

Improving Biomedical Named Entity Recognition with Syntactic Information

YUANHE TIAN

University of Washington <https://orcid.org/0000-0001-6841-2341>

Wang Shen

Hunan University

Yan Song (✉ clksong@gmail.com)

<https://orcid.org/0000-0002-2849-2962>

Fei Xia

University of Washington

Min He

Hunan University

Kenli Li

Hunan University

Research article

Keywords: Machine Learning, Artificial Intelligence, Bioinformatics

Posted Date: April 30th, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-21994/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published on November 25th, 2020. See the published version at <https://doi.org/10.1186/s12859-020-03834-6>.

RESEARCH

Improving Biomedical Named Entity Recognition with Syntactic Information

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

*Correspondence:

clksong@gmail.com

¹University of Washington,
Seattle, USA

³Sinovation Ventures,
Shenzhen, China

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

[†]Equal contributor

Abstract

Background: Biomedical named entity recognition (BioNER) is an important task for understanding biomedical texts. The task can be challenging due to the lack of large-scale labeled training data and domain knowledge. Previous studies have shown that syntactic information can be useful for named entity recognition; however, most of them fail to weigh that information with respect to its contribution as they treat the syntactic information as gold reference.

Results: In this paper, we propose BioKMNER, a BioNER model for biomedical texts with key-value memory networks to incorporate syntactic information, which is extracted from syntactic structures automatically generated by existing toolkits. Our approach outperforms baselines without memories and achieves new state-of-the-art results on four biomedical datasets compared with previous studies, i.e., 85.67% on BC2GM, 94.22% on BC5CDR-chemical, 90.11% on NCBI-disease, and 76.33% on Species-800.

Conclusion: Experimental results on four benchmark datasets demonstrate the effectiveness of our method, where the state-of-the-art performance is achieved on all of them.

Keywords: Named Entity Recognition; Text Mining; Key-value Memory Networks; Syntactic Information; Neural Networks

1 Background

Biomedical named entity recognition (BioNER) is an important yet 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 [14, 23] and knowledge base completion [27, 34]. Compared to named entity recognition in the general domain, BioNER is considered more difficult due to the lack of large-scale labeled training data and domain knowledge.

In the past decade, there have been many studies on BioNER, ranging from traditional feature based methods [2, 8–11, 13, 26] to recent deep learning based methods [4, 7, 12, 16, 22, 32]. Syntactic information has been playing an important role in many of these studies to help recognize biomedical named entities [2, 4, 13, 16, 26]. The reason for that is that biomedical text often includes formal, well-structured, long sentences, where syntactic information can provide useful cues for BioNER and it is demonstrated that leveraging extra information can help with the text understanding of NLP systems [25]. Figure 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.

Previous studies have tried to incorporate syntactic information into the BioNER system in several ways. In the feature engineering methods, researchers use syntactic information to generate handcrafted features to help BioNER. For example, Song et al. [26] used POS and noun phrase tag features in a CRF-based BioNER system. In recent deep learning based methods, syntactic information is represented by vectorized embedding, which is combined with word embedding through vector concatenation or element-wise summation; the combined vector is then fed into neural models to improve bioNER. For example, Luo et al. [16] 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 attention mechanism to detect chemical NE. Dang et al. [4] proposed a model named D3NER, where the embeddings of various informative syntactic information are used to improve the results. While those previous studies demonstrated the potential benefits of incorporating syntactic information, they were limited in either treating noisy syntactic information as gold references for training their taggers or using direct concatenation to combine that information with context information without weighing the information with respect to its contribution to the BioNER task. The goal of our study is to find a way to better incorporate different types of syntactic information into neural models for NER in the biomedical domain.

In this paper, we propose a BioNER model, named BioKMNER (KM stands for Key-value Memory network), which uses key-value memory networks (KVMN) [18] to incorporate syntactic information into the backbone sequence labeling based tagger to improve BioNER. Firstly, because there is not a lot of syntactically annotated biomedical text available to the public, we use off-the-shelf toolkits to parse biomedical text sentences and extract three types of syntactic information; namely, part-of-speech (POS) labels, syntactic constituents, and dependency relations. Then, the context features and syntactic information generated by the toolkits are mapped to keys and values in KVMN. The KVMN weighs the corresponding syntactic information (values) according to the importance of context features (keys), and combines the weighted syntactic information with the output of the encoder. Finally, the decoder receives the combined embedding and tags the input sequence accordingly. Through this way, the model can incorporate the pair-wise organized context features and syntactic information obtained from the toolkits simultaneously. Different from previous studies that directly use noisy syntactic information and treat them as gold references, our BioKMNER weighs them in KVMN and thus reduces the effect of error propagation caused by incorrect parsing results. The results on four commonly used BioNER datasets with each of them focusing on one specific entity type (i.e., chemical, disease, gene/protein, and species) demonstrate the effectiveness of our model in recognizing different types of named entities, where BioKMNER achieves state-of-the-art results on all of them.

2 Results

2.1 Datasets

We evaluate our methods on four benchmark datasets that are widely used in previous studies [6, 7, 9, 12, 35]. Each dataset focuses on one specific biomedical entity type: BC2GM dataset [24] for gene/protein NER, BC5CDR-chemical dataset [31] for chemical NER, NCBI-disease dataset [5] for disease NER, and Species-800 [20] for species NER.

BC2GM BC2GM is a dataset used for 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 NER of gene/proteins entity type [6, 7, 12, 21, 30, 35].

BC5CDR-chemical BC5CDR is a dataset used for the BioCreative V Chemical Disease Relation (CDR) Task^[2]. It contains 1,500 titles and abstracts from PubMed^[3], where chemical and disease mentions are annotated by human. Following previous studies [16], we only use the subset with chemical entities and denote it as BC5CDR-chemical.

NCBI-disease NCBI-disease contains 793 PubMed abstracts which 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 were mapped to a single disease concept. It has been a widely used research resource for the disease NER.

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

Following Lee et al. [12], we obtain BC2GM, BC5CDR-chemical, and NCBI-disease datasets from Wang et al. [30], and Species-800 dataset from Pyysalo^[4]. The statistics of all datasets with respect to the number of tokens, sentences, and entities are reported in table 1.

2.2 Implementation

Because the amount of syntactically annotated biomedical text is limited, we use off-the-shelf NLP toolkits to generate syntactic information, following the common practice in previous studies such as Mohit and Hwa [19], Tkachenko and Simanovsky [29], and Luo et al. [16]. In our study, we focused on three types of syntactic information: POS labels, syntactic constituents and dependency relations, which are

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

^[2]<https://biocreative.bioinformatics.udel.edu/tasks/biocreative-v/task-3-cdr/>

^[3]<https://pubmed.ncbi.nlm.nih.gov/>

^[4]<https://github.com/spyyalo/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
BC5CDR-chemical	Chemical	Train	118.1K	4.5K	4.1K
		Dev	117.4K	4.5K	4.2K
		Test	124.7K	4.7K	4.4K
NCBI-disease	Disease	Train	135.7K	5.4K	5.1K
		Dev	23.9K	923	787
		Test	24.4K	940	960
Species-800	Species	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.

obtained from the tagging and parsing results of Stanford CoreNLP Toolkits^[5] [17] for a given input sentence.

For the encoder in the sequence labeling based backbone model, we use both the base and large version of bioBERT^[6] and follow the hyper-parameters used by Lee et al. [12] (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. In addition, we follow the setting of Lee et al. [12] to run the training process and evaluate all models based on the micro F1 scores with respect to entity-level exact matches.^[7]

2.3 Overall results

We run experiments on four aforementioned datasets with each of them focusing on one specific entity type (i.e., chemical, disease, gene/protein, and species), where two encoders, i.e., BioBERT-Base and BioBERT-Large, are used. To enhance the baselines, we try three types of syntactic information with two incorporation methods, i.e., direct concatenation and KVMN, and report the results in Table 2. There are some observations.

First, our model with KVMN works well with both bioBERT-base and bioBERT-large, where improvements over the baseline models are observed among all datasets. This demonstrates the validity and effectiveness of the method to improve NER for biomedical texts. Specifically, among all datasets, the highest improvement is observed on the entity type of species (i.e., the Species-800 dataset), where BioKMN-NER with dependency relations improves the BioBERT-Large baseline by 0.91% with respect to the F1 score.

Second, in models with KVMN, the information of syntactic constituents and dependency relations offer higher improvement than POS labels. For example, on

^[5]We use v3.9.2, downloaded from <https://stanfordnlp.github.io/CoreNLP/>.

^[6]We obtain the pre-trained models v1.1 from <https://github.com/naver/biobert-pretrained>

^[7]We evaluate all models by the widely used seqeval framework at <https://github.com/chakki-works/seqeval>.

Table 2 Experimental results of models on four benchmark datasets.

Models	BC2			BC5			NC			SP		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Base	84.64	84.57	84.60	92.88	93.96	93.42	88.09	91.20	89.60	72.15	78.36	75.12
+ PL (DC)	83.93	85.20	84.56	93.57	94.02	93.79	88.82	91.04	89.92	72.07	79.40	75.58
+ SC (DC)	83.97	85.28	84.62	93.90	93.50	93.70	88.75	90.42	89.58	73.94	77.31	75.59
+ DR (DC)	83.97	85.38	84.67	93.18	93.91	93.54	88.57	90.42	89.48	74.37	76.80	75.56
+ PL ($\mathcal{M}$)	83.90	85.63	84.76	93.31	94.28	93.79	88.52	91.56	90.02	72.92	79.01	75.84
+ SC ($\mathcal{M}$)	84.46	85.45	84.96	93.65	94.15	93.90	88.22	92.08	90.11	72.86	79.79	76.17
+ DR ($\mathcal{M}$)	85.06	84.79	84.92	94.10	93.89	94.00	88.81	91.77	90.08	73.49	79.14	76.21
Large	84.35	85.39	84.87	93.05	94.22	93.63	88.43	89.17	88.80	72.82	78.23	75.43
+ PL (DC)	85.04	85.03	85.03	93.20	94.41	93.80	86.91	90.63	88.73	69.61	79.14	74.06
+ SC (DC)	85.22	85.04	85.13	92.97	95.04	93.99	88.82	90.21	89.51	71.15	80.05	75.33
+ DR (DC)	84.47	85.39	84.92	93.36	94.50	93.93	87.98	90.73	89.33	69.26	80.18	74.32
+ PL ($\mathcal{M}$)	84.86	85.50	85.18	93.76	94.06	93.91	87.33	91.15	89.19	73.81	79.01	76.32
+ SC ($\mathcal{M}$)	84.76	86.59	85.67	93.98	94.24	94.11	88.34	90.73	89.52	73.14	79.53	76.20
+ DR ($\mathcal{M}$)	85.06	85.15	85.11	94.73	93.72	94.22	89.26	90.00	89.63	72.73	80.31	76.33

The results are reported in terms of precision (P), recall (R) and F1-score (F1). The models in the group “Base” and “Large” refer to baselines with BioBERT-base and BioBERT-large encoder and our models with three types of syntactic information, i.e., POS labels (PL), syntactic constituents (SC), and dependency relations (DR) incorporated by direct concatenation (DC) and KVMN ($\mathcal{M}$) on all datasets.

BC5CDR-chemical dataset, BioKMNER with the BioBERT-Large encoder obtains the F1 scores of 94.11% and 94.22% when it is enhanced by syntactic constituents and dependency relations, respectively, while its F1 score is 93.91% when POS labels are incorporated. The reasons to explain the phenomenon that could be: (1) the syntactic constituents provide a cue of phrase functions and their boundaries (e.g., an NP treelet is not only the signal suggesting there can be an NE inside, but also gives information about the possible start and end positions for it.); (2) the dependency relations link words in long distance with their dependency relationships, which is especially useful for biomedical texts that generally include long sentences and entities.

Third, compared with the way of incorporating syntactic information through direct concatenation (DC), models with KVMN achieve better results. For example, on BC5CDR-chemical dataset, our model (Large + DR ($\mathcal{M}$)) obtains F1 score of 94.22%, while its corresponding DC-based method (Large + DR (DC)) obtains a lower F1 score of 93.93%. Besides, in some cases where DC is applied, the syntactic information causes inferior results than baselines. For example, on NCBI-disease dataset, the results of the DC-based model with the dependency information are lower than the baseline (Base) results. The explanation is that: 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, KVMN can weigh syntactic information according to the importance of the context features, while directly concatenating the embeddings of syntactic information and the context features with the BioBERT output fails to distinguish the salient syntactic information that contributes more to the bioNER task. Therefore, models with KVMN can effectively avoid the error spread caused by the direct concatenation way to incorporate auto-processed syntactic information.

Table 3 Comparison with previous studies.

Models	BC2			BC5			NC			SP		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Leaman et al. [10]	-	-	-	-	-	-	82.80	81.90	80.90	-	-	-
Leaman and Lu [9]	-	-	-	94.20	88.80	91.40	85.10	80.80	82.90	-	-	-
Habibi et al. [7]	78.99	78.16	78.57	92.18	89.94	91.05	86.43	82.92	84.64	77.66	69.08	73.11
Luo et al. [16]	-	-	-	93.49	91.68	92.57	-	-	-	-	-	-
Wang et al. [30]	82.10	79.42	80.74	-	-	-	85.86	86.42	86.14	-	-	-
Sachan et al. [21]	81.81	81.57	81.69	-	-	-	86.41	88.31	87.34	-	-	-
Giorgi and Bader [6]	78.62	78.71	78.66	92.70	90.62	91.64	82.62	86.95	84.72	74.34	75.96	74.98
Cho and Lee [3]	-	-	-	-	-	-	84.73	86.67	85.68	-	-	-
Yoon et al. [35]	80.49	78.99	79.73	94.26	92.38	93.31	85.48	87.27	86.36	-	-	-
Beltagy et al. [1]	-	-	-	-	-	-	-	-	88.57	-	-	-
Lee et al. [12]	84.78	85.25	85.01	93.68	93.26	93.47	88.22	91.25	89.71	72.84	77.97	75.31
Our Model	85.06	86.59	85.67	94.73	93.72	94.22	88.22	92.08	90.11	72.73	80.31	76.33

The results of our model is from the best performed model in Table 2.

3 Discussion

3.1 Comparison with previous studies

We compare the results of our best preformed model with previous studies on all benchmark datasets. The results are summarized in table 3, where our model BioKMNER outperforms all previous studies and obtains new state-of-the-art results on all datasets, because our model can discriminate the important syntactic information instances that contribute much to BioNER. Specifically, the increase of F1 scores on BC2GM, BC5CDR-chemical, NCBI-disease, and Species-800 are 0.66%, 0.75%, 0.40%, 1.02%, respectively. These results demonstrate the validity and effectiveness of our model BioKMNER in recognizing different types of named entities in the biomedical domain.

3.2 Effect of syntactic information ensemble

To compare the performance of different syntactic information ensemble, we conduct additional experiments on BC5CDR-chemical dataset. In the experiments, we test different combinations of different types of syntactic information and use two methods to do so: (1) summing the weighted value embeddings of each type of syntactic information; and (2) concatenating the weighted value embeddings of each type of syntactic information. The results are reported in table 3.2, and we have several observations from it. First, the overall results shows that for both base and large version of the bioBERT encoder, BioKMNER can appropriately combine different types of syntactic information to enhance the model performance on BioNER; and the more types of syntactic information are leveraged, the better performance can be observed. For example, the ensemble of constituent and dependency information obtains a higher result than only using one of them (see Table 2); the ensemble of all types of syntactic information outperforms all other combinations of syntactic information. Second, the concatenating method performs better than the summing method in syntactic information fusion. The explanation can be straightforward: summing the embeddings of different types of syntactic information may lose some information while concatenating can keep all information on all types of syntactic embedding.

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

Models	Syntactic Info.			Sum			Concatenation		
	PL	SC	DR	P	R	F1	P	R	F1
BioBERT-Base	✓	✓		93.14	94.78	93.95	93.71	94.39	94.05
	✓		✓	94.27	93.80	94.03	93.33	94.84	94.08
		✓	✓	93.32	94.76	94.04	93.09	95.08	94.07
	✓	✓	✓	93.80	94.39	94.09	93.87	94.47	94.16
BioBERT-Large	✓	✓		93.77	94.50	94.14	93.92	94.41	94.17
	✓		✓	94.01	94.47	94.23	93.35	95.19	94.26
		✓	✓	94.11	94.35	94.23	93.71	94.91	94.30
	✓	✓	✓	93.69	94.86	94.27	93.55	95.12	94.33

Sum and concatenation are two ensemble strategies applied to our model.

3.3 Case study

To better understand the effectiveness of BioKMNER to improve BioNER, we use two examples from the datasets to show the way of syntactic constituents and dependency relations to improve bioNER in Figure 2(a) and 2(b), 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).

Syntactic constituents In the example sentence shown in Figure 2(a), 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 key-value memory module weighs “abnormalities” and its carrying syntactic information of “NP” with high values among all other syntactic instances since they are strong signals for disease names, and therefore assigns the correct NE labels to “SEP”. Likewise, the situation for “pure” is on the opposite and thus receives lower weights than other words.

Dependency relations In addition, we also visualize the weights assigned to dependency relations for the word “dystrophy” in another example sentence in Figure 2(b). In this case, dependency information successfully finds the dependents (i.e., “Myotonic”, “DM”, and “disorder”) of “dystrophy” and links those dependents and “dystrophy” with the dependency relations between them. Among those dependents, KVMN distinguishes that the dependent “discorder” 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 weighted by the KVMN, and is assigned the second highest weight 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 long-distance; dependency information is able to capture that information and helps our model distinguish the NE tag for the word “dystrophy”.

4 Conclusion

In this paper, we propose BioKMNER, a model 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 instances are mapped into keys and values, respectively. The values are weighted according to the importance of the keys and the weighted averaged values are fed back to the backbone tagging process to improve BioNER. In doing so, compared to previous studies that regard syntactic information as gold references and treat them equally, BioKMNER can discriminately learn the auto-processed syntactic information and avoid the error spread caused by the direct use of auto-processed syntactic results. The experimental results on four benchmark datasets demonstrate the effectiveness of our model where the state-of-the-art results are observed on all datasets.

5 Methods

The overall architecture of BioKMNER is shown in Figure 3. Following the common approach in BioNER, we treat BioNER 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 $\mathcal{Y} = \{y_1, y_2, \dots, y_i, \dots, y_l\}$. In BioKMNER, the words in $\mathcal{X}$ are converted into hidden vectors through BioBERT [12], and then the key-value memory networks (KVMN) [18] weigh the syntactic information in values according to the importance of keys and feed the weighted syntactic information back to the tagging process. In this section, we firstly introduce the syntactic information extraction process. Then we explain in detail how syntactic information is incorporated into the backbone tagger to enhance BioNER.

5.1 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 instances. Figure 4 shows the three types of context features and their corresponding syntactic instances^[8] for the sentence “Dihydropyrimidine dehydrogenase deficiency is an autosomal recessive disease”.^[9] This figure focuses on the word

^[8]We combine the word and its syntactic information to increase the diversity of the syntactic instance.

^[9]The POS tags and parse trees are obtained from running the online demo of Stanford CoreNLP Toolkits [17] at <https://corenlp.run/>.

“deficiency” (in bold) 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 figure 4(a), 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 types^[10] 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 figure 4(b), 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].

Dependency relations: According to the dependency relations of the words in the sentence, we collect the dependents and the governor of the given word (i.e., first order dependency relations), and regard them, as well as the word itself, as the context features and the combination of these words and their dependency types as the syntactic instances. In figure 4(c), 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 relations, the contextual features of “deficiency” are [deficiency, dihydropyrimidine, dehydrogenase, a, metabolic], the syntactic instances are [deficiency_nsubj, dihydropyrimidine_compound, dehydrogenase_compound, disease_ROOT].

Through these processes, the context features $\mathcal{K}$ and the syntactic instance set $\mathcal{V}$ are built upon the extraction results for each type of syntactic information. For each word x_i in word sequence $\mathcal{X}$, in the both training and predicting process, associated context features and syntactic instances in $\mathcal{K}$ and $\mathcal{V}$ are activated and computed, which are denoted by $\mathcal{K}_i = [k_{i,1}, \dots, k_{i,j}, \dots, k_{i,m_i}]$ and $\mathcal{V}_i = [v_{i,1}, \dots, v_{i,j}, \dots, v_{i,m_i}]$, respectively. Note that the context feature set $\mathcal{K}$ and syntactic instance set $\mathcal{V}$ used in our model do not necessarily need to include all three types of the syntactic

^[10]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 [28].

information discussed above. In other words, our model can leverage each type of syntactic information independently.

5.2 Information incorporation

In this subsection, we illustrate the way to leverage the context features $\mathcal{K}$ and syntactic instances $\mathcal{V}$ obtained from the syntactic information extraction process. Firstly, we use BioBERT [12], an effective encoders for many biomedical NLP tasks, such as relation extraction [15] and natural language inference [33], to generate the hidden vectors $\mathbf{h}_i$ of each word x_i in the input $\mathcal{X}$, which is represented by

$$[\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_i, \dots, \mathbf{h}_l] = \text{BioBERT}(\mathcal{X}) \quad (1)$$

Next, we incorporate the obtained context features $\mathcal{K}_i$ and syntactic instances $\mathcal{V}_i$ for x_i into our model through the syntactic information incorporation process and then obtain the vectorized embeddings $\mathbf{o}_i$ that contain the information from the BioBERT output, the context features, and their corresponding syntactic instances. Afterwards, we use a trainable matrix $\mathbf{W}$ to align the dimension of $\mathbf{o}_i$ to the output space by

$$\mathbf{u}_i = \mathbf{W} \cdot \mathbf{o}_i \quad (2)$$

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

$$y_i^* = \arg \max \frac{\exp(u_i^t)}{\sum_{t=1}^{|\mathcal{T}|} \exp(u_i^t)} \quad (3)$$

where $\mathcal{T}$ refers to the set of all NE label types and u_i^t denotes the value at dimension t in the weight vector $\mathbf{u}_i$. As for the aforementioned syntactic information incorporation process, which is the core part of our model, we proposed two different methods to get $\mathbf{o}_i$. The detail of the two methods are illustrated as follows.

Direct concatenation based syntactic information incorporation In most previous deep learning based BioNER methods that use the syntactic information, the syntactic information is represented by vectorized embeddings and they are concatenated or element-wise summed with the word embeddings in the input sentence [4, 16]. Similarly, we firstly use two matrices to map the context features and their corresponding syntactic instances to their embedding vectors. Then, for each word x_i , we get the embeddings of its context features $\mathcal{K}_i$ (denoted by $\mathbf{e}_{i,j}^k$ for $k_{i,j} \in \mathcal{K}_i$) and the embeddings of the corresponding syntactic instances $\mathcal{V}_i$ (denoted by $\mathbf{e}_{i,j}^v$ for $v_{i,j} \in \mathcal{V}_i$). Afterwards, we element-wisely average the context feature embeddings and the syntactic instance embeddings separately, and concatenate them with the hidden vector $\mathbf{h}_i$ obtained from the bioBERT encoder (See Eq. (1)) to obtain $\mathbf{o}_i$ by

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

Memory based syntactic information incorporation Normally, directly concatenating context features and syntactic information to hidden vectors fails to distinguish the useful syntactic instances, which in most cases contribute more than other features to detecting NEs. Therefore, we propose to use KVMN [18] to enhance the incorporation process of the syntactic information. Originally, KVMN is proposed to incorporate a list of knowledge pairs (k_j, v_j) (k_j and v_j refer to keys and values, respectively) extracted from external resources, such as knowledge bases, into a classification task, formalized by

$$\mathbf{v}' = \sum_j (\mathbf{e}_j^k \cdot \mathbf{e}^x) \cdot \mathbf{e}_j^v \quad (5)$$

where $\mathbf{e}_j^k$ and $\mathbf{e}_j^v$ are embeddings for key (k_j) and value (v_j), respectively, and $\mathbf{e}^x$ is the embedding of the input sample. Afterwards, $\mathbf{v}'$ is incorporated into the input embedding $\mathbf{e}^x$ by element-wise summation: $\mathbf{e}^{x'} = \mathbf{e}^x + \mathbf{v}'$. In doing so, the knowledge carried by values are weighted and incorporated into the backbone classifier through the transition between keys and values, which can finally enhance the model performance.

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 $\mathbf{h}_i$ serves as the counterpart of input embedding $\mathbf{e}^x$ and its associated context features and the corresponding syntactic instances stand for the keys k_j and values v_j , respectively. In details, the memory module takes $\mathbf{h}_i$ from BioBERT for each x_i , activates the keys in $\mathcal{K}_i$ to address their embeddings and computes the probability distribution for them by

$$p_{i,j} = \frac{\exp(\mathbf{h}_i \cdot \mathbf{e}_{i,j}^k)}{\sum_{j=1}^{m_i} \exp(\mathbf{h}_i \cdot \mathbf{e}_{i,j}^k)} \quad (6)$$

Afterwards, we use the resulted probabilities on syntactic instances in $\mathcal{V}_i$ to get the weighted value embedding $\mathbf{c}_i$:

$$\mathbf{c}_i = \sum_{j=1}^{m_i} p_{i,j} \mathbf{e}_{i,j}^v \quad (7)$$

The weighted value embedding $\mathbf{c}_i$ contains information of context features and syntactic information. Once $\mathbf{c}_i$ is obtained for each x_i , we concatenate^[11] it with $\mathbf{h}_i$ to get the $\mathbf{o}_i$, which can be represented by $\mathbf{o}_i = \mathbf{h}_i \oplus \mathbf{c}_i$.

List of abbreviations

BioNER: Biomedical named entity recognition

KVMN: key-value memory networks

NER: named entity recognition

NE: named entity

POS: part-of-speech

^[11]We use concatenation instead of element-wise summation in KVMN, which does not change the nature of using KVMN.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Availability of data and materials

The datasets generated and/or analysed during the current study are available in the bmc_BioInfo_NER repository https://github.com/azxcvsdsqen/bmc_BioInfo_NER.

Competing interests

The authors declare that they have no competing interests.

Funding

Not applicable.

Author's 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.

Acknowledgements

Not applicable.

Author details

¹University of Washington, Seattle, USA. ²Hunan University, , Changsha, China. ³Sinovation Ventures, Shenzhen, China.

References

1. 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)
2. Chang, F., Guo, J., Xu, W., Chung, S.R.: Application of Word Embeddings in Biomedical Named Entity Recognition Tasks. *Journal of Digital Information Management* 13(5) (2015)
3. Cho, H., Lee, H.: Biomedical Named Entity Recognition Using Deep Neural Networks with Contextual Information. *BMC bioinformatics* 20(1), 735 (2019)
4. Dang, T.H., Le, H.-Q., Nguyen, T.M., Vu, S.T.: D3NER: Biomedical Named Entity Recognition using CRF-biLSTM Improved with Fine-tuned Embeddings of Various Linguistic Information. *Bioinformatics* 34(20), 3539–3546 (2018)
5. Dogan, R.I., Leaman, R., Lu, Z.: NCBI Disease Corpus: A Resource for Disease Name Recognition and Concept Normalization. *Journal of biomedical informatics* 47, 1–10 (2014)
6. Giorgi, J.M., Bader, G.D.: Transfer Learning for Biomedical Named Entity Recognition with Neural Networks. *Bioinformatics* 34(23), 4087–4094 (2018)
7. Habibi, M., Weber, L., Neves, M., Wiegandt, D.L., Leser, U.: Deep Learning with Word Embeddings Improves Biomedical Named Entity Recognition. *Bioinformatics* 33(14), 37–48 (2017)
8. Le, H.-Q., Tran, M.-V., Dang, T.H., Collier, N., et al.: The UET-CAM System in the BioCreAtIvE V CDR Task. In: Fifth BioCreative Challenge Evaluation Workshop, pp. 208–213 (2015)
9. Leaman, R., Lu, Z.: TaggerOne: Joint Named Entity Recognition and Normalization with Semi-Markov Models. *Bioinformatics* 32(18), 2839–2846 (2016)
10. Leaman, R., Islamaj Dogan, R., Lu, Z.: DNorm: Disease Name Normalization with Pairwise Learning to Rank. *Bioinformatics* 29(22), 2909–2917 (2013)
11. Leaman, R., Wei, C.-H., Lu, Z.: TmChem: a High Performance Approach for Chemical Named Entity Recognition and Normalization. *Journal of cheminformatics* 7(1), 3 (2015)
12. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: BioBERT: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746* (2019)
13. Liao, Z., Wu, H.: Biomedical Named Entity Recognition Based on Skip-chain Crfs. In: 2012 International Conference on Industrial Control and Electronics Engineering, pp. 1495–1498 (2012)
14. Lim, S., Lee, K., Kang, J.: Drug Drug Interaction Extraction from the Literature using a Recursive Neural Network. *PloS one* 13(1), 0190926 (2018)
15. 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, pp. 65–71 (2019)
16. 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)* 34(8), 1381 (2018)
17. 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, pp. 55–60 (2014)
18. 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, pp. 1400–1409 (2016)

19. Mohit, B., Hwa, R.: Syntax-based Semi-Supervised Named Entity Tagging. In: Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions, pp. 57–60 (2005)
20. Pafilis, E., Frankild, S.P., Fanini, L., Faulwetter, S., Pavloudi, C., Vasileiadou, A., Arvanitidis, C., Jensen, L.J.: The SPECIES and ORGANISMS Resources for Fast and Accurate Identification of Taxonomic Names in Text. PLoS One 8(6), 65390 (2013)
21. Sachan, D.S., Xie, P., Sachan, M., Xing, E.P.: Effective Use of Bidirectional Language Modeling for Transfer Learning in Biomedical Named Entity Recognition. In: Machine Learning for Healthcare Conference, pp. 383–402 (2018)
22. 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, pp. 2216–2225 (2016)
23. Segura Bedmar, I., Martínez, P., Herrero Zazo, M.: Semeval-2013 Task 9: Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013). (2013)
24. Smith, L., Tanabe, L.K., Ando, R.J., Kuo, C.-J., Chung, I.-F., Hsu, C.-N., Lin, Y.-S., Klinger, R., Friedrich, C.M., Ganchev, K., et al.: Overview of Biocreative II Gene Mention Recognition. Genome biology 9(2), 2 (2008)
25. 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, pp. 143–152 (2017)
26. Song, Y., Kim, E., Lee, G.G., Yi, B.-K.: POSBIOTM—NER: a Trainable Biomedical Named-entity Recognition System. Bioinformatics 21(11), 2794–2796 (2005)
27. Szklarczyk, D., Morris, J.H., Cook, H., Kuhn, M., Wyder, S., Simonovic, M., Santos, A., Doncheva, N.T., Roth, A., Bork, P., et al.: The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. Nucleic acids research, 937 (2016)
28. Tjong Kim Sang, E.F., 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, pp. 142–147 (2003)
29. Tkachenko, M., Simanovsky, A.: Named Entity Recognition: Exploring Features. In: KONVENS, pp. 118–127 (2012)
30. 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 35(10), 1745–1752 (2018)
31. Wei, C.-H., Peng, Y., Leaman, R., Davis, A.P., Mattingly, C.J., Li, J., Wieggers, T.C., Lu, Z.: Overview of the Biocreative V Chemical Disease Relation (cdr) Task. In: Proceedings of the Fifth BioCreative Challenge Evaluation Workshop, vol. 14 (2015)
32. 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: The Journal of Biological Databases and Curation 2016 (2016)
33. 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, pp. 415–426 (2019)
34. Xie, B., Ding, Q., Han, H., Wu, D.: MiRCancer: a MicroRNA–cancer Association Database Constructed by Text Mining on Literature. Bioinformatics 29(5), 638–644 (2013)
35. Yoon, W., So, C.H., Lee, J., Kang, J.: CollaboNet: Collaboration of Deep Neural Networks for Biomedical Named Entity Recognition. BMC bioinformatics 20(10), 249 (2019)

Additional Files

1_background_example.png — An example sentence

An example where the object noun phrase (“Huntington disease”) is a named entity. The labels under the words are BIO tags.

2_case_study.png — Sample additional file title

In the figure, (a) and (b) are two examples of syntactic information (i.e., syntactic constituents and dependency relations) and the context features for “SEP” and “dystrophy”, respectively. The weights for syntactic information learned from the memories are highlighted with the darker color referring to greater value.

3_model.png — Sample additional file title

The part top of the figure shows the syntactic information extraction process: for the input word sentence, we use off-the-shelf NLP toolkits to obtain its syntactic information (e.g., syntax tree), map the context features and the syntactic information into keys and values, and convert them into embeddings. The bottom part is our sequence labeling based BioNER tagger, which uses BioBERT [12] as the encoder and a softmax layer as the decoder. Between the encoder and decoder are the memory networks which weigh syntactic information (values) according to the importance of the context features (keys), and whose output is fed into the decoder to predict output labels.

4_syntactic_info.png — Syntactic information extraction

Three types of syntactic information extracted for an example “Dihydropyrimidine dehydrogenase deficiency is an autosomal recessive disease” in the biomedical domain. The context features and their corresponding POS labels, syntactic constituents, and dependency relations for x_5 = “deficiency” are highlighted in part (a), (b) and (c) respectively.

Figures

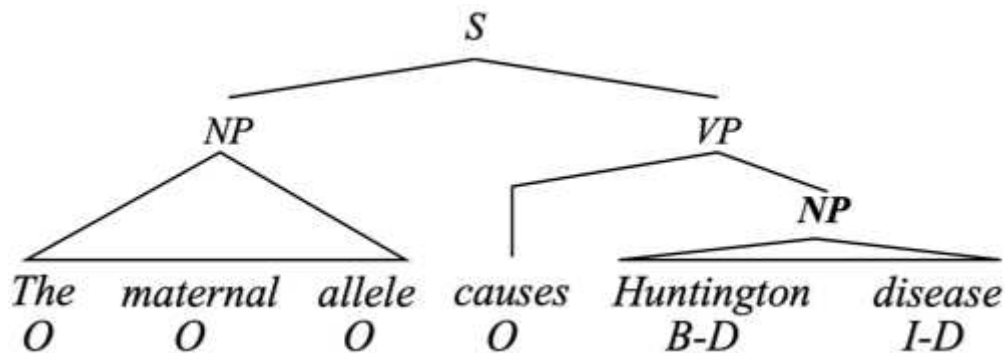


Figure 1

An example sentence An example where the object noun phrase ("Huntington disease") is a named entity. The labels under the words are BIO tags.

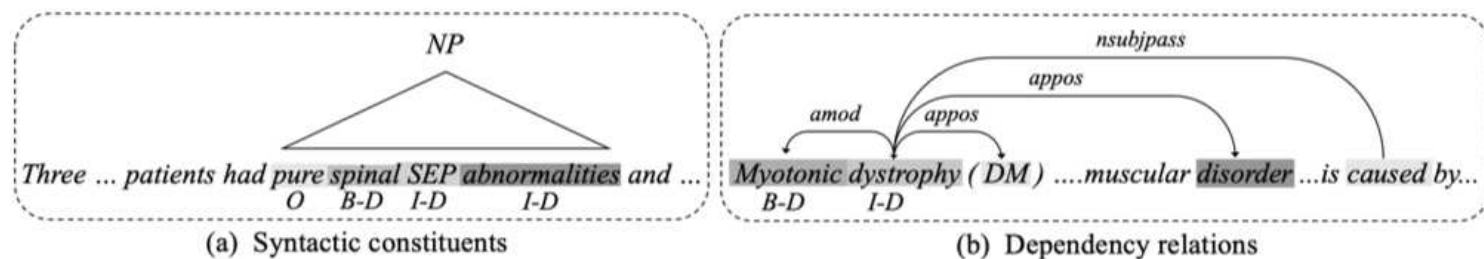


Figure 2

Case study. In the figure, (a) and (b) are two examples of syntactic information (i.e., syntactic constituents and dependency relations) and the context features for "SEP" and "dystrophy", respectively. The weights for syntactic information learned from the memories are highlighted with the darker color referring to greater value.

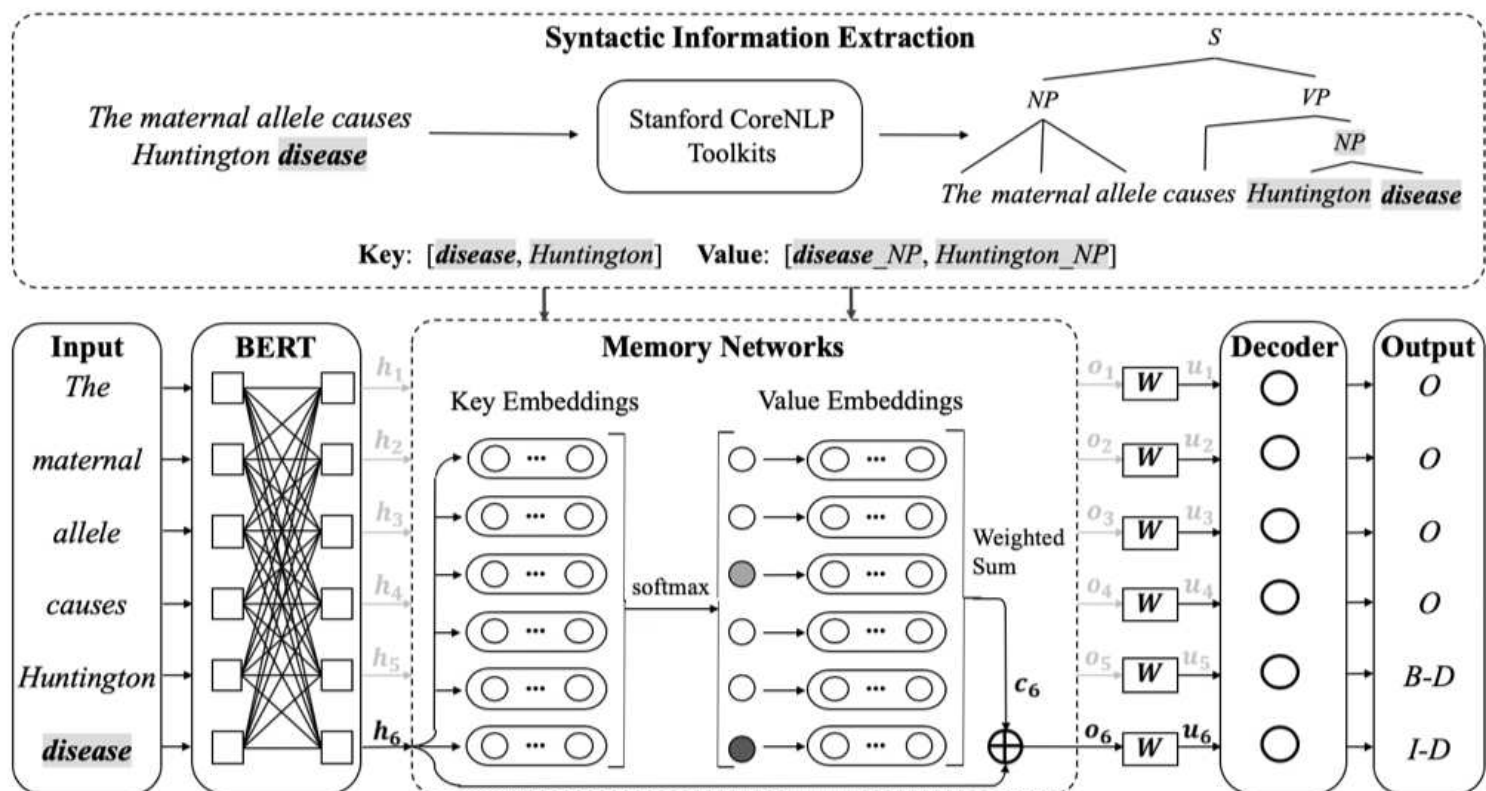


Figure 3

The overall architecture of BioKMNER. The top part of the figure shows the syntactic information extraction process: for the input word sentence, we use off-the-shelf NLP toolkits to obtain its syntactic information (e.g., syntax tree), map the context features and the syntactic information into keys and values, and convert them into embeddings. The bottom part is our sequence labeling based BioNER tagger, which uses BioBERT [12] as the encoder and a softmax layer as the decoder. Between the encoder and decoder are the memory networks which weigh syntactic information (values) according to the importance of the context features (keys), and whose output is fed into the decoder to predict output labels.

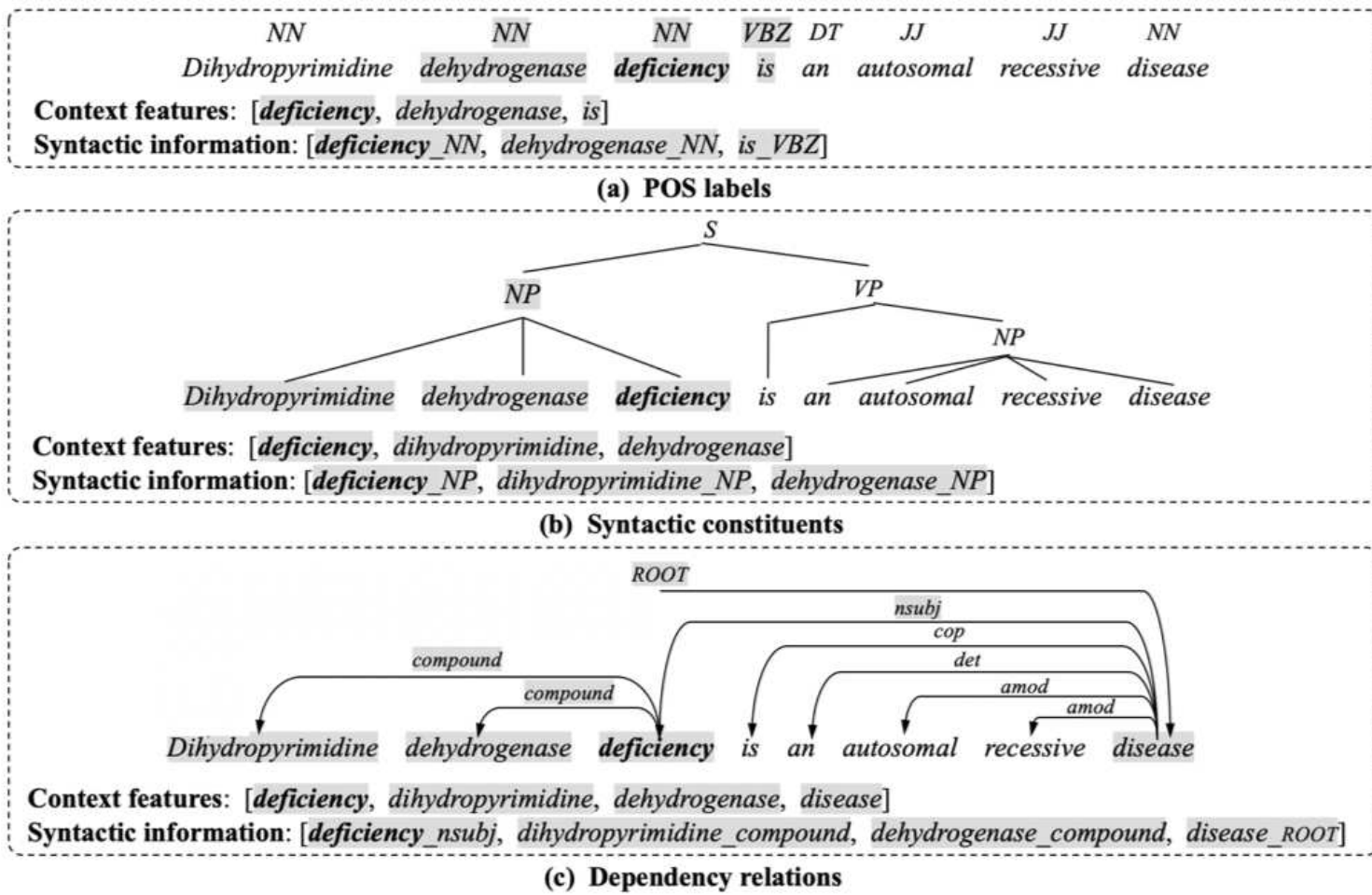


Figure 4

Syntactic information extraction Three types of syntactic information extracted for an example “Dihydropyrimidine dehydrogenase deficiency is an autosomal recessive disease” in the biomedical domain. The context features and their corresponding POS labels, syntactic constituents, and dependency relations for x5=“deficiency” are highlighted in part (a), (b) and (c) respectively.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [vancouver.bst](#)
- [spbasic.bst](#)
- [ReferencePDF.pdf](#)
- [bmcartbiblio.sty](#)
- [bmcart.cls](#)
- [bmcmathphys.bst](#)