

RESEARCH ARTICLE

Open Access



Improving biomedical named entity recognition with syntactic information

Yuanhe Tian^{1†}, Wang Shen^{2†}, Yan Song^{3,4*} , Fei Xia¹, Min He² and Kenli Li²

*Correspondence:

songyan@cuhk.edu.cn

[†]Yuanhe Tian and Wang Shen contributed equally to this work

³The Chinese University of Hong Kong, Shenzhen, China

Full list of author information is available at the end of the article

Abstract

Background: Biomedical named entity recognition (BioNER) is an important task for understanding biomedical texts, which can be challenging due to the lack of large-scale labeled training data and domain knowledge. To address the challenge, in addition to using powerful encoders (e.g., biLSTM and BioBERT), one possible method is to leverage extra knowledge that is easy to obtain. Previous studies have shown that auto-processed syntactic information can be a useful resource to improve model performance, but their approaches are limited to directly concatenating the embeddings of syntactic information to the input word embeddings. Therefore, such syntactic information is leveraged in an inflexible way, where inaccurate one may hurt model performance.

Results: In this paper, we propose BioKMNER, a BioNER model for biomedical texts with key-value memory networks (KVMN) to incorporate auto-processed syntactic information. We evaluate BioKMNER on six English biomedical datasets, where our method with KVMN outperforms the strong baseline method, namely, BioBERT, from the previous study on all datasets. Specifically, the F1 scores of our best performing model are 85.29% on BC2GM, 77.83% on JNLPBA, 94.22% on BC5CDR-chemical, 90.08% on NCBI-disease, 89.24% on LINNAEUS, and 76.33% on Species-800, where state-of-the-art performance is obtained on four of them (i.e., BC2GM, BC5CDR-chemical, NCBI-disease, and Species-800).

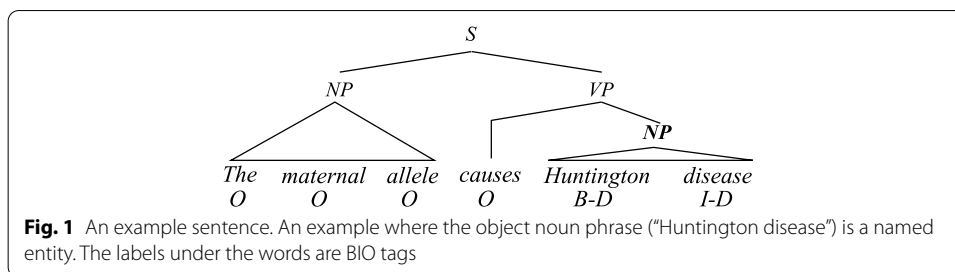
Conclusion: The experimental results on six English benchmark datasets demonstrate that auto-processed syntactic information can be a useful resource for BioNER and our method with KVMN can appropriately leverage such information to improve model performance.

Keywords: Named entity recognition, Text mining, Key-value memory networks, Syntactic information, Neural networks

Background

Biomedical named entity recognition (BioNER) is an important and challenging task for understanding biomedical texts. It aims to recognize named entities (NEs), such as diseases, gene, species, etc., in biomedical texts and plays an important role in many downstream natural language processing (NLP) tasks, such as drug-drug interaction task [21, 34] and knowledge base completion [38, 47]. Compared to named entity recognition in





the general domain, BioNER is considered to be more difficult due to the lack of large-scale labeled training data and domain knowledge.

In the past decades, there have been many studies on BioNER, ranging from traditional feature based methods [4, 15–18, 20, 37] to recent deep learning based neural methods [5, 12, 19, 23, 32, 45]. Among the neural methods, the ones leveraging powerful encoders (e.g., biLSTM) achieve better results comparing with feature based methods because such encoders are good at modeling contextual information. More recently, pre-trained models such as ELMo [30] and BERT [6] achieved state-of-the-art performance on many NLP tasks in the general domain. Therefore, some studies [13, 19] applied them to BioNER yet found that these models cannot perform as well as in the general domain when there is no domain-specific information integrated. Therefore, Lee et al. [19] proposed a variant of BERT, namely, BioBERT, for the biomedical domain, which is pre-trained on large raw biomedical corpora and achieves state-of-the-art performance in BioNER.

In addition to the powerful encoders, syntactic information has also been playing an important role in many previous studies to help recognize biomedical named entities [4, 5, 20, 23, 37]. Intuitively, biomedical text often includes formal, well-structured, and long sentences, where syntactic information could be helpful because it can provide useful cues for recognizing NEs and thus help with the text understanding of NLP systems [36]. For example, Fig. 1 shows the parse tree of a sentence where the disease entity “Huntington disease” forms the object; thus, the boundary of a noun phrase can be a good cue for NER. Moreover, comparing with other types of extra resources, e.g., knowledge base [1, 24, 49], which are generally not publicly available or require human annotations, the syntactic information is easier to obtain through off-the-shelf NLP toolkits. Therefore, considering that the state-of-the-art BioBERT [19] does not leverage any syntactic information, we propose to improve BioBERT by incorporating the syntactic information of the input text, which is obtained from the parsing results of off-the-shelf toolkits.

To incorporate syntactic information into BioNER methods, previous studies have tried several ways. In the feature engineering methods, researchers use syntactic information to generate handcrafted features to help BioNER. For example, Song et al. [37] used part-of-speech (POS) and noun phrase tag features in a CRF-based BioNER system. In recent deep learning based methods, syntactic information is firstly represented by vectorized embeddings and then combined with word embedding through vector concatenation or element-wise summation, after which the resulting vector is fed into neural models to improve bioNER. For example, Luo et al. [23] used embedding vectors to represent syntactic information including POS and constituent labels, and concatenated

those vectors with word embeddings. The combined embeddings were then sent into a biLSTM-CRF model with an attention mechanism to detect chemical NE. Dang et al. [5] proposed a model named D₃NER, where the embeddings of various informative syntactic information are used to improve the results. Overall, previous approaches to leverage auto-processed syntactic information were limited to directly concatenating the embeddings of the syntactic information instances and the input words, without weighing the syntactic information instances in a specific context, where noisy information may hurt model performance. Therefore, we need to find a better method to leverage auto-processed syntactic information.

To weigh the syntactic information instances and leverage the important ones to improve BioNER methods, key-value memory networks (KVMN) [26] could be potentially useful, because it is demonstrated to be useful in leveraging extra information, e.g. knowledge base entities, to improve question answering tasks. In KVMN, the information is represented by key-value memory slots, where the keys are used to compute the weights for values by comparing these keys with the input, and the values are weighted summed according to the resulting weights and then used to make predictions. In addition, although the KVMN is originally proposed for question answering tasks, its variants also demonstrate impressive performance in many NLP tasks, such as Chinese word segmentation [40], semantic role labeling [11], and machine translation [27]. This motivates us to explore the possibility of using KVMN to leverage the syntactic information to improve BioNER.

Therefore, in this paper, we propose BioKMNER (KM stands for Key-value Memory network), which uses KVMN to incorporate syntactic information into the backbone sequence labeling tagger to improve BioNER. Specifically, we firstly use off-the-shelf toolkits to parse biomedical text sentences and extract three types of syntactic information: namely, POS labels, syntactic constituents, and dependency relations. Then, for each word in the input sentence, in the KVMN, we use the keys to represent the context features associated with the word and the values to denote the corresponding syntactic information instances. Therefore, context features (keys) are used to compute the weights by comparing them with the input word, and syntactic information instances (values) are weighed accordingly. Finally, the weighted summed values are concatenated with the output of the encoder, where the resulting vector is fed into the decoder for prediction. In this way, the method can incorporate the pair-wisely organized context features and syntactic information instances obtained from the toolkits simultaneously. Different from previous studies that directly use noisy syntactic information instances by embedding concatenation, our BioKMNER weighs them in KVMN and thus reduces the effect of error propagation caused by the noisy parsing results. We experiment BioKMNER on six English benchmark BioNER datasets covering four different entity types (i.e., chemical, disease, gene/protein, and species). The results demonstrate the effectiveness of our method for BioNER, where BioKMNER outperforms the BioBERT results reported by Lee et al. [19] on all datasets and achieves state-of-the-art results on four of them.

Results

Datasets

We evaluate our methods on six English benchmark datasets that are widely used in previous studies [10, 12, 16, 19, 48]. These datasets focus on four different biomedical entity types: BC2GM dataset [35] and JNLPBA dataset [14] for gene/protein NER, BC5CDR-chemical dataset [44] for chemical NER, NCBI-disease dataset [8] for disease NER, and LINNAEUS dataset [9] and Species-800 dataset [29] for species NER.

BC2GM BC2GM is a dataset used for the BioCreative II gene mention tagging task.¹ It contains 20,000 sentences from abstracts of biomedical publications and is annotated with 24,583 mentions of the names of genes, proteins and related entities. It has become the major benchmark for the NER of gene/proteins entity type [10, 12, 19, 31, 43, 48].

JNLPBA JNLPBA is the dataset for the Joint Workshop on NLP in Biomedicine and its Application Shared task.² It was organized by the GENIA Project based on the annotations of the GENIA Term corpus and consists of 2404 publication abstracts. It is widely used for evaluating multi-class biomedical entity taggers.

BC5CDR-chemical BC5CDR is a dataset used for the BioCreative V Chemical Disease Relation (CDR) Task.³ It contains 1500 titles and abstracts from PubMed,⁴ where chemical and disease mentions are annotated by human annotators. Following previous studies [23], we only use the subset with chemical entities and denote it as BC5CDR-chemical.

NCBI-disease NCBI-disease contains 793 PubMed abstracts that are annotated with disease mentions and their corresponding concepts. There are 6,892 disease mentions from 790 unique disease concepts in this dataset and 91% of the mentions are mapped to a single disease concept. It has been a widely used research resource for the disease NER.

LINNAEUS The LINNAEUS dataset was created specifically for species named entity recognition and consists of 100 full-text documents. In the LINNAEUS dataset, all mentions of species terms were manually annotated and normalized to the NCBI taxonomy IDs of the intended species.

Species-800 Species-800 is a novel benchmark corpus for species entities, which is based on manually annotated abstracts. It comprises 800 PubMed abstracts that contain identified organism mentions. Because the abstracts are select from journals on 8 different categories, the diversity of Species-800 is high and thus it is more challenging for NER systems.

We follow the study of Lee et al. [19] to pre-process all datasets. In details, BC2GM, BC5CDR-chemical, LINNAEUS, and NCBI-disease datasets are pre-processed based on the schema of Wang et al. [43]; JNLPBA is pre-processed by CoNLL format;⁵ and Species-800 is pre-processed by Pyysalo.⁶ The statistics of all datasets in terms of the number of tokens, sentences, and entities are reported in Table 1.

¹ <https://biocreative.bioinformatics.udel.edu/tasks/biocreative-ii/task-1a-gene-mention-tagging/>.

² <http://www.geniaproject.org/shared-tasks/bionlp-jnlpba-shared-task-2004>.

³ <https://biocreative.bioinformatics.udel.edu/tasks/biocreative-v/track-3-cdr/>.

⁴ <https://pubmed.ncbi.nlm.nih.gov/>.

⁵ <https://github.com/spyysalo/standoff2conll>.

⁶ <https://github.com/spyysalo/s800>.

Table 1 The statistics of the four benchmark datasets

Datasets	Entity type		Token #	Sent. #	Entity #
BC2GM	Gene/protein	Train	355.4k	12.5k	15.1k
		Dev	71.0k	2.5k	3.0k
		Test	143.4k	5.0k	6.3k
JNLPBA		Train	443.6k	14.6k	32.1k
		Dev	117.2k	3.8k	8.5k
		Test	114.7k	3.8k	6.2k
BC5CDR-chemical	Chemical	Train	118.1K	4.5K	5.2K
		Dev	117.4K	4.5K	5.3K
		Test	124.7K	4.7K	5.3K
NCBI-disease	Disease	Train	135.7K	5.4K	5.1K
		Dev	23.9K	923	787
		Test	24.4K	940	960
LINNAEUS	Species	Train	281.2k	11.9k	2.1k
		Dev	93.8k	4.0k	711
		Test	165k	7.1k	1.4k
Species-800		Train	147.2K	5.7K	2.5K
		Dev	22.2K	830	384
		Test	42.2K	1.6K	767

"Token #", "Sent. #" and "Entity #" represent the number of tokens, sentences, and entities

Implementation

In the experiments, we use off-the-shelf NLP toolkits to generate syntactic information, following the common practice in previous studies such as Mohit and Hwa [28], Tkachenko and Simanovsky [42], and Luo et al. [23]. In our study, we focus on three types of syntactic information: POS labels, syntactic constituents, and dependency relations. We use Stanford CoreNLP Toolkits (SCT)⁷ [25], which is a well-known NLP toolkit used in many studies [33, 39], to obtain the POS tagging, constituency, and dependency parsing results of a given input sentence.

For the encoder, considering that BERT [6] and its variants [2, 3, 7, 19] achieve state-of-the-art performance on many NLP tasks, we use the variant for the medical domain, i.e., BioBERT [19], in our method. Specifically, we use both the base and large version of BioBERT⁸ and follow the hyper-parameters used by Lee et al. [19] (i.e., for BioBERT-Base, there are 12 self-attention heads with 768-dimensional hidden vectors; for BioBERT-Large, the number of heads is 24 with 1024-dimensional hidden vectors). All parameters in the encoder are fine-tuned in training. For the KVMN module, the embeddings of all keys and values are randomly initialized, with their dimension matching the dimension of hidden vectors in the BioBERT encoder. Besides, we follow the setting of Lee et al. [19] to run the training process, where we do not use the "alternate" annotations for the BC2GM dataset. Moreover, for each method, we train five models with different random seeds to initialize the model parameters and use the average of their micro F1 scores for evaluation.⁹ In the experiments, we train each model for 150

⁷ We use v3.9.2, downloaded from <https://stanfordnlp.github.io/CoreNLP/>.

⁸ We obtain the pre-trained models v1.1 from <https://github.com/naver/biobert-pretrained>

⁹ We evaluate all models by the widely used *seqeval* framework at <https://github.com/chakki-works/seqeval>.

Table 2 Experimental results of models on six benchmark datasets

Methods	BC2GM		JNLPBA		BC5CDR-chemical		NCBI-disease		LINNAEUS		Species-800	
	F1	σ	F1	σ	F1	σ	F1	σ	F1	σ	F1	σ
Base	84.61	0.21	76.85	0.31	93.50	0.10	88.63	0.71	88.27	0.32	74.97	0.46
+ PL (DC)	84.47	0.15	77.17	0.45	93.66	0.15	89.09	0.55	88.36	0.16	75.04	0.46
+ PL (\mathcal{M})	84.74	0.10	77.06	0.05	93.73	0.19	89.47	0.56	88.44	0.30	75.45	0.41
+ SC (DC)	84.45	0.19	76.80	0.45	93.68	0.13	89.18	0.26	88.23	0.33	75.37	0.51
+ SC (\mathcal{M})	84.76	0.21	77.17	0.16	93.74	0.11	89.27	0.52	88.68	0.30	75.65	0.50
+ DR (DC)	84.33	0.30	77.01	0.28	93.66	0.15	89.05	0.23	88.43	0.19	75.12	0.52
+ DR (\mathcal{M})	84.65	0.27	77.32	0.35	93.78	0.18	89.24	0.60	88.57	0.15	75.81	0.71
Large	84.89	0.17	77.29	0.19	93.90	0.31	88.65	0.59	88.87	0.65	74.98	0.59
+ PL (DC)	85.06	0.08	77.56	0.18	93.90	0.16	88.74	0.26	88.65	0.39	74.92	0.86
+ PL (\mathcal{M})	85.07	0.12	77.50	0.19	94.05	0.23	88.86	0.29	89.01	0.31	75.34	0.95
+ SC (DC)	85.12	0.13	77.56	0.12	93.95	0.09	88.78	0.54	89.01	0.28	75.38	0.29
+ SC (\mathcal{M})	85.43	0.15	77.83	0.19	93.99	0.13	88.87	0.37	88.92	0.35	75.08	0.68
+ DR (DC)	85.01	0.12	77.58	0.10	93.97	0.17	89.37	0.30	88.99	0.22	75.01	0.83
+ DR (\mathcal{M})	85.17	0.10	77.73	0.11	94.05	0.10	88.81	0.51	89.04	0.27	75.17	0.91

The experimental results are reported in terms of average F1 scores (F1) and the standard deviation σ . The methods in the group "Base" and "Large" refer to baselines with BioBERT-Base and BioBERT-Large encoder and our methods with KVMN (\mathcal{M}). "DC" refers to the baseline method using direct concatenation to incorporate syntactic information. "PL", "SC", and "DR" stand for POS labels, syntactic constituents, and dependency relations, respectively

epochs for the BC2GM dataset and for 60 epochs for all other datasets.¹⁰ In each run, we evaluate the model on the development set after each epoch to find its best performing result.

Overall results

We run the baseline methods without using syntactic information and our methods with KVMN (\mathcal{M}) to incorporate three types of syntactic information obtained from SCT on six aforementioned datasets, where two different encoders, i.e., BioBERT-Base and BioBERT-Large, are used. For reference, we also run baseline methods that use direct concatenation (DC) to incorporate such syntactic information, where the embeddings of context features and syntactic information are directly concatenated with the output of the BioBERT encoder. We report the experimental results (the average F1 scores of the five runs for each method as well as the standard deviations σ) in Table 2. There are some observations.

First, comparing with the baseline methods without using any syntactic information, our method with KVMN can work well with both BioBERT-Base and BioBERT-Large encoder, where decent improvements over the baseline methods are observed among all datasets.

Second, compared with DC, our methods with KVMN to incorporate syntactic information achieve better results in most cases. For example, on the Species-800 dataset, our method (Base + DR (\mathcal{M})) obtains an average F1 score of 75.81%, while its corresponding DC-based method (Base + DR (DC)) obtains a lower average F1 score of 75.12%.

¹⁰ All experiments are run on a single Nvidia Tesla V100 GPU with 16G RAM.

Table 3 Comparison with previous deep learning based methods

Methods	BC2GM	JNLPBA	BC5CDR-chemical	NCBI-disease	LINNAEUS	Species-800
biLSTM + pre-trained embeddings [12]	78.57	77.25	91.05	84.64	94.13	73.11
biLSTM + attentions [23]	–	–	92.57	–	–	–
biLSTM + multi-task learning [43]	80.74	73.52	–	86.14	–	–
biLSTM + pre-training [31]	81.69	75.03	–	87.34	–	–
biLSTM + transfer learning [10]	78.66	–	91.64	84.72	93.54	74.98
biLSTM + model ensemble [48]	79.73	78.58	93.31	86.36	–	–
SciBERT [3]	–	77.28	–	88.57	–	–
BERT [19]	81.79	74.94	91.16	85.63	87.60	71.63
BioBERT (Base) [19]	84.72	77.49	93.47	89.71	88.24	75.31
BioBERT (Large) [19]	85.01	–	–	88.79	–	–
BioBERT (Base) + DR (\mathcal{M})	84.92	77.72	94.00	90.08	88.79	76.21
BioBERT (Large) + DR (\mathcal{M})	85.29	77.83	94.22	89.63	89.24	76.33

The result (F1 scores) of our method on each dataset comes from the best performing model. The results for the base and large version of BioBERT [19] are from their paper and GitHub repository

We report the results of their version 1.1, which is identical to the BioBERT version used in our experiments

Besides, in some cases where DC is applied, the syntactic information causes inferior results than baselines. For example, on the LINNAEUS dataset, the average F1 score of the DC-based method with the POS labels (Large + PL (DC)) is lower than the baseline (Large) results. One possible explanation could be: there are some noisy syntactic results extracted by off-the-shelf toolkits, which may influence the performance of the model and lead to worse results compared to the baselines only using BioBERT. Under this condition, methods with DC fails to distinguish the salient syntactic information that contributes more to the bioNER task in a specific context. On the contrary, KVMN can weigh such syntactic information according to the importance of the context features and thus, to some extent, avoid the errors caused by incorporating auto-processed syntactic information.

Third, in many cases, in methods with KVMN, the information of syntactic constituents (SC) and dependency relations (DR) offers higher improvement than POS labels (PL). For example, on the BC2GM dataset, our method with the BioBERT-Large encoder obtains the average F1 scores of 85.43% and 85.17% when it is enhanced by SC and DR, respectively, while its average F1 score is 85.07% when PL is incorporated. One possible reason to explain the phenomenon could be: (1) the syntactic constituents can provide a cue of phrase functions and their boundaries (e.g., an NP treelet is not only a signal that can suggest there might be an NE inside, but also can give information about the possible starting and ending positions for that potential NE.); (2) the dependency relations link words in long-distance with their dependency relationships, which could be especially useful for biomedical texts that generally include long sentences and entities.

Discussion

Comparison with previous studies

We compare the results of our best performing model with previous studies on all aforementioned datasets. The results (F1 scores) are summarized in Table 3, where our

Table 4 Results of the syntactic information ensemble on BC5CDR-chemical dataset

Ensemble strategies	Syntactic info.			BioBERT-Base		BioBERT-Large	
	PL	SC	DR	F1	σ	F1	σ
Baseline	×	×	×	93.50	0.10	93.90	0.31
Sum	✓	✓	×	93.66	0.17	94.20	0.15
	✓	×	✓	93.76	0.16	94.10	0.15
	×	✓	✓	93.81	0.15	94.12	0.14
	✓	✓	✓	93.78	0.25	94.26	0.16
Concatenation	✓	✓	×	93.75	0.23	94.25	0.12
	✓	×	✓	93.80	0.26	94.22	0.16
	×	✓	✓	93.83	0.20	94.31	0.08
	✓	✓	✓	93.88	0.26	94.36	0.25

The three types of syntactic information used for the ensemble are POS labels (PL), syntactic constituents (SC), and dependency relations (DR). The results are reported in terms of the average F1 scores and the standard deviation (σ). Sum and concatenation are two ensemble strategies applied to our method

method outperforms the previous study (i.e., Lee et al. [19]) using the base and large version of BioBERT on all datasets. Specifically, for the BioBERT-Base, the improvement of F1 scores on BC2GM, JNLPBA, BC5CDR-chemical, NCBI-disease, LINNAEUS, and Species-800 are 0.20%, 0.23%, 0.53%, 0.37%, 0.55%, and 0.90% respectively; for the BioBERT-Large, the improvement on BC2GM and NCBI-disease are 0.28% and 0.84%, respectively. These results demonstrate the effectiveness of our method to leverage auto-processed syntactic information in recognizing different types of named entities in the biomedical domain. In addition, our method achieves state-of-the-art performance on four datasets, i.e., BC2GM, BC5CDR-chemical, NCBI-disease, and Species-800. Compared with [48] and [12], we do not outperform their results on JNLPBA and LINNAEUS, because the gaps between their results and our baseline method, i.e., BioBERT from Lee et al. [19], are big on these datasets, which could be hard to compensate for by using syntactic information. Except for the two datasets, our method outperforms their methods on all other datasets.

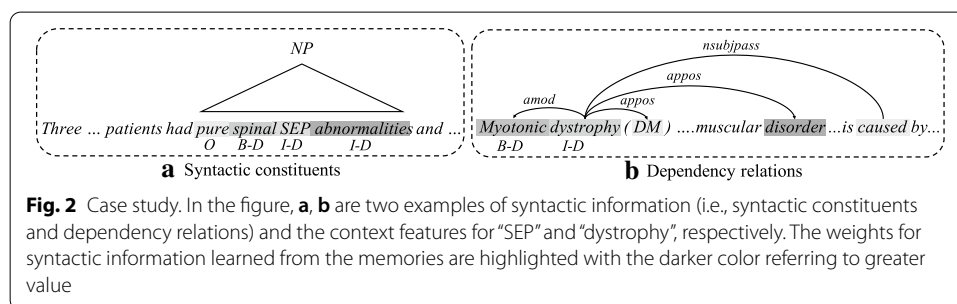
Effect of syntactic information ensemble

To explore the effect of using different types of syntactic information together, we conduct syntactic information ensemble experiments on the BC5CDR-chemical dataset. In the experiments, we test different combinations of different types of syntactic information, where two ensemble strategies are used. The first sums the weighted value embeddings of each type of syntactic information; and the second uses concatenation. The results of the average F1 scores of different settings are reported in Table 4, where the results from the baseline methods without using any syntactic information are also included for reference. We have several observations from it. First, overall, compared with the baseline methods, our methods achieve better results with both the base and large versions of the BioBERT encoder. This indicates that the combination of different types of syntactic information can help with the performance of the baseline method for BioNER. Second, the concatenation strategy performs better than the summing strategy in syntactic information fusion. One possible explanation could be: summing the

Table 5 Results of using different NLP toolkits on the BC5CDR-chemical dataset

	BioBERT-base		BioBERT-large	
	F1	σ	F1	σ
Baseline	93.50	0.10	93.90	0.31
Stanford CoreNLP Toolkits				
PL (\mathcal{M})	93.73	0.19	94.05	0.23
DR (\mathcal{M})	93.78	0.18	94.05	0.10
spaCy				
PL (\mathcal{M})	93.69	0.12	94.06	0.10
DR (\mathcal{M})	93.71	0.12	93.97	0.13

The experimental results [the average F1 scores and the standard deviation (σ)] of our method with KVMN (\mathcal{M}) using different NLP toolkits (i.e., Stanford CoreNLP Toolkits and spaCy) to obtain POS labels (PL) and dependency relations (DR). The results of baseline methods without using any syntactic information are also reported for reference



embeddings of different types of syntactic information may lose some information while concatenating them can keep all information on all types of syntactic embedding.

Effect of different toolkits

To explore the effect of using syntactic information from different NLP toolkits, in addition to SCT, we try another toolkit, i.e., spaCy,¹¹ to obtain the auto-processed syntactic information. In the experiments, we try two types of syntactic information, i.e., POS labels (PL) and dependency relations (DR), from the POS tagger and dependency parser of spaCy. We report the results (the average F1 scores and the standard deviation σ) of our methods with KVMN on the BC5CDR-chemical dataset in Table 5. For reference, the results of our method using SCT as well as the baseline results are also reported. From the results, we can find that, for both base and large BioBERT encoders, our method can leverage the syntactic information from different NLP toolkits and thus achieves better performance comparing with the baseline methods.

Case study

To better understand how our method improves BioNER, we conduct a case study where two example sentences are used. In Fig. 2a, b, we show two sentences and illustrate the way of syntactic constituents and dependency relations to improve

¹¹ <https://spacy.io/>.

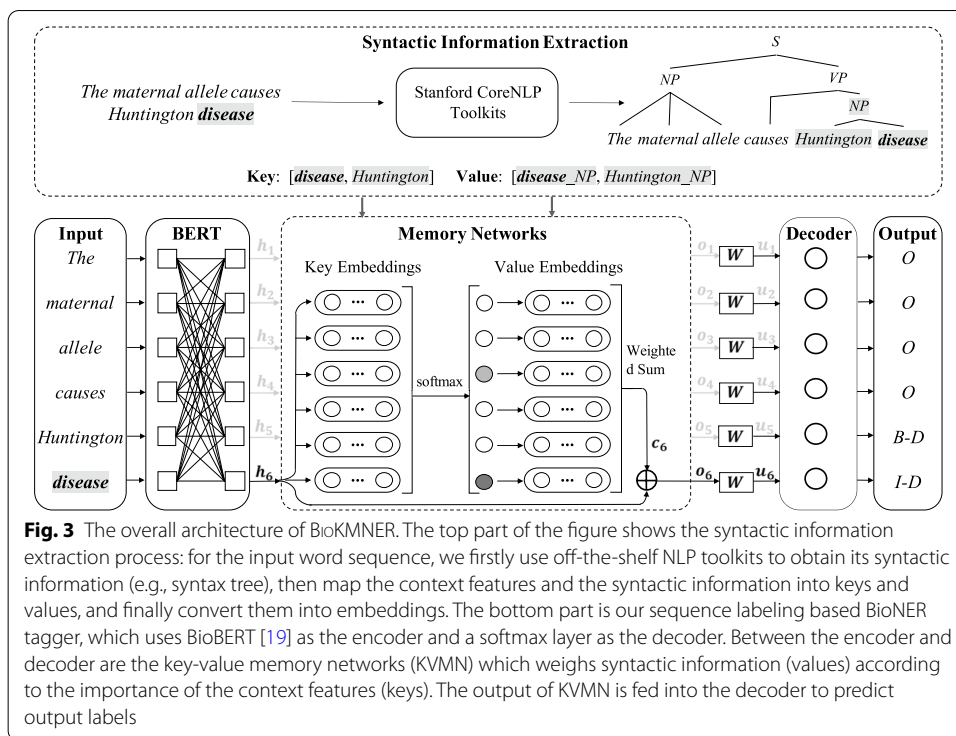
bioNER, respectively. In both cases, for a specific word, we visualize the weights assigned to the corresponding syntactic information instances (values) on its associated contextual features (keys), where the deeper color refers to the higher weight.

Syntactic constituents In the example sentence shown in Fig. 2a, the word we focus on is “SEP”. In this case, the constituent information firstly narrows the context features of “SEP” down to the words within the noun phrase “pure spinal SEP abnormalities”. Then, the KVMN module assigns the highest weight to “abnormalities” and its carrying syntactic information of “NP” among all other syntactic instances since they could be strong signals for disease names. Therefore, our method could assign the correct NE label to “SEP”. Likewise, the situation for “pure” is on the opposite and thus it receives the lowest weight among other words.

Dependency relations In addition, in Fig. 2b, we visualize the weights assigned to dependency relations for the word “dystrophy” in another example sentence. In this case, dependency information successfully finds the dependents, i.e., “Myotonic”, “DM”, and “disorder”, of “dystrophy”, which could suggest useful cues to predict the NE labels. Among those dependents, KVMN distinguishes that the dependent “disorder” with an “appos” dependency relation (appositional modifier) strongly suggests “dystrophy” is a disease entity. Therefore, KVMN assigns the highest weight to the dependency relation offered by “disorder”. Similarly, another modifier (i.e., “Myotonic”) of “dystrophy” is also distinguished and weighed by the KVMN, and the second-highest weight is assigned to it accordingly. It is worth noting that the dependency information that contributes most to recognizing “dystrophy” as a part of an NE comes from a word (“disorder”) in the long-distance; dependency information is able to capture that information and helps our method predict the NE tag for the word “dystrophy”.

Conclusion

In this paper, we propose a method named BioKMNER with KVMN to enhance BioNER with auto-processed syntactic information (i.e., POS labels, syntactic constituents, and dependency relations) from off-the-shelf toolkits. In KVMN, context features and their corresponding syntactic information instances are mapped into keys and values, respectively. The values are weighed according to the comparison between the keys and the input words. Then the values are weighed summed and the resulting vector is fed back to the backbone tagging process to make predictions. In doing so, compared with previous studies that treat different syntactic information equally and leverage them by embedding concatenation, our method can discriminatively leverage the auto-processed syntactic information and avoid the error spread caused by the direct use of auto-processed syntactic results. The experimental results on six English benchmark datasets demonstrate that syntactic information can be a good resource to improve bioNER and our method with KVMN can appropriately leverage such information. In addition, our method outperforms the strong baseline method from the previous study using BioBERT [19] on all datasets and achieves state-of-the-art results on BC2GM, BC5CDR-chemical, NCBI-disease, and Species-800 datasets.



Methods

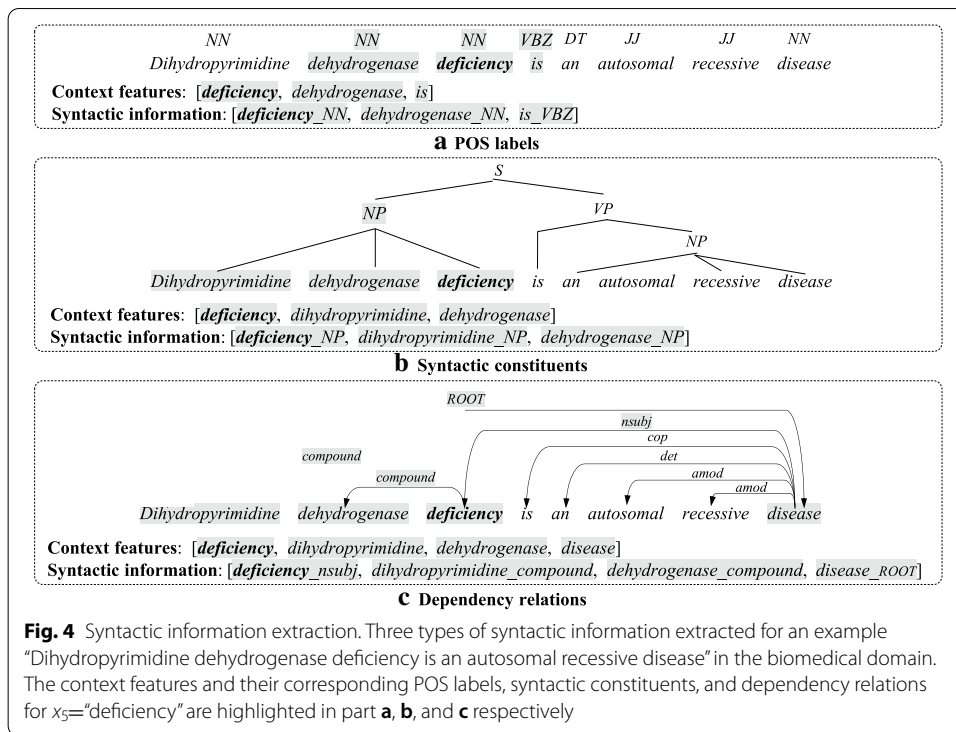
The overall architecture of our BioKMNER is shown in Fig. 3. Following the common approaches in BioNER, we treat it as a sequence labeling task, where the input word sequence $\mathcal{X} = \{x_1, x_2, \dots, x_i, \dots, x_l\}$ is tagged with a sequence of NE labels $\hat{\mathcal{Y}} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_i, \dots, \hat{y}_l\}$. In our method, we propose key-value memory networks (KVMN) [26] to incorporate syntactic information. Specifically, context features and their carrying syntactic information instances are mapped to keys and values in KVMN, where the values are weighed according to the comparison between the keys and the input words.

In this section, we firstly introduce the syntactic information extraction process. Then we elaborate the KVMN module used to incorporate the syntactic information. Finally, we explain how our NER method works with the KVMN module.

Syntactic information extraction

In our study, we focused on three types of syntactic information: POS labels, syntactic constituents, and dependency relations. To obtain such information, we first run the off-the-shelf NLP toolkits on the input sentence \mathcal{X} . Then for each word x_i in \mathcal{X} , we extract the context features associated with x_i and their corresponding syntactic information instances. Figure 4 shows the three types of context features and their corresponding syntactic information instances¹² for the sentence “Dihydropyrimidine dehydrogenase

¹² We combine the word and its syntactic information to increase the diversity of the syntactic information instance.



deficiency is an autosomal recessive disease”¹³ This figure focuses on the word “deficiency” (in boldface) with its highlighted context features and their corresponding syntactic information.

POS labels Given a current word x_i in \mathcal{X} , we use a 1-word window to extract the context words and their POS labels at both sides of x_i . As is shown in Fig. 4a, for word “deficiency”, the context features are [deficiency,dehydrogenase,is] and the corresponding syntactic instances are [deficiency_NN,dehydrogenase_NN,is_VBZ].

Syntactic constituents

First, we define a set of acceptable syntactic nodes (denoted by \mathcal{L}) which contains 10 different constituent types¹⁴ to selected syntactic constituents from the syntax tree of the input \mathcal{X} . Then, for each word x_i in \mathcal{X} , we start with the leaf of x_i in the parse tree, search up to find the first acceptable syntactic node which is in \mathcal{L} . After finding the first acceptable node of x_i , the words under that node and their combination with the node type label are selected as the context features and their corresponding syntactic information respectively. As is shown in Fig. 4b, for word “deficiency”, the first acceptable node is NP, and there are three words under this NP span. So the context features are [deficiency, dihydropyrimidine, dehydrogenase], and the syntactic instances are [deficiency_NP, dihydropyrimidine_NP, dehydrogenase_NP].

¹³ The POS tags and parse trees are obtained from running the online demo of Stanford CoreNLP Toolkits [25] at <https://corenlp.run/>.

¹⁴ The 10 accepted constituent types are NP, VP, PP, ADVP, SBAR, ADJP, PRT, INTJ, CONJP and LST, which are selected from the types used in the CoNLL-2003 shared task [41].

Dependency relations According to the dependency relations of the words in the sentence, we first collect the dependents and the governor of the given word (i.e., first-order dependency relations). Then, we regard its dependents, its governor, and the word itself, as the context features and regard the combination of these words and their dependency types as the syntactic instances. In Fig. 4c, for the given word “deficiency”, it has two dependents (i.e., “dihydropyrimidine” and “dehydrogenase”) and one governor (i.e., “disease”, which is the root of the sentence). According to these dependency relations, the context features of “deficiency” are [deficiency, dihydropyrimidine, dehydrogenase, a, metabolic] and the syntactic information instances are [deficiency_nsubj, dihydropyrimidine_compound, dehydrogenase_compound, disease_ROOT].

Through these processes, the context feature list \mathcal{K} and the syntactic instance list \mathcal{V} are built upon the extraction results for each type of syntactic information. For each word x_i in the word sequence \mathcal{X} , in both training and predicting process, associated context features and syntactic information instances in \mathcal{K} and \mathcal{V} are activated and computed. We denote the context features and the syntactic information instances for x_i as $\mathcal{K}_i = [k_{i,1}, \dots, k_{i,j}, \dots, k_{i,m_i}]$ and $\mathcal{V}_i = [v_{i,1}, \dots, v_{i,j}, \dots, v_{i,m_i}]$, respectively. Note that the context feature list \mathcal{K} and syntactic instance list \mathcal{V} used in our model do not necessarily need to include all three types of the syntactic information discussed above. In other words, our model can leverage each type of syntactic information independently. In the following subsection, we illustrate our method to leverage the keys and values through KVMN.

The memory module

Previous methods to leverage syntactic information for BioNER are limited in concatenating the embeddings of syntactic information instances with the input word embeddings. This method fails to distinguish the useful syntactic instances in a specific context, so that noisy syntactic information may hurt model performance. Therefore, we propose to use KVMN to enhance the incorporation process of syntactic information. Originally, KVMN is firstly proposed to incorporate the information in a list of memory slots (k_j, v_j) (where k_j and v_j refer to keys and values, respectively)¹⁵ into a model for question answering tasks. In KVMN, it addresses the keys by assigning a probability weight to the value in each memory slot by comparing the question (which is denoted as x) to each key:

$$p_j = \text{softmax}(\mathbf{A}\Phi_{\mathcal{X}}(x) \cdot \mathbf{A}\Phi_{\mathcal{K}}(k_j)) \quad (1)$$

where Φ are feature mapping matrices and \mathbf{A} is a matrix. Then, KVMN reads the values by computing the weighted sum using the resulting probability weights:

$$\mathbf{o} = \sum_j p_j \cdot \mathbf{A}\Phi_{\mathcal{V}}(v_j) \quad (2)$$

Afterwards, \mathbf{o} is incorporated into the question representation by an element-wise summation: $\mathbf{o}' = \mathbf{A}\Phi_{\mathcal{X}}(x) + \mathbf{o}$ and the resulting \mathbf{o}' is used to predict the answers of

¹⁵ Here, we use the subscript j instead of i in the original paper to avoid confusion, because i is already used to refer to the input word x_i at the position i .

the question. Therefore, in KVMN, the keys are used to compute the weights, which is used to address the values with respect to the input; the values are used to incorporate useful information into the input presentation and thus improve model performance. Considering that knowledge base entries have been used as a possible type of resources for the memory slots to incorporate extra knowledge into the input representation by transforms between the keys and values [26], we can also use such transforms between context features and syntactic information instances to incorporate the syntactic information into our backbone method. In doing so, not only is the syntactic information addressed by comparing the input with context features (which we think is more intuitive than comparing the input with syntactic information), but also different syntactic information instances are weighed according to the comparison between keys and the input, which allows our method to distinguish the important syntactic information instances and leverage them accordingly.

In our approach to bioNER, we adapt the KVMN to a sequence labeling paradigm by applying it to each word x_i in the input. Therefore, for x_i , its hidden vector \mathbf{h}_i obtained from an encoder serves as the counterpart of input representation $\mathbf{A}\Phi_{\mathcal{X}}(x)$; its associated context features and the corresponding syntactic information instances stand for the keys k_j and values v_j , respectively. In details, the memory module takes \mathbf{h}_i for each x_i , activates the keys to address their embeddings and computes the probability weights for them by

$$p_{i,j} = \frac{\exp(\mathbf{h}_i \cdot \mathbf{e}_{i,j}^k)}{\sum_{j=1}^{m_i} \exp(\mathbf{h}_i \cdot \mathbf{e}_{i,j}^k)} \quad (3)$$

where $\mathbf{e}_{i,j}^k$ is the embedding vector of $k_{i,j}$. Afterwards, we use the resulting probabilities on syntactic information instances in \mathcal{V}_i to get the weighted value embedding \mathbf{o}_i :

$$\mathbf{o}_i = \sum_{j=1}^{m_i} p_{i,j} \mathbf{e}_{i,j}^v \quad (4)$$

where $\mathbf{e}_{i,j}^v$ is the embedding vector of the value $v_{i,j}$. Once \mathbf{o}_i is obtained for each x_i , we concatenate¹⁶ it with \mathbf{h}_i to get the \mathbf{o}'_i , which can be represented by $\mathbf{o}'_i = \mathbf{h}_i \oplus \mathbf{o}_i$.

Tagging with KVMN

To facilitate the process of leveraging syntactic information through KVMN, we firstly use an encoder to obtain the hidden vector \mathbf{h}_i for each x_i . Among different types of encoders, in our method, we use the prevailing BioBERT [19], which is demonstrated to be an effective encoders for many biomedical NLP tasks, such as relation extraction [22] and natural language inference [46]. Therefore, the process to obtain the hidden vectors for the input \mathcal{X} can be represented by

$$[\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_i, \dots, \mathbf{h}_l] = \text{BioBERT}(\mathcal{X}) \quad (5)$$

¹⁶ We use concatenation instead of element-wise summation in KVMN, which does not change the nature of using KVMN.

Once \mathbf{o}'_i is obtained from the KVMN module, we apply a trainable matrix \mathbf{W} to it to align its dimension to the output space, which is formalized by

$$\mathbf{u}_i = \mathbf{W} \cdot \mathbf{o}'_i \quad (6)$$

The resulting vector \mathbf{u}_i is a weight vector with each dimension corresponding to a type of BioNER labels (so its vector dimension matches the number of NE types). Finally, we apply a softmax function to \mathbf{u}_i to predict the output label \hat{y}_i for x_i by

$$\hat{y}_i = \arg \max \frac{\exp(u_i^t)}{\sum_{t=1}^{|\mathcal{T}|} \exp(u_i^t)} \quad (7)$$

where \mathcal{T} refers to the label set and u_i^t is the value at dimension t in the weight vector \mathbf{u}_i .

Abbreviations

BioNER: Biomedical named entity recognition; KVMN: Key-value memory networks; NER: Named entity recognition; NE: Named entity; POS: Part-of-speech.

Acknowledgements

This work is supported by The Chinese University of Hong Kong (Shenzhen) under University Development Fund UDF01001809.

Authors' contributions

YS designs the study, analyzes the results, and write the paper; YT implements the code, analyzes the results, and writes the paper; WS implements the code, performs the experiments, and writes the paper; FX analyzes the results and writes the paper; MH designs the study and analyzes the results; KL performs the experiments. All authors read and approved the final manuscript.

Funding

This work is supported by The Chinese University of Hong Kong (Shenzhen) under University Development Fund UDF01001809.

Availability of data and materials

The datasets generated and/or analysed during the current study are available in the *BioKMNER* repository <https://github.com/cuhk-nlp/BioKMNER>. The code is available at <https://github.com/cuhk-nlp/BioKMNER>.

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 University of Washington, Seattle, USA. ² Hunan University, Changsha, China. ³ The Chinese University of Hong Kong, Shenzhen, China. ⁴ Shenzhen Research Institute of Big Data, Shenzhen, China.

Received: 13 April 2020 Accepted: 23 October 2020

Published online: 25 November 2020

References

1. Akhondi SA, Hettne KM, Van Der Horst E, Van Mulligen EM, Kors JA. Recognition of chemical entities: combining dictionary-based and grammar-based approaches. *J Cheminform*. 2015;7(51):10.
2. Antoun W, Baly F, Hajj H. AraBERT: transformer-based model for Arabic language understanding. arXiv preprint [arXiv:2003.00104](https://arxiv.org/abs/2003.00104) 2020.
3. Beltagy I, Lo K, Cohan A. SciBERT: a pretrained language model for scientific text. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), Hong Kong, China, pp 3606–3611 (2019)
4. Chang F, Guo J, Xu W, Chung SR. Application of word embeddings in biomedical named entity recognition tasks. *J Digit Inf Manag*. 2015; <https://doi.org/10.1155/2020/8894760>.
5. Dang TH, Le H-Q, Nguyen TM, Vu ST. D3NER: biomedical named entity recognition using CRF-biLSTM improved with fine-tuned embeddings of various linguistic information. *Bioinformatics*. 2018;34(20):3539–46.

6. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the Association for Computational Linguistics: human language technologies, vol 1 (Long and Short Papers), 2019; p. 4171–86.
7. Diao S, Bai J, Song Y, Zhang T, Wang Y. ZEN: pre-training chinese text encoder enhanced by N-gram representations. [arXiv: abs/1911.00720](https://arxiv.org/abs/1911.00720); 2019.
8. Doğan Rİ, Leaman R, Lu Z. NCBI disease corpus: a resource for disease name recognition and concept normalization. *J Biomed Inform.* 2014;47:1–10.
9. Gerner M, Nenadic G, Bergman C. Linnaeus: a species name identification system for biomedical literature. *BMC Bioinform.* 2010;11:85.
10. Giorgi JM, Bader GD. Transfer learning for biomedical named entity recognition with neural networks. *Bioinformatics.* 2018;34(23):4087–94.
11. Guan C, Cheng Y, Zhao H. Semantic role labeling with associated memory network. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (long and short papers), Minneapolis, Minnesota, 2019; p. 3361–71.
12. Habibi M, Weber L, Neves M, Wiegandt DL, Leser U. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics.* 2017;33(14):37–48.
13. Jin Q, Dhingra B, Cohen WW, Lu X. Probing biomedical embeddings from language models. [arXiv preprint arXiv:1904.02181](https://arxiv.org/abs/1904.02181); 2019.
14. Kim J-D, Ohta T, Tsuruoka Y, Tateisi Y, Collier N. Introduction to the bio-entity recognition task at JNLPBA. In: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications, 2004; p. 70–5.
15. Le H-Q, Tran M-V, Dang TH, Collier N et al. The UET-CAM system in the BioCreAtivE V CDR task. In: Fifth biocreative challenge evaluation workshop, 2015; p. 208–13.
16. Leaman R, Lu Z. TaggerOne: joint named entity recognition and normalization with semi-Markov models. *Bioinformatics.* 2016;32(18):2839–46.
17. Leaman R, Islamaj Doğan R, Lu Z. DNORM: disease name normalization with pairwise learning to rank. *Bioinformatics.* 2013;29(22):2909–17.
18. Leaman R, Wei C-H, Lu Z. TmChem: a high performance approach for chemical named entity recognition and normalization. *J Cheminform.* 2015;7(1):3.
19. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J. BioBERT: pre-trained biomedical language representation model for biomedical text mining. [arXiv preprint arXiv:1901.08746](https://arxiv.org/abs/1901.08746); 2019.
20. Liao Z, Wu H. Biomedical named entity recognition based on skip-chain CRFs. In: 2012 international conference on industrial control and electronics engineering, 2012; p. 1495–8.
21. Lim S, Lee K, Kang J. Drug drug interaction extraction from the literature using a recursive neural network. *PLoS ONE.* 2018;13(1):0190926.
22. Lin C, Miller T, Dligach D, Bethard S, Savova G. A BERT-based universal model for both within- and cross-sentence clinical temporal relation extraction. In: Proceedings of the 2nd clinical natural language processing workshop, Minneapolis, Minnesota, USA, 2019; p. 65–71.
23. Luo L, Yang Z, Yang P, Zhang Y, Wang L, Lin H, Wang J. An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. *Bioinformatics (Oxford, England).* 2018;34(8):1381.
24. Luo L, Yang Z, Yang P, Zhang Y, Wang L, Lin H, Wang J. An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. *Bioinformatics.* 2017;34(8):1381–8.
25. Manning C, Surdeanu M, Bauer J, Finkel J, Bethard S, McClosky D. The Stanford CoreNLP Natural Language Processing Toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, 2014; p. 55–60.
26. Miller A, Fisch A, Dodge J, Karimi A-H, Bordes A, Weston J. Key-value memory networks for directly reading documents. In: Proceedings of the 2016 conference on empirical methods in natural language processing, 2016; p. 1400–9.
27. Mino H, Utiyama M, Sumita E, Tokunaga T. Key-value attention mechanism for neural machine translation. In: Proceedings of the eighth international joint conference on natural language processing (volume 2: short papers), Taipei, Taiwan, 2017; p. 290–5.
28. Mohit B, Hwa R. Syntax-based semi-supervised named entity tagging. In: Proceedings of the ACL 2005 on interactive poster and demonstration sessions, 2005; p. 57–60.
29. Pafilis E, Frankild SP, Fanini L, Faulwetter S, Pavloudi C, Vasileiadou A, Arvanitidis C, Jensen LJ. The SPECIES and ORGANISMS resources for fast and accurate identification of taxonomic names in text. *PLoS ONE.* 2013;8(6):65390.
30. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L. Deep contextualized word representations. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1–6, 2018, vol 1 (long papers), 2018; p. 2227–37.
31. Sachan DS, Xie P, Sachan M, Xing EP. Effective use of bidirectional language modeling for transfer learning in biomedical named entity recognition. In: Machine learning for healthcare conference (2018); p. 383–402.
32. Sahu S, Anand A. Recurrent neural network models for disease name recognition using domain invariant features. In: Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers), Berlin, Germany, 2016; pp 2216–25.
33. Scozzafava F, Maru M, Brignone F, Torrisi G, Navigli R. Personalized PageRank with syntagmatic information for multilingual word sense disambiguation. In: Proceedings of the 58th annual meeting of the association for computational linguistics: system demonstrations, Online, 2020; pp. 37–46.
34. Segura Bedmar I, Martínez P, Herrero Zazo M. Semeval-2013 Task 9: extraction of drug–drug interactions from biomedical texts (DDIExtraction 2013). 2013.
35. Smith L, Tanabe LK, Nee Ando RJ, Kuo C-J, Chung I-F, Hsu C-N, Lin Y-S, Klinger R, Friedrich CM, Ganchev K, et al. Overview of biocreative II gene mention recognition. *Genome Biol.* 2008;9(2):2.

36. Song Y, Lee C-J, Xia F. Learning word representations with regularization from prior knowledge. In: Proceedings of the 21st conference on computational natural language learning (CoNLL 2017), Vancouver, Canada, (2017); p. 143–52.
37. Song Y, Kim E, Lee GG, Yi B-K. POSBIOTM-NER: a trainable biomedical named-entity recognition system. *Bioinformatics*. 2005;21(11):2794–6.
38. Szklarczyk D, Morris JH, Cook H, Kuhn M, Wyder S, Simonovic M, Santos A, Doncheva NT, Roth A, Bork P, et al. The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Res*. 2016;937(45):D362–8.
39. Tian Y, Song Y, Ao X, Xia F, Quan X, Zhang T, Wang Y. Joint Chinese word segmentation and part-of-speech tagging via two-way attentions of auto-analyzed knowledge. In: Proceedings of the 58th annual meeting of the association for computational linguistics, Online, 2020; pp. 8286–96.
40. Tian Y, Song Y, Xia F, Zhang T, Wang Y. Improving Chinese word segmentation with wordhood memory networks. In: Proceedings of the 58th annual meeting of the association for computational linguistics, online, 2020; pp. 8274–85.
41. Tjong Kim Sang EF, De Meulder F. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: Daelemans W, Osborne M (eds.), Proceedings of CoNLL-2003, 2003; pp. 142–7.
42. Tkachenko M, Simanovsky A. Named entity recognition: exploring features. In: KONVENS, 2012; p. 118–27.
43. Wang X, Zhang Y, Ren X, Zhang Y, Zitnik M, Shang J, Langlotz C, Han J. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*. 2018;35(10):1745–52.
44. Wei C-H, Peng Y, Leaman R, Davis AP, Mattingly CJ, Li J, Wieggers TC, Lu Z (2015) Overview of the biocreative V chemical disease relation (CDR) task. In: Proceedings of the fifth biocreative challenge evaluation workshop, vol. 14
45. Wei Q, Chen T, Xu R, He Y, Gui L. Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks. *Database J Biol Databases Curation* 2016.
46. Wu Z, Song Y, Huang S, Tian Y, Xia F. WTMed at MEDIQA 2019: a hybrid approach to biomedical natural language inference. In: Proceedings of the 18th BioNLP workshop and shared task, Florence, Italy, 2019; p. 415–26.
47. Xie B, Ding Q, Han H, Wu D. MiRCancer: a MicroRNA-cancer association database constructed by text mining on literature. *Bioinformatics*. 2013;29(5):638–44.
48. Yoon W, So CH, Lee J, Kang J. CollaboNet: collaboration of deep neural networks for biomedical named entity recognition. *BMC Bioinform*. 2019;20(10):249.
49. Zhou H, Ning S, Liu Z, Lang C, Liu Z, Lei B. Knowledge-enhanced biomedical named entity recognition and normalization: application to proteins and genes. *BMC Bioinform*. 2020;21(1):35.

Publisher's note

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

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

