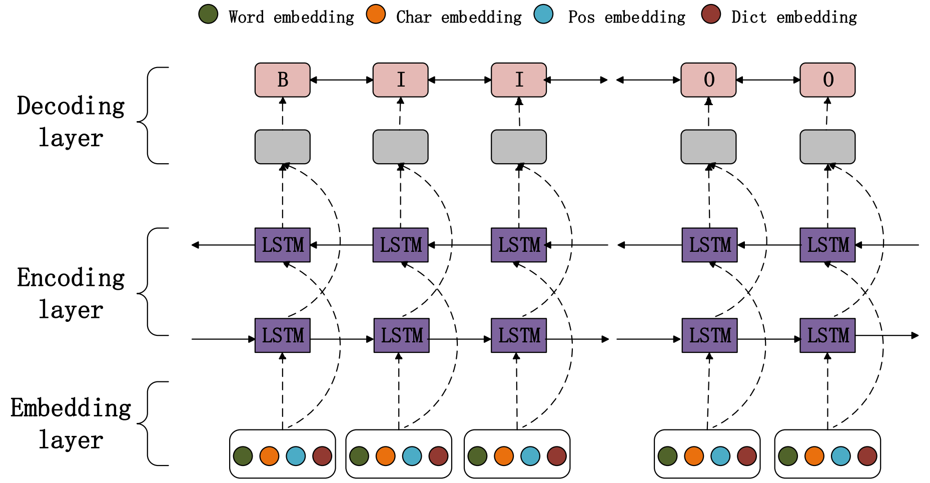

Fig. 1 The model proposed in this paper. The concatenated word-level embedding, char-level embedding, pos embedding and dict embedding are input into encoding layer for learning, then the output of encoding layer are input into decoding layer for predict

Embedding layer

Word embedding

According to a recent study, word embedding has achieved outstanding results in the field of NLP. Compared with the traditional encoding method, the word embedding technique can fully exploit semantic information between words, for example “king” – “man” + “ woman” = “queen”, as well as using a low-dimensional continuous vector to represent the vector of words. This not only solves the sparse problem of the vector, but also obtains semantic information of the word. Currently, there are some well-performed word embedding tools which are widely used, such as fastText [23], glove [24], Word2vec [25]. At the same time, Moen [26] pre-trained a word embedding PubMed2vec with word2vec in the field of biomedical text mining. In our work, in order to obtain higher quality of word vectors, we downloaded more than 400 thousand abstracts about bacteria from PubMed and then used them together with our corpus to train word vectors. We adopted the skip-gram model of word2vec provided in gensim [27] to train our corpus.

Char embedding

As shown by previous studies, character-level features have been proved to be work well in many NLP tasks. Kim [28] used CNN to obtain character representation and then utilized LSTM to train a language model. Santos and Chiu [29] showed that CNN could extract word morphological features (prefix and suffix etc.) effectively and encoded them into neural network. Lample [17] also demonstrated that LSTM could extract morphological features of words. But, experiment results show that CNN is better than LSTM in the task of NER. As a consequence, in this paper, we use the CNN to obtain the character-level features of words. Figure 2 illustrates detailed process of our method. Given a word W= $[c_i]_0^T$, T is the length of sequence, c_i represents the character of the word, e(c_i) is the character vector for each character. In order to acquire morphological features of words, we use N times of convolution kernels X to perform convolution operations. The size of convolution kernels is k. The calculation formula of O_i output for each convolution can be written as:

$$O_i = \mathrm{relu}(W_1 X_i + b_1) \tag{1}$$

Where W_1 denote the weight matrix and b_1 denote the bias vector, $X_i = [e(c_{i-k}), \ldots, e(c_i), \ldots, e(c_{i+k})]$, relu denote the activation function. Finally, for each convolution kernel output O_1, …, O_i, …, O_N, the max-pooling operation is performed to obtain the character vector representation of the word. The j-th vector representing W_j can be computed as:

$$W_j = \max_{1 \le i \le N} O_{i_j} \tag{2}$$

Domain features

Inspired by the related work of Chiu [29] and Huang [30], some artificial designed features and domain knowledge can also promote the effectiveness of the neural network model. Consequently, in this paper, we discuss the influence of POS and dictionary features on the neural network model.

In fact, although the model of neural network can extract feature automatically to some extent, some linguistic features cannot be well learned on account of the complicacy of natural language processing. We use the nltk [31] tool to get the POS features of each word, and

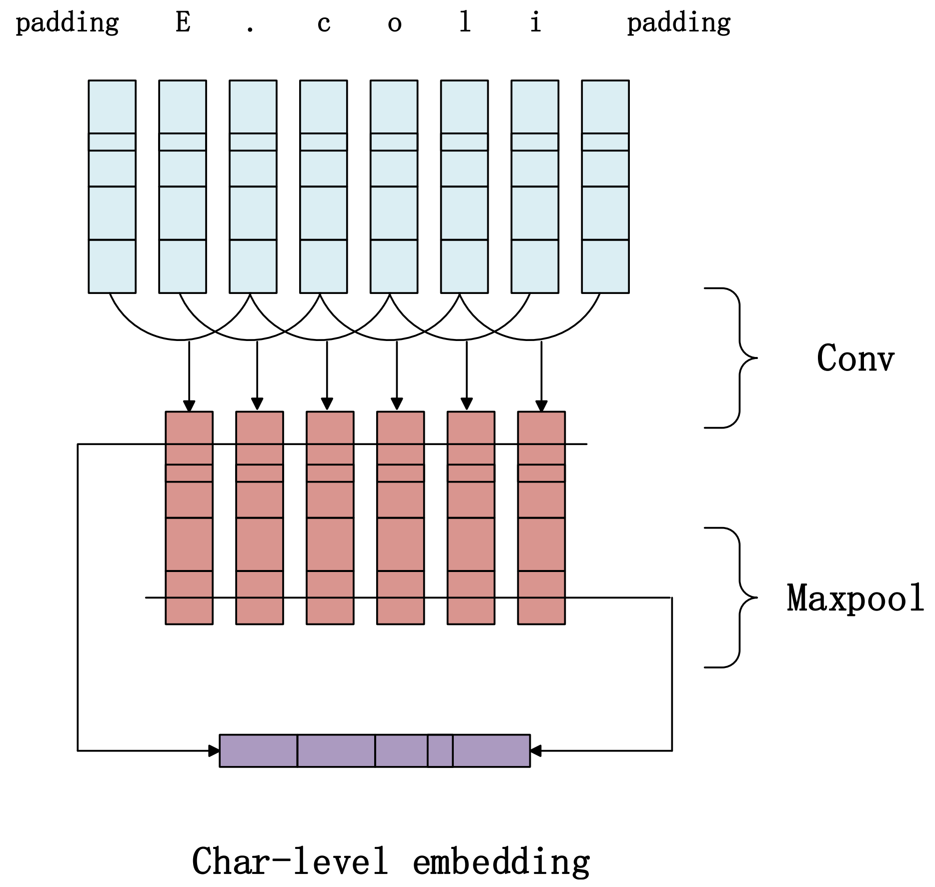

Fig. 2 The method to get char-level embedding in our paper. The characters in a word are transfer to vectors, then though a convolution layer and a max-pooling, finally the output are concatenated to represent the word

bidirectional maximum matching algorithm (BDMM) [32] to obtain dictionary features. UMLS [33] is a unified medical database, which contains volume of standardized names and abbreviations for diseases, proteins, genes and microorganisms. Hence we extract all the bacterial names from UMLS and integrate them into a bacterial dictionary. Table 1 gives an example of our preprocessing data.

Table 1 The example of the data format in our paper

sentence	pos	dict	tag
Actinobacillus	NNP	B-bacteria	B-bacteria
actinomycetemcomitans	NNS	I-bacteria	I-bacteria
,	,	O	O
Porphyromonas	NNP	B-bacteria	B-bacteria
gingivalis	NN	I-bacteria	I-bacteria
,	,	O	O
and	CC	O	O
Peptostreptococcus	NNP	B-bacteria	B-bacteria
micros	NNS	I-bacteria	I-bacteria

Encoding layer

The long short-term memory network is a [34] variant of recurrent neural network (RNN). It solves the problems of the gradient disappearance and the gradient explosion in the training process of RNN [35, 36]. In the practical application process, LSTM can handle the time series problem and the long-distance dependence problem well. It mainly consists of three gates: input gate, output gate and forget gate. The main formula is as follows:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{3}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * \tanh(\mathrm{W} \cdot [h_{t-1}, x_t] + b) \tag{6}$$

$$h_t = o_t * \tanh(C_t) \tag{7}$$

Where σ denote sigmoid function, x_t denote the input of LSTM, h_t denote the output of LSTM, W_f, W, W_o, W_i denote the weight matrix in the process of training , b_f, b_i, b_o, b is the bias vector.

For many sequence labeling tasks, we should consider the context information of the word at the same time, but a single LSTM structure can only obtain the historical information of the word. For this reason, Dyer [37] proposed a bidirectional long short-term memory (Bi-LSTM) network for acquiring the history information and future information of words. At first, given a sequence X= $[x_t]_0^n$, n represents the length of sequence, x_t is the input vector at time t, use a forward LSTM to obtain historical information $\overrightarrow{h_t}$ =LSTM ($\overrightarrow{h}_{t-1}$, x_t). Then a backward LSTM to obtain future information $\overleftarrow{h_t}= LSTM(\overleftarrow{h}_{t+1}, x_t)$. Finally, the outputs from both directions are concatenated to represent the word information $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$ learned at time t.

Decoding layer

For the task of sequence labeling, we should consider the dependency problem between words, because the neighboring words of the current word contribute to the labeling of the word, so we introduce the conditional random fields (CRF) [38] on the top of encoding layer. CRF has been proved to have a good effect on sequence labeling. Given the input of a sentence:

$$\mathrm{X} = (x_1, \ldots, x_i, \ldots, x_n) \tag{8}$$

Where x_i denote the vector representation of the output of encoding layer. We define P as the score matrix output by Bi- LSTM, the size of the matrix P is n × m, n represents the length of the sentence, m is the number of types of output tags and P_{ij} represents the probability of the j-th tag of the i-th word. The output of the definition sentence is:

$$\mathrm{y} \ \ = (y_1, \ldots, y_i, \ldots, y_n) \tag{9}$$

Where y_i represents the output prediction for each word. The score we define for the sentence is:

$$\mathrm{S(X, y)} = \sum_{i=0}^{n} T_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i} \tag{10}$$

Where T represents the tag transition matrix, for example, T_{ij} represents the transition probability from tag i to tag j. y_0 and y_{n+1} denote the start and end that we add to the matrix, so the size of T is m + 2. T is learned during training. Then, softmax function is used to normalize the output path y:

$$\mathrm{P(y|X)} = \frac{e^{s(X,y)}}{\sum_{\widetilde{y} \in Y} e^{S(X,\tilde{y})}} \tag{11}$$

Where Y is the set of all possible output sequences of sentence X, and we maximize log-probability of the correct output sequence during the training, which can represented as follows:

$$\log(\mathrm{P(y|X)}) = \mathrm{S(X, y)} - \log\left(\sum_{\tilde{y} \in Y} e^{S(X,\tilde{y})}\right) \tag{12}$$

In the decoding stage, we predict the best output path through maximizing the score function:

$$\mathrm{y}* = \underset{\tilde{y} \in Y}{argmax} S(X, \tilde{y}) \tag{13}$$

This process can be implemented by dynamic programming and inferred by Viterbi algorithm [39].

Dataset

In this paper, we utilize the dataset proposed by Wang [11] . They used "bacteria", "oral" and "human" as keywords to retrieve relevant abstracts from PubMed for nearly 10 years. At last they selected 1030 abstracts as train set and 314 abstracts as test set. The statistics about dataset are shown in Table 2. In order to evaluate the performance of the model, we divided it into training set, validation set and test set, in which 20% of the original training set was taken as validation set. We downloaded all abstracts related to "bacteria" from PubMed in the past decade and then trained word vectors along with the dataset.

Tagging scheme

In this experiment, our task is to give each word in the sentence a tag. As we investigated, a bacterial entity in a sentence may be composed of multiple words, so we need a set of identifiers to represent it. Currently, there are three main types of tagging scheme: IOB2, BIOE and BIOES. To compare the performance with other models, we use the IOB2 format as our tagging scheme. In the IOB2 tagging method, B-label represents the starting word of an entity, I-label represents the inside word of an entity, and O represents the word is not in entity.

Table 2 The statistics of the dataset in our experiment

Data set	abstract	sentence	token	The kind of entities	Entities	Entities token
Train set	1030	10094	252109	1767	7637	15272
Test set	314	3159	77638	770	2260	4611

Training and hyper-parameter settings

In this experiment, the following four parts constitute the input of our model: word embedding, character embedding, pos embedding, dict embedding. The word embedding is trained by word2vec with the dimension is 300, and the character embedding is trained by CNN. The initial input of the characters vector are 25-dimensional. The dimensions of the pos embedding and the dict embedding are 25, 5, respectively. The input embeddings all randomly initialized with uniform samples from $[\sqrt{-3/\ dim}, \sqrt{3/\ dim}]$ where dim is the dimension of embeddings [40]. The convolutional layers and fully connect layers were initialized with glorot uniform initialization [41], bias vectors are initialized with 0. Then the four embeddings are concatenated to input the model for parameter learning. During the training, we use the back propagation algorithm to update the parameters. Our optimization function is Adam [42] algorithms with a learning rate of 0.001 and a decay rate of 0.9.

We introduce dropout [43] and early stopping [44] technology to the model during the process of training. The purpose of the dropout technique is to prevent over-fitting of the model by randomly dropping some hidden nodes during the training process. We introduce dropout technology both before and after the decoding layer, which set dropout rate = 0.5. The principle of early stopping technology is to stop training when the result of the validation set is no longer improved within a tolerance range class, and record the parameters of model which has best result. It can prevent over-fitting of the model and select the best iteration number effectively. In this experiment, we set patience = 5. The detailed parameters are shown in Table 3.

Evaluation metrics

In order to evaluate the performance of the model proposed in this paper, we choose P (precision), R (recall) and F1 (F1-measure) as experiment metrics.

Table 3 The hyper-parameter in our experiment

Hyper-parameter	
word embedding	300
char embedding	25
pos embedding	25
dict embedding	5
filter size	3
filter deep	30
lstm hidden	100
Dropout	0.5

$$\mathrm{P} = \frac{TP}{TP + FP} \tag{14}$$

$$\mathrm{R} = \frac{TP}{TP + FN} \tag{15}$$

$$\mathrm{F1} = \frac{2 \times P \times R}{P + R} \tag{16}$$

Where TP is the number of entities correctly identified and FP is the number of non-entities identified as entities. F1-measure is the harmonic average of P and R.

Results and discussion

The experimental results are shown in Table 4. Model 1 and Model 2 were proposed by Wang [11, 12], and their models were based on traditional machine learning methods. Therefore, they manually extracted 43 groups of features, and then achieved good results on the dataset through feature combination and selection. Besides, the model based on Spark was greatly improved in speed. The model we proposed previously was based on neural network and did not need to extract features manually [22]. It was an end-to-end model and had enhanced the effect of the bacteria NER to some extent, but it did not make full use of the linguistic features and existing resources. In this paper, we consider the influence of domain features on the model. The experimental results show that the F1-measure of the model achieves 89.4% when adding the POS feature. With dictionary features and POS features are added, the model's F1-measure is up to 89.7%. From the above, we can include that these two features can effectively improve the effect of the model.

In order to evaluate the impact of word embedding on the model, we compare the performance of four pre-trained word embedding: glove [24], fastText [23], word2vec [25] and PubMed2vec [26] as well as random initialization in our model. Among them, glove and fastText are trained on Wikipedia which the dimension are 300, Pubmed2vec is 200 dimension which is trained on PubMed and PMC articles, and word2vec is based on the bacterial abstract training we downloaded from PubMed for 10 years. The experimental results are shown in Fig. 3. As can be seen from the figure, the use of the word embedding in the general domain has a

Table 4 The result of our model

Model	P	R	F
CRF and dictionary [11]	88.476%	81.149%	84.654%
spark [12]	89.443%	82.899%	86.047%
HDL_CRF [22]	90.009%	88.300%	89.146%
+pos	90.502%	88.344%	89.410%
+pos + dict	90.404%	89.007%	89.700%

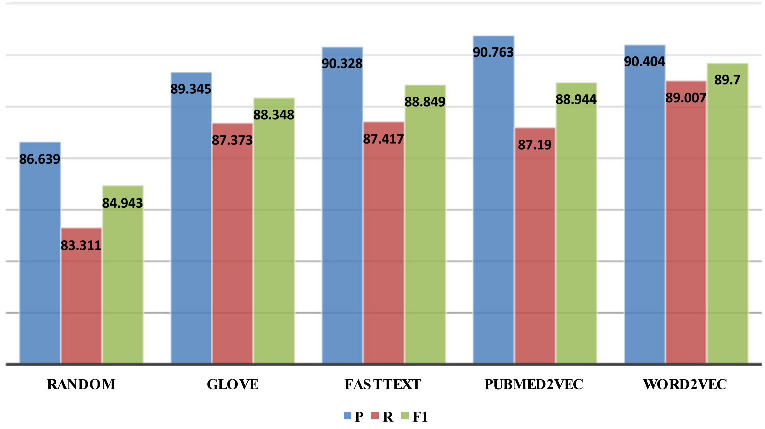

Fig. 3 The influence of different embedding in model

certain effect on the model compared with the random initialization and the performance is better than the model based on machine learning. Also, we can know that the result of using the medical field word vector is better than the general domain word vector, although it is not reach the highest. However, the F1-measure is the best when using the word vector of the bacterial field. As a result, the experiment proves that word vectors in different fields should be used for different professional problems, so that the model effect can be optimal and the error rate will be reduced.

To evaluate the practicability of our model, we utilize the model for named entity recognition on real data. We downloaded more than 400 thousand bacteria-related abstracts from PubMed for bacterial NER, and then compared the identified entity with the bacterial dictionary. UMLS [33] has collected nearly 4.5 million bacterial entities, which is relatively a large database of bacterial entities. Therefore, we extracted all bacterial entities from UMLS to construct a bacterial dictionary. Figure 4 is a comparison of experiments. Compared with 4.5 million bacterial entities in UMLS, more than 500 thousand bacterial entities are not in the dictionary when exact matching; however, when appending some rules, there still have more than 300 thousand entities not in the dictionary. Analyzing the entities predicted by our model shows that even though some predicted entities may be misidentified, our model can still largely predict mainly bacterial strains and bacteria in different ways of writing, and most of them are not updated or included in current dictionary.

Conclusion and outlook

This paper proposes a method for bacterial named entity recognition based on deep learning and domain features, integrating convolutional neural network, long short-term memory network, and conditional random fields. The experimental results demonstrate that the use of POS features and dictionary features can well promote the recognition of bacterial named entities. At the same time, we also compare the effects of different word embedding on the experimental results. The results

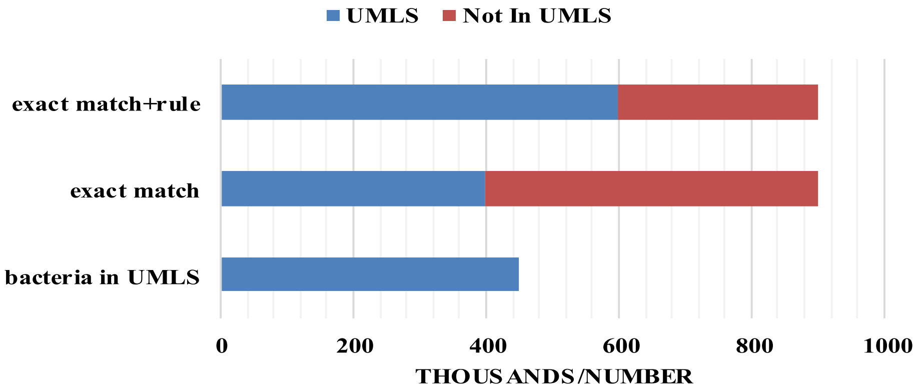

Fig. 4 The performance of our model in real dataset

illustrate that domain-specific embedding is more effective for bacterial named entity recognition.

Recently, language models have been widely used in the field of natural language, these models have achieved good results in many NLP tasks. In the future, we will combine the language model with bacterial named entity recognition, improve the effect of bacterial named entity recognition, and combine our task with interaction extraction.

About this supplement

This article has been published as part of *BMC Bioinformatics Volume 20 Supplement 16, 2019: Selected articles from the IEEE BIBM International Conference on Bioinformatics & Biomedicine (BIBM) 2018: bioinformatics and systems biology*. The full contents of the supplement are available online at https://bmcbioinformatics.biomedcentral.com/articles/supplements/volume-20-supplement-16.

Authors' contributions

XL and XJ proposed the method for bacterial named entity recognition, XL, CF, DZ and RZ write the paper and discussion. TH supervised and helped conceive the study. All of the authors have read and approved the final manuscript.

Funding

The research was supported by the National Key Research and Development Program of China (2017YFC0909502), the National Natural Science Foundation of China (61532008 and 61872157).
Specifically, the publication costs are funded by the National Key Research and Development Program of China (2017YFC0909502).

Availability of data and materials

The dataset and model can available at https://github.com/lixusheng1/bacterial_NER.git

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]School of Computer, Central China Normal University, Wuhan, Hubei, China. [2]Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning, Central China Normal University, Wuhan, Hubei, China. [3]Collaborative & Innovation Center, Central China Normal University, Wuhan, Hubei, China.

Published: 2 December 2019

References

1. Thomas S, Izard J, Walsh E, Batich K, Chongsathidkiet P, Clarke G, Sela DA, Muller AJ, Mullin JM, Albert K. The host microbiome regulates and maintains human health: a primer and perspective for non-microbiologists. Cancer Res. 2017;77(8):1783–812.
2. Dinan TG, Cryan JF. Melancholic microbes: a link between gut microbiota and depression? Neurogastroenterol Motil. 2013;25(9):713–9.
3. Larsen N, Vogensen FK, van den Berg FW, Nielsen DS, Andreasen AS, Pedersen BK, Al-Soud WA, Sørensen SJ, Hansen LH, Jakobsen M. Gut microbiota in human adults with type 2 diabetes differs from non-diabetic adults. PLoS One. 2010;5(2):e9085.
4. Bäckhed F, Ding H, Wang T, Hooper LV, Koh GY, Nagy A, Semenkovich CF, Gordon JI. The gut microbiota as an environmental factor that regulates fat storage. Proc Natl Acad Sci. 2004;101(44):15718–23.
5. Li C, Lim KMK, Chng KR, Nagarajan N. Predicting microbial interactions through computational approaches. Methods. 2016;102:12–9.
6. Freilich S, Kreimer A, Meilijson I, Gophna U, Sharan R, Ruppin E. The large-scale organization of the bacterial network of ecological co-occurrence interactions. Nucleic Acids Res. 2010;38(12):3857.
7. Lim KMK, Li C, Chng KR, Nagarajan N. @MInter: automated text-mining of microbial interactions. Bioinformatics. 2016;32(19):2981.
8. Proux D, Rechenmann F, Julliard L, Pillet V, Jacq B. Detecting gene symbols and names in biological texts. Genome Inform. 1998;9:72–80.
9. Settles B. Biomedical named entity recognition using conditional random fields and rich feature sets. In: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP): 2004. 107–10.
10. Habibi M, Weber L, Neves M, Wiegandt DL, Leser U. Deep learning with word embeddings improves biomedical named entity recognition. Bioinformatics. 2017;33(14):i37–48.
11. Wang X, Jiang X, Liu M, He T, Hu X. Bacterial named entity recognition based on dictionary and conditional random field. In: IEEE International Conference on Bioinformatics and Biomedicine; 2017. p. 439–44.
12. Wang X, Li Y, He T, Jiang X, Hu X. Recognition of bacteria named entity using conditional random fields in spark. BMC Syst Biol. 2018;12(6):106.
13. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, Van Der Laak JA, Van Ginneken B, Sánchez CI. A survey on deep learning in medical image analysis. Med Image Anal. 2017;42:60–88.
14. Deng L, Li J, Huang J-T, Yao K, Yu D, Seide F, Seltzer ML, Zweig G, He X, Williams JD. Recent advances in deep learning for speech research at Microsoft. In: ICASSP, vol. 64; 2013.
15. Chaudhary JR, Patel AC. Machine translation using deep learning: a survey; 2018.
16. Wang Z, Mi H, Hamza W, Florian R: Multi-perspective context matching for machine comprehension. arXiv preprint arXiv:161204211 2016.
17. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C: Neural architectures for named entity recognition. arXiv preprint arXiv:160301360 2016.
18. Ma X, Hovy E: End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:160301354 2016.
19. Nédellec C, Bossy R, Kim J-D, Kim J-J, Ohta T, Pyysalo S, Zweigenbaum P. Overview of BioNLP shared task 2013. In: Proceedings of the BioNLP shared task 2013 workshop; 2013. p. 1–7.
20. Wei C-H, Peng Y, Leaman R, Davis AP, Mattingly CJ, Li J, Wiegers TC, Lu Z. Overview of the BioCreative V chemical disease relation (CDR) task. In: Proceedings of the fifth BioCreative challenge evaluation workshop; 2015. p. 154–66.
21. Crichton G, Pyysalo S, Chiu B, Korhonen A. A neural network multi-task learning approach to biomedical named entity recognition. BMC Bioinformatics. 2017;18(1):368.
22. Li X, Wang X, Zhong R, Zhong D, Jiang X, He T, Hu X: A hybrid deep learning framework for bacterial named entity recognition. IEEE International Conference on Bioinformatics and Biomedicine: 2018.
23. Joulin A, Grave E, Bojanowski P, Douze M, Jégou H, Mikolov T: Fasttext. Zip: compressing text classification models. arXiv preprint arXiv:161203651 2016.
24. Pennington J, Socher R, Manning C. Glove: global vectors for word representation. In: Conference on Empirical Methods in Natural Language Processing; 2014. p. 1532–43.
25. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. Adv Neural Inf Proces Syst. 2013;26:3111–9.
26. Moen S, Ananiadou TSS. Distributional semantics resources for biomedical text processing. Proceedings of LBM 2013:39–44.
27. Řehůřek R, Sojka P. Software framework for topic modelling with large corpora. In: Lrec 2010 workshop on new challenges for Nlp frameworks; 2010. p. 45–50.
28. Kim Y, Jernite Y, Sontag D, Rush AM. Character-aware neural language models. In: AAAI; 2016. 27412749.
29. Chiu JP, Nichols E: Named entity recognition with bidirectional LSTM-CNNs. arXiv preprint arXiv:151108308 2015.
30. Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:150801991 2015.
31. Loper E, Bird S. NLTK: the natural language toolkit. arXiv preprint cs/0205028 2002.
32. Gai RL, Gao F, Duan LM, Sun XH, Li HZ. Bidirectional maximal matching word segmentation algorithm with rules. In: Advanced Materials Research; 2014. p. 3368–72. Trans Tech Publ.

33. Bodenreider O. The unified medical language system (UMLS): integrating biomedical terminology. Nucleic Acids Res. 2004;32(suppl_1):D267–70.
34. Sepp Hochreiter, Jürgen Schmidhuber, (1997) Long Short-Term Memory. Neural Computation 9 (8):1735–80.
35. Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. In: International conference on international conference on machine learning; 2013. III-1310.
36. Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw. 1994;5(2):157–66.
37. Dyer C, Ballesteros M, Ling W, Matthews A, Smith NA: Transition-based dependency parsing with stack long short-term memory. arXiv preprint arXiv:150508075 2015.
38. Lafferty JD, Mccallum A, Pereira FCN. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Eighteenth international conference on machine learning; 2001. p. 282–9.
39. Forney GD Jr. The viterbi algorithm. Proc IEEE. 1993;61(5):268–78.
40. He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 1026–34.
41. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics; 2010. p. 249–56.
42. Kingma DP, Ba J: Adam: a method for stochastic optimization. arXiv preprint arXiv:14126980 2014.
43. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014;15(1):1929–58.
44. Prechelt L. Automatic early stopping using cross validation: quantifying the criteria. Neural Netw. 1998;11(4):761–7.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.