Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

Cross-type Biomedical Named Entity Recognition with Deep Multi-Task Learning

Xuan Wang [1,*], Yu Zhang [1], Xiang Ren [2,*], Yuhao Zhang [3], Marinka Zitnik [4], Jingbo Shang [1], Curtis Langlotz [3] and Jiawei Han [1]

[1]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA,
[2]Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA,
[3]School of Medicine, Stanford University, Stanford, CA 94305, USA, and
[4]Department of Computer Science, Stanford University, Stanford, CA 94305, USA

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: State-of-the-art biomedical named entity recognition systems often require specific handcrafted features for each entity type. The feature generation process is time and labor consuming, and leads to highly specialized systems not adaptable to new entity types. Although recent studies explored using neural network models for BioNER to free experts from manual feature generation, the performance remains limited by the available training data for each entity type.

Results: We propose a multi-task learning framework for BioNER to collectively use the training data of different entity types and improve the performance on each of them. In experiments on five BioNER datasets covering four major biomedical entity types, our multi-task model outperforms state-of-the-art systems and other neural network models by a large margin. Further analysis shows that the large performance gains come from sharing character- and word-level information across different biomedical entities.

Availability: The source code for our models is available at https://github.com/yuzhimanhua/lm-lstm-crf, and the corpora are available at https://github.com/cambridgeltl/MTL-Bioinformatics-2016.

Contact: xwang174@illinois.edu, xiangren@usc.edu.

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Biomedical named entity recognition (BioNER) is the most fundamental task in biomedical text mining, which automatically recognizes and extracts biomedical entities (e.g., genes, proteins, chemicals and diseases) from text. BioNER can be used to identify new gene names from text (Smith *et al.*, 2008). It also serves as a primitive step of many downstream applications, such as relation extraction (Cokol *et al.*, 2005) and knowledge base completion (Szklarczyk *et al.*, 2017, Wei *et al.*, 2013, Xie *et al.*, 2013).

BioNER is typically formulated as a sequence labeling problem whose goal is to assign a label to each word in a sentence. State-of-the-art BioNER systems often require handcrafted features (e.g., capitalization, prefix and suffix) to be specifically designed for each entity type (Ando, 2007, Leaman and Lu, 2016, Zhou and Su, 2004, Lu *et al.*, 2015). This feature generation process often takes the majority of time and cost in the BioNER system development (Leser and Hakenberg, 2005). Furthermore, it leads to highly specialized systems that cannot be directly used to recognize new entity types. Moreover, the accuracy of BioNER tools is still a limiting factor of the performance of current biomedical text mining pipelines (Huang and Lu, 2015).

Recent NER studies considered neural network models to automatically generate quality features (Huang *et al.*, 2015, Chiu and Nichols, 2016, Ma and Hovy, 2016, Lample *et al.*, 2016, Liu *et al.*, 2018). For BioNER, Crichton *et al.* took each word token and its surrounding context words as input into a convolutional neural network (CNN). Habibi *et al.* adopted the model from Lample *et al.* and used word embeddings as input into a bidirectional long short-term memory-conditional random field (BiLSTM-CRF) model. These neural network models free experts

from manual feature generation. However, the models can have millions of parameters and require very large datasets to reliably estimate the parameters. In many biomedical settings, datasets at this scale are not readily available and thus neural network models cannot realize their potential performance to the fullest. Although neural network models can outperform powerful shallow machine learning models (e.g., CRF models (Lafferty *et al.*, 2001)), they still cannot always outperform state-of-the-art BioNER systems utilizing handcrafted features.

One way to improve is to collectively use the training data of different entity types and improve performance on each of them. A native combination of all the datasets could introduce a lot of false negative labels since each dataset is specifically annotated for one or a few entity types. Multi-task learning (MTL) (Collobert and Weston, 2008, Søgaard and Goldberg, 2016) is an approach to collectively train a model on several related tasks at the same time, so that each task can benefit from annotations available in other tasks without introducing training errors. MTL has been successfully applied in several domains, including natural language processing (Collobert and Weston, 2008), speech recognition (Deng *et al.*, 2013), computer vision (Girshick, 2015) and drug discovery (Ramsundar *et al.*, 2015). However, MTL has seen limited success in BioNER so far. For example, Crichton *et al.* incorporated MTL with CNN for BioNER. However, their CNN model is not as efficient as recent BiLSTM models. Additionally, Crichton *et al.* only considers word-level features as input without capturing any character-level lexical information. As a result, their best performing multi-task CNN model still cannot outperform state-of-the-art systems that utilize handcrafted features.

In this paper, we propose a new multi-task learning framework for BioNER based on neural network models. The proposed framework frees biomedical experts from manual feature generation and also achieves excellent performance. Our multi-task model is built upon a single-task neural network model (Liu *et al.*, 2018). In particular, we consider a BiLSTM-CRF model with an additional context-dependent BiLSTM layer for character embedding. A prominent property of our model is that inputs from different datasets can share both character- and word-level information. The sharing of information is achieved by re-using the same parameters in the BiLSTM units and has important implications for model performance. Specifically, in the multi-task setting, input sentences from different datasets go through the same neural network model. The model outputs the original sentences, each accompanied with a sequence of labels representing biomedical entity types identified in the sentence (Figure 1). We compare the proposed multi-task model with state-of-the-art BioNER systems and neural network NER tools on five benchmark datasets covering four major entity types. Results show that the proposed multi-task neural approach achieves substantially better performance than other models. Interestingly, the proposed approach outperforms several neural models that do not consider multi-task learning, suggesting that multi-task learning plays an important role in a successful BioNER system. Altogether, this work introduces a new text-mining approach that can help scientists exploit knowledge buried in biomedical literature in a systematic and unbiased way.

2 Background

In this section, we introduce basic neural network architectures that are relevant for our multi-task learning approach.

2.1 Long Short-Term Memory (LSTM)

Long short-term memory neural network is a specific type of recurrent neural network that models dependencies between elements in a sequence through recurrent connections. The input to an LSTM network is a sequence of vectors $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T\}$, where vector $\boldsymbol{x}_i$ is a representation vector of a word in the input sentence. The output is a sequence of vectors $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_T\}$, where $\boldsymbol{h}_i$ is a hidden state vector. At step t of the recurrent calculation, the network takes $\boldsymbol{x}_t, \boldsymbol{c}_{t-1}, \boldsymbol{h}_{t-1}$ as inputs and produces $\boldsymbol{c}_t, \boldsymbol{h}_t$ via the following intermediate calculations:

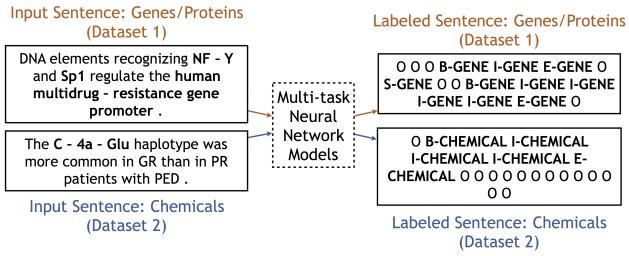


Fig. 1. The illustrative figure of neural network based multi-task framework. The input are sentences from different biomedical datasets. Each sentence will go through the same multi-task neural network models in the center and update the same set of parameters. Then the model will output the entity type labels for each word in the input sentences, such as genes and chemicals, using the BIOES schema.

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}^i \boldsymbol{x}_t + \boldsymbol{U}^i \boldsymbol{h}_{t-1} + \boldsymbol{b}^i) \tag{1}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}^f \boldsymbol{x}_t + \boldsymbol{U}^f \boldsymbol{h}_{t-1} + \boldsymbol{b}^f) \tag{2}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}^o \boldsymbol{x}_t + \boldsymbol{U}^o \boldsymbol{h}_{t-1} + \boldsymbol{b}^o) \tag{3}$$

$$\boldsymbol{g}_t = tanh(\boldsymbol{W}^g \boldsymbol{x}_t + \boldsymbol{U}^g \boldsymbol{h}_{t-1} + \boldsymbol{b}^g) \tag{4}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \boldsymbol{g}_t \tag{5}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot tanh(\boldsymbol{c}_t), \tag{6}$$

where $\sigma(\cdot)$ and $tanh(\cdot)$ denote element-wise sigmoid and hyperbolic tangent functions, respectively, and $\odot$ denotes element-wise multiplication. The $\boldsymbol{i}_t$, $\boldsymbol{f}_t$ and $\boldsymbol{o}_t$ are referred to as input, forget, and output gates, respectively. The $\boldsymbol{g}_t$ and $\boldsymbol{c}_t$ are intermediate calculation steps. At $t = 1$, $\boldsymbol{h}_0$ and $\boldsymbol{c}_0$ are initialized to zero vectors. The trainable parameters are $\boldsymbol{W}^j$,$\boldsymbol{U}^j$ and $\boldsymbol{b}^j$ for $j \in \{i, f, o, g\}$.

The LSTM architecture described above can only process the input in one direction. The bi-directional long short-term memory (BiLSTM) model improves the LSTM by feeding the input to the LSTM network twice, once in the original direction and once in the reversed direction. Outputs from both directions are concatenated to represent the final output. This design allows for detection of dependencies from both previous and subsequent words in a sequence.

2.2 Bi-directional Long Short-Term Memory-Conditional Random Field (BiLSTM-CRF)

A naive way of applying the BiLSTM network to sequence labeling is to use the output hidden state vectors to make independent tagging decisions. However, in many sequence labeling tasks such as BioNER, it is useful to also model the dependencies across output tags. The BiLSTM-CRF network adds a conditional random field (CRF) layer on top of a BiLSTM network. This BiLSTM-CRF network takes the input sequence $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$ to predict an output label sequence $\boldsymbol{y} = \{y_1, y_2, ..., y_n\}$. A score is defined as:

$$s(\boldsymbol{X}, \boldsymbol{y}) = \sum_{i=0}^{n} \boldsymbol{A}_{y_i, y_{i+1}} + \sum_{i=1}^{n} \boldsymbol{P}_{i, y_i}, \tag{7}$$

where $\boldsymbol{P}$ is an $n \times k$ matrix of the output from the BiLSTM layer, n is the sequence length, k is the number of distinct labels, $\boldsymbol{A}$ is a $(k+2) \times (k+2)$

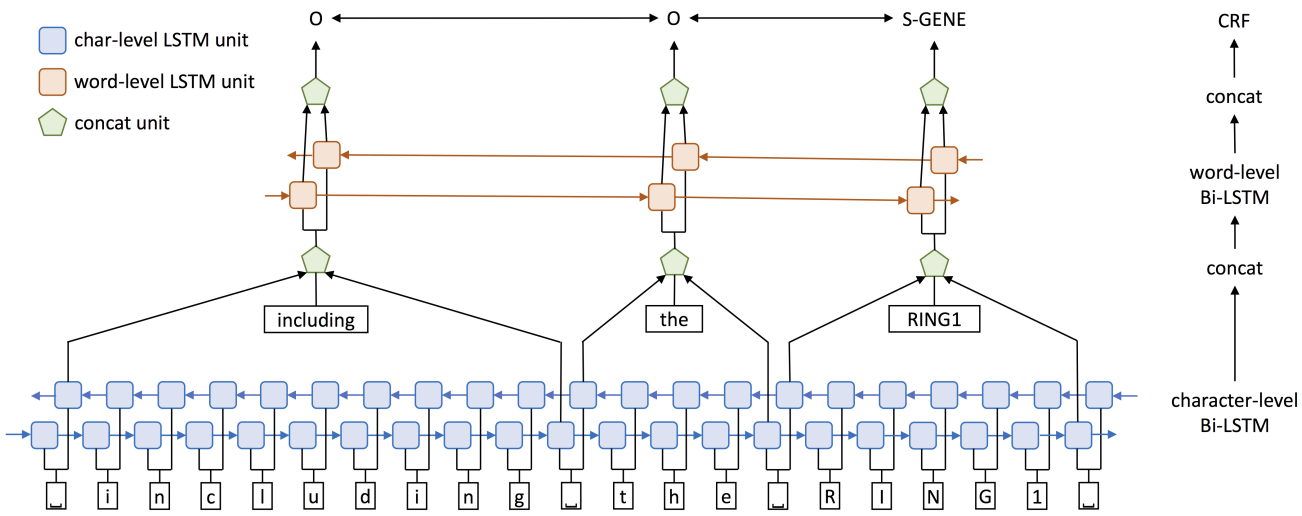


Fig. 2. Single-task learning neural network architecture. The input is a sentence from the biomedical literature. The white rectangles represent character and word embeddings. The blue rectangles represent the first character-level BiLSTM. The red rectangles represent the second word-level BiLSTM. The green pentagons represent the concatenation units. The tags on the top, e.g., 'O', 'S-GENE', are the output of the final CRF layer, which are the entity labels we get for each word in the sentence.

transition matrix and $\boldsymbol{A}_{i,j}$ represents the transition probability from the i-th label to the j-th label. Note that two additional labels <*start*> and <*end*> are used to represent the start and end of a sentence, respectively. We further define $\boldsymbol{Y_X}$ as all possible sequence labels given the input sequence $\boldsymbol{X}$. The training process maximizes the log-probability of the label sequence $\boldsymbol{y}$ given the input sequence $\boldsymbol{X}$:

$$log(p(\boldsymbol{y}|\boldsymbol{X})) = log\frac{e^{s(\boldsymbol{X},\boldsymbol{y})}}{\sum_{\boldsymbol{y'}\in\boldsymbol{Y_X}} e^{s(\boldsymbol{X},\boldsymbol{y'})}}. \quad (8)$$

A three-layer BiLSTM-CRF architecture is employed by Lample *et al.* and Habibi *et al.* to jointly model the word and the character sequences in the input sentence. In this architecture, the first BiLSTM layer takes character embedding sequence of each word as input, and produces a character-level representation vector for this word as output. This character-level vector is then concatenated with a word embedding vector, and fed into a second BiLSTM layer. Lastly, a CRF layer takes the output vectors from the second BiLSTM layer, and outputs the best tag sequence by maximizing the log-probability in Equation 8.

In practice, the character embedding vectors are randomly initialized and co-trained during the model training process. The word embedding vectors are retrieved directly from a pre-trained word embedding lookup table. The classical Viterbi algorithm is used to infer the final labels for the CRF model. The three-layer BiLSTM-CRF model is a differentiable neural network architecture that can be trained by backpropagation.

3 Deep multi-task learning for BioNER

In this section, we first describe a single-task neural model that better handles out-of-vocabulary words by modeling character sequences. We then introduce multi-task models that combine single-task models in three different ways and together represent a flexible deep multi-task learning framework for BioNER.

3.1 Baseline single-task model (STM)

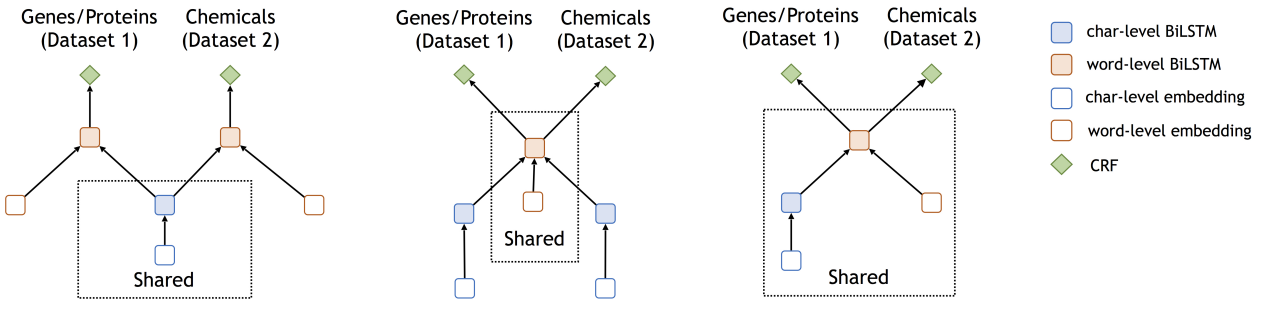
The vanilla BiLSTM-CRF model can learn high-quality representations for words that appeared in the training dataset. However, it often fails to generalize to out-of-vocabulary words, i.e., words that did not appear in the training dataset. These out-of-vocabulary words are especially common in biomedical text. Therefore, for the baseline single-task BioNER model, we use a neural network architecture that better handles out-of-vocabulary words. As shown in Figure 2, our single-task model consists of three layers. In the first layer, a BiLSTM network is used to model the character sequence of the input sentence. We use character embedding vectors as input to the network. Hidden state vectors at the word boundaries of this character-level BiLSTM are then selected and concatenated with word embedding vectors to form word representations. Next, these word representation vectors are fed into a second word-level BiLSTM layer. Lastly, output of this word-level BiLSTM is fed into the a CRF layer for label prediction. Compared to the vanilla BiLSTM-CRF model, a major advantage of this model is that it can infer the meaning of an out-of-vocabulary word from its character sequence and other characters around it. For example, the network is able to infer that "RING2" is likely to represent a gene symbol, even though then network may have only seen the word "RING1" during training.

3.2 Multi-task models (MTMs)

An important characteristic of the BioNER task is the limited availability of supervised training data for each entity type. We propose a multi-task learning approach to address this problem by training different BioNER models on datasets with different entity types while sharing parameters across these models. We hypothesize that the proposed approach can make more efficient use of the data and encourage the models to learn representations that better generalize.

We give a formal definition of the multi-task setting as the following. Given m datasets, for $i \in \{1, ..., m\}$, each dataset D_i consists of n_i training samples, i.e., $D_i = \{\boldsymbol{x}_j^i, y_j^i\}_{j=1}^{n_i}$. We denote the training matrix for each dataset as $\boldsymbol{X}^i = \{\boldsymbol{x}_1^i, ..., \boldsymbol{x}_{n_i}^i\}$ and the labels for each dataset as $\boldsymbol{y}^i = \{y_1^i, ..., y_{n_i}^i\}$. A multi-task model therefore consists of m different models, each trained on a separate dataset, while sharing part of the model parameters across datasets. The loss function L is:

$$L = \sum_{i=1}^{m} \lambda_i L_i = \sum_{i=1}^{m} \lambda_i \log(p(\boldsymbol{y}^i|\boldsymbol{X}^i)). \quad (9)$$



(a) MTM-C (b) MTM-W (c) MTM-CW

Fig. 3. Three multi-task learning neural network models. The blue empty rectangles represent the character embeddings. The blue filled rectangles represent the character-level BiLSTM. The red empty rectangles represent the word-level embeddings. The red filled rectangles represent the word-level BiLSTM. The green pentagons represent the CRF layer. (a) MTM-C: multi-task learning neural network with a shared character layer and a task-specific word layer, (b): MTM-W: multi-task learning neural network with a task-specific character layer and a shared word layer, (c) MTM-CW: multi-task learning neural network with shared character and word layers.

The log-likelihood term is shown in Equation 8 and λ_i is a positive regularization parameter that controls the contribution of each dataset. In the experiments, we set $\lambda_i = 1$ for $i \in \{1, ..., m\}$.

We propose three different multi-task models, as illustrated in Figure 3. These three models differ in which part of the model parameters are shared across multiple datasets:

MTM-C This model shares the character-level parameters among tasks but uses task-specific word-level parameters for each task. All datasets are iteratively used to train the model. When a dataset is used, the parameters updated during the training are the word-level parameters specific to this task and the shared character-level parameters. The detailed architecture of this multi-task model is shown in Figure 3(a).

MTM-W This model uses task-specific character-level parameters for each task but shares the word-level parameters among tasks. When a dataset is used, the parameters updated during the training are the character-level parameters specific to this task and the shared word-level parameters. The detailed architecture of this multi-task model is shown in Figure 3(b).

MTM-CW This model shares both character- and word-level parameters among tasks. During the training, all datasets share and update the same set of parameters at both the character level and the word level. Each dataset has its specific CRF layer for label prediction. MTM-CW is the most comprehensive among the three proposed multi-task models. It enables sharing both character- and word-level information between different biomedical entities, while the other two models only enable sharing part of the information. The detailed architecture of this multi-task model is shown in Figure 3(c).

4 Experimental setup

In this section, we describe the datasets, evaluation metrics, and provide an overview of experimental setup.

4.1 Datasets

We use five BioNER datasets collected by Crichton *et al.* (Table 1). These datasets cover major biomedical entities (e.g., genes, proteins, chemicals, diseases) and come with state-of-the-art performance values that can be used for comparison. Each dataset is divided into three subsets: a training set, a development, and a test set. We use training and development sets to train the final model. All datasets are publicly available and can be downloaded from: https://github.com/cambridgeltl/MTL-Bioinformatics-2016. As part of preprocessing, word labels are encoded using an IOBES scheme. In this scheme, for example, a word describing a gene entity is tagged with "B-Gene" if it is at the beginning of the entity, "I-Gene" if it is in the middle of the entity, and "E-Gene" if it is at the end of the entity. Single-word gene entities are tagged with "S-Gene". All other words not describing entities of interest are tagged as 'O'. Next, we briefly describe each dataset and its corresponding state-of-the-art BioNER system.

Table 1. Biomedical NER datasets used in the experiments.

Dataset	Size	Entity types and counts
BC2GM	20,000 sentences	Gene/Protein (24,583)
BC4CHEMD	10,000 abstracts	Chemical (84,310)
BC5CDR	1,500 articles	Chemical (15,935), Disease (12,852)
NCBI-Disease	793 abstracts	Disease (6,881)
JNLPBA	2,404 abstracts	Gene/Protein (35,336), Cell Line (4,330), DNA (10,589), Cell Type (8,649), RNA (1,069)

BC2GM The state-of-the-art system in the BioCreative II gene mention recognition task is a semi-supervised learning method using alternating structure optimization (Ando, 2007).

BC4CHEMD The state-of-the-art system in the BioCreative IV chemical entity mention recognition task is the *CHEMDNER* system, which is based on mixed conditional random fields with Brown clustering of words (Lu *et al.*, 2015).

BC5CDR The state-of-the-art system in the most recent BioCreative V chemical and disease mention recognition task is the *TaggerOne* system, which uses a semi-Markov model for joint entity recognition and normalization (Leaman and Lu, 2016).

NCBI-Disease The NCBI disease corpus was initially introduced for disease name recognition and normalization. It has been widely used for a lot of applications. The state-of-the-art system on this dataset (Leaman and Lu, 2016) is also the *TaggerOne* system.

JNLPBA The state-of-the-art system (Zhou and Su, 2004) for the 2004 JNLPBA shared task on biomedical entity recognition uses a hidden markov model (HMM). Although this task and the model is a bit old compared with the others, it still remains a competitive benchmark method for comparison.

Table 2. Performance of baseline neural network models and the proposed MTM-CW model. Bold: best scores, *: significantly worse than the MTM-CW model ($p \leq 0.05$), **: significantly worse than the MTM-CW model ($p \leq 0.01$). The details of dataset benchmark systems and evaluation methods are described in Section 4.1 and 4.2, respectively.

		Dataset Benchmark	Crichton *et al.*	Lample *et al.* Habibi *et al.*	Ma and Hovy	STM	MTM-CW
BC2GM (Exact)	Precision	-	-	81.57*	79.09**	81.11*	**82.10**
	Recall	-	-	**79.48**	77.87**	78.91**	79.42
	F1	-	73.17**	80.51	78.48**	80.00*	**80.74**
BC2GM (Alternative)	Precision	88.48	-	87.27**	83.50**	88.21*	**89.45**
	Recall	85.97**	-	87.84	87.13*	87.43*	**88.67**
	F1	87.21**	84.41**	87.55*	85.27**	87.82*	**89.06**
BC4CHEMD	Precision	88.73**	-	89.68*	90.83	90.53*	**91.30**
	Recall	87.41	-	85.87*	83.19**	87.04	**87.53**
	F1	88.06*	83.02**	87.74**	86.84**	88.75	**89.37**
BC5CDR	Precision	**89.21**	-	87.60**	89.16	88.84	89.10
	Recall	84.45**	-	86.25**	84.28**	85.16**	**88.47**
	F1	86.76**	83.90**	86.92**	86.65**	86.96**	**88.78**
NCBI-Disease	Precision	85.10	-	86.11	**86.89**	84.95	85.86
	Recall	80.80**	-	85.49	78.75**	82.92*	**86.42**
	F1	82.90**	80.37**	85.80	82.62**	83.92*	**86.14**
JNLPBA	Precision	69.42**	-	**71.35**	70.28*	69.60**	70.91
	Recall	75.99	-	75.74	75.26	74.95*	**76.34**
	F1	72.55**	70.09**	73.48	72.68*	72.17**	**73.52**

4.2 Evaluation metrics

We report the performance of all the models on the test set. We deem each predicted entity as correct only if it is an *exact match* to an entity at the same position in the ground truth annotation. Then we calculate the precision, recall and F1 scores on all datasets and macro-averaged F1 scores on all entity types. For error analysis, we compare the ratios of false positive (FP) and false negative (FN) labels in the single-task and the multi-task models in Supplementary Material: False error analysis.

The test set of the BC2GM dataset is constructed slightly differently compared to the test sets of other datasets. BC2GM additionally provides a list of alternative answers for each entity in the test set. A predicted entity is deemed correct as long as it matches the ground truth or one of the alternative answers. We refer to this measurement as *alternative match* and report scores under both *exact match* and *alternative match* for the BC2GM dataset.

4.3 Pre-trained word embeddings

We initialize the word embedding matrix with pre-trained word vectors from Pyysalo *et al.*, 2013 in all experiments. These word vectors are trained using the skip-gram model, as described in Mikolov *et al.*, 2013, for learning distributed representations of words using contextual information. Three sets of word vectors are provided with different training data. The first one is trained on the whole PubMed abstracts, the second one is trained on the PubMed abstracts together with all the full-text articles from PubMed Central (PMC), and the thrid one is different from the second one by also training the vectors on the Wikipedia corpora. We found the third set of word vectors to be the most effective and therefore used it for the model development. We also provide a full comparison of the impact of using the three sets of word embeddings in Supplementary Material: Performance of Word Embeddings. In all five datasets, rare words (i.e., words with frequency less than 5) are replaced by a special <*UNK*> token, whose embedding is randomly initialized and fine-tuned during model training.

4.4 Training details

To train the proposed single-task and multi-task models, we use a learning rate of 0.01 with a decay rate of 0.05 after every epoch of training. The dimension of word and character embedding vectors are set to be 200 and 30, respectively. We adopt a hidden state size of 200 for both character- and word-level BiLSTM layers. Note that in the original paper of Liu *et al.*, they also incorporate advanced strategies such as highway structures to further improve the NER performance. We tested these variations but did not observe any significant boost in the performance. Therefore, we do not adopt these strategies in this work.

To compare the performance with other neural network models on each dataset, we trained their model on the five datasets with the default parameter settings as used in their paper. We directly reported the performance from all the benchmark systems and Crichton *et al.*, 2017.

5 Results

5.1 Performance comparison on benchmark datasets

We compare the proposed single-task (Section 3.1) and multi-task models (Section 3.2) with state-of-the-art BioNER systems and three neural network models from Crichton *et al.*, Lample *et al.*, Habibi *et al.*, and Ma and Hovy. The results (precision, recall and F1) are shown in Table 2. We measure statistical significance through a two-tailed t-test in all reported experiments by repeating each experiment three times.

We observe that the MTM-CW model achieves significantly higher F1 scores than state-of-the-art benchmark systems (column Dataset Benchmark in Table 2) on all of the five datasets. Following established practice in the literature, we use exact match to compare benchmark performance on all the datasets except for the BC2GM, where we report benchmark performance based on alternative match. Furthermore, MTM-CW generally achieves significantly higher F1 scores than other neural network models. These results show that the proposed multi-task learning neural network significantly outperforms state-of-the-art systems and other neural networks. In particular, the MTM-CW model consistently achieves

Table 3. F1 scores of three multi-task models proposed in this paper. Bold: best scores, *: significantly worse than the MTM-CW model ($p \leq 0.05$), **: significantly worse than the MTM-CW model ($p \leq 0.01$).

Dataset	MTM-C	MTM-W	MTM-CW
BC2GM	77.80**	79.42**	**80.74**
BC4CHEMD	88.16*	88.49*	**89.37**
BC5CDR	86.05**	88.26*	**88.78**
NCBI-Disease	82.94**	84.81	**86.14**
JNLPBA	71.79**	73.21	**73.52**

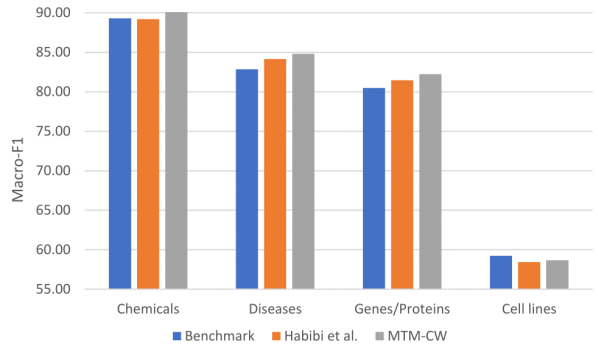


Fig. 4. Macro-averaged F1 scores of the proposed multi-task model compared with benchmark on different entities. Benchmark refers to the performance of state-of-the-art BioNER systems.

a better performance than the single task model, demonstrating that multi-task learning is able to successfully leverage information across BioNER tasks and mutually enhance performance on each single task. We further investigate the performance of three multi-task models (MTM-C, MTM-W, and MTM-CW, Table 3). Results show that the best performing multi-task model is MTM-CW, indicating the importance of lexical features in character-level BiLSTM as well as semantic features in word-level BiLSTM.

5.2 Performance on major biomedical entity types

We also compare all models on four major biomedical entity types: genes/proteins, chemicals, diseases and cell lines since they are the most often annotated entity types. Each entity type comes from multiple datasets: genes/proteins from BC2GM and JNLPBA, chemicals from BC4CHEMD and BC5CDR, diseases from BC5CDR and NCBI-Disease, and cell lines from JNLPBA. The results of macro-F1 scores are shown in Figure 4.

The MTM-CW model performs consistently better than the neural network model (Habibi *et al.*, 2017) on all four entity types. It also outperforms the state-of-the-art systems (Benchmark in Figure 4) on three entity types except for cell lines. These results further confirm that the multi-task neural network model achieves a significantly better performance compared with state-of-art systems and other neural network models for BioNER.

5.3 Integration of biomedical entity dictionaries

A biomedical entity dictionary is a manually-curated list of entity names that belong to a specific entity type. Traditional BioNER systems often make heavy use of these dictionaries in addition to other features. To study whether our system can benefit from the use of additional dictionaries, we retrieve three biomedical entity dictionaries, i.e., genes/proteins, chemicals and diseases, from the comparative toxicogenomics database (CTD) (Davis *et al.*, 2017). We incorporate the dictionary information into the neural network models in two ways: (1) dictionary post-processing to match the 'O'-labeled entities with the dictionary to reduce the false negative rate, (2) dictionary pre-processing to add extra dimensions as part of the input into the word-level BiLSTM. The added dimensions represent whether a word sequence consisting of the word and its consecutive neighbors is in the dictionary. The length of the word sequence is limited to six words, thus 21 dimensions are added for each entity type. We compare the performance of MTM-CW with and without added dictionaries. The results are shown in Table 4.

Table 4. F1 scores of the proposed multi-task model using the CTD entity dictionary. Bold: best scores, *: significantly worse than the MTM-CW model ($p \leq 0.05$), **: significantly worse than the MTM-CW model ($p \leq 0.01$).

Dataset	MTM-CW	+Dictionary Pre-process	+Dictionary Post-process
BC2GM	**80.74**	80.70	61.56**
BC4CHEMD	**89.37**	88.92	83.83**
BC5CDR	88.78	**88.82**	87.90**
NCBI-Disease	**86.14**	85.48	83.80*
JNLPBA	**73.52**	73.35	63.62**

No significant improvement in performance is observed when biomedical entity dictionaries are included into the MTM-CW model at the pre-processing stage. Moreover, including dictionaries at the post-processing stage even reduces the performance due to an increased false positive rate. This is because some 'O'-labeled entities may share the surface name with dictionary entities while not having the same meaning or entity type. These results indicate that our multi-task model can learn an excellent representation using only labeled training data and generalize it to previously unseen test data. As a result, integrating additional signal into the training process in the form of entity dictionaries does not lead to an obvious performance gain.

5.4 Case study

To investigate the major advantages of the multi-task models compared with the single task models, we examine some sentences with predicted labels shown in Table 5. The true labels and the predicted labels of each model are underlined in a sentence.

One major challenge of BioNER is to recognize a long entity with integrity. In Case 1, the true gene entity is "endo-beta-1,4-glucanase-encoding genes". The single-task model tends to break this whole entity into two parts separated by a comma, while the multi-task model can detect this gene entity as a whole. This result could due to the co-training of multiple datasets containing long entity training examples. Another challenge is to detect the correct boundaries of biomedical entities. In Case 2, the correct protein entity is "SMase" in the phrase "SMase - sphingomyelin complex structure". The single-task models recognize the whole phrase as a protein entity. Our multi-task model is able to detect the correct right boundary of the protein entity, probably also due to seeing more examples from other datasets which may contain "sphingomyelin" as a non-chemical entity. In Case 3, the adjective words "human" and "complement factor" in front of "H deficiency" should be included as part of the true entity. The single-task models missed the adjective words while the multi-task model is able to detect the correct right boundary of the disease entity. In summary, the multi-task model works better at dealing with two critical challenges for BioNER: (1) recognizing long entities with integrity and (2) detecting correct left and right boundaries of biomedical entities. Both improvements come from collectively training multiple datasets with different entity types and sharing useful information between datasets.

Table 5. Case study of the prediction results. It shows the advantage of the multi-task neural network model compared with the baseline model and single-task model. The true labels and the predicted labels of each model are underlined in the sentence.

		Genes/Proteins
Case 1	True label	This fragment contains two complete endo - beta - 1, 4 - glucanase - encoding genes, designated celCCC and celCCG.
	Habibi	This fragment contains two complete endo - beta - 1, 4 - glucanase - encoding genes, designated celCCC and celCCG.
	STM	This fragment contains two complete endo - beta - 1, 4 - glucanase - encoding genes, designated celCCC and celCCG.
	MTM-CW	This fragment contains two complete endo - beta - 1, 4 - glucanase - encoding genes, designated celCCC and celCCG.
Error		Entity integrity: break a long entity into parts and lose the entity integrity.
Case 2	True label	A model for the SMase - sphingomyelin complex structure was built to investigate how the SMase specifically recognizes its substrate.
	Habibi	A model for the SMase - sphingomyelin complex structure was built to investigate how the SMase specifically recognizes its substrate.
	STM	A model for the SMase - sphingomyelin complex structure was built to investigate how the SMase specifically recognizes its substrate.
	MTM-CW	A model for the SMase - sphingomyelin complex structure was built to investigate how the SMase specifically recognizes its substrate.
Error		Right boundary error: false detection of non-entity tokens as part of the true entity.
		Diseases
Case 3	True label	... human complement factor H deficiency associated with hemolytic uremic syndrome.
	Habibi	...human complement factor H deficiency associated with hemolytic uremic syndrome.
	STM	... human complement factor H deficiency associated with hemolytic uremic syndrome.
	MTM-CW	... human complement factor H deficiency associated with hemolytic uremic syndrome.
Error		Left boundary error: fail to detect the correct left boundary of the true entity due to some adjective words in front.

6 Conclusion

We propose a neural network based multi-task learning framework for biomedical named entity recognition. The proposed framework frees experts from manual feature generation and outperforms the state-of-the-art systems and other neural network models on five benchmark datasets and four major biomedical entity types. Integrating entity dictionaries in the training process does not lead to obvious performance gain. Furthermore, the analysis suggests that the performance improvement mainly comes from sharing character- and word-level information between different biomedical entity types.

There are several further directions with the multi-task model for BioNER. Combining single-task and multi-task models is a useful direction. Also, by further resolving the entity boundary and type conflict problem, we could build a unified system for recognizing multiple types of biomedical entities with high performance and efficiency.

References

Ando, R. K. (2007). Biocreative ii gene mention tagging system at ibm watson. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, volume 23, pages 101–103.

Chiu, J. P. and Nichols, E. (2016). Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, **4**, 357–370.

Cokol, M., Iossifov, I., Weinreb, C., and Rzhetsky, A. (2005). Emergent behavior of growing knowledge about molecular interactions. *Nature biotechnology*, **23**(10), 1243–1247.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 160–167.

Crichton, G., Pyysalo, S., Chiu, B., and Korhonen, A. (2017). A neural network multi-task learning approach to biomedical named entity recognition. *BMC Bioinformatics*, **18**(1), 368.

Davis, A. P., Grondin, C. J., Johnson, R. J., Sciaky, D., King, B. L., McMorran, R., Wiegers, J., Wiegers, T. C., and Mattingly, C. J. (2017). The comparative toxicogenomics database: update 2017. *Nucleic acids research*, **45**(D1), D972–D978.

Deng, L., Hinton, G., and Kingsbury, B. (2013). New types of deep neural network learning for speech recognition and related applications: An overview. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8599–8603.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448.

Habibi, M., Weber, L., Neves, M., Wiegandt, D. L., and Leser, U. (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, **33**(14), i37–i48.

Huang, C.-C. and Lu, Z. (2015). Community challenges in biomedical text mining over 10 years: success, failure and the future. *Briefings in bioinformatics*, **17**(1), 132–144.

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pages 282–289.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 260–270.

Leaman, R. and Lu, Z. (2016). Taggerone: joint named entity recognition and normalization with semi-markov models. *Bioinformatics*, **32**(18), 2839–2846.

Leser, U. and Hakenberg, J. (2005). What makes a gene name? named entity recognition in the biomedical literature. *Briefings in bioinformatics*, **6**(4), 357–369.

Liu, L., Shang, J., Xu, F., Ren, X., Gui, H., Peng, J., and Han, J. (2018). Empower Sequence Labeling with Task-Aware Neural Language Model. In *AAAI*.

Lu, Y., Ji, D., Yao, X., Wei, X., and Liang, X. (2015). Chemdner system with mixed conditional random fields and multi-scale word clustering. *Journal of cheminformatics*, **7**(S1), S4.

Ma, X. and Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1064–1074.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119.

Pyysalo, S., Ginter, F., Moen, H., Salakoski, T., and Ananiadou, S. (2013). Distributional semantics resources for biomedical text processing. In *Proceedings of Languages in Biology and Medicine*.

Ramsundar, B., Kearnes, S., Riley, P., Webster, D., Konerding, D., and Pande, V. (2015). Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072*.

Smith, L., Tanabe, L. K., nee Ando, R. J., Kuo, C.-J., Chung, I.-F., Hsu, C.-N., Lin, Y.-S., Klinger, R., Friedrich, C. M., Ganchev, K., *et al.* (2008). Overview of biocreative ii gene mention recognition. *Genome Biology*, **9**(S2), S2.

Søgaard, A. and Goldberg, Y. (2016). Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 231–235.

Szklarczyk, D., Morris, J. H., Cook, H., Kuhn, M., Wyder, S., Simonovic, M., Santos, A., Doncheva, N. T., Roth, A., Bork, P., *et al.* (2017). The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic acids research*, **45**(D1), D362–D368.

Wei, C.-H., Kao, H.-Y., and Lu, Z. (2013). Pubtator: a web-based text mining tool for assisting biocuration. *Nucleic acids research*, **41**(W1), W518–W522.

Xie, B., Ding, Q., Han, H., and Wu, D. (2013). mircancer: a microrna–cancer association database constructed by text mining on literature. *Bioinformatics*, **29**(5), 638–644.

Zhou, G. and Su, J. (2004). Exploring deep knowledge resources in biomedical name recognition. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 96–99.