applied sciences

MDPI

Article

Information Extraction from Electronic Medical Records Using Multitask Recurrent Neural Network with Contextual Word Embedding

Jianliang Yang [1], Yuenan Liu [1], Minghui Qian [1,*], Chenghua Guan [2] and Xiangfei Yuan [2]

[1] School of Information Resource Management, Renmin University of China, 59 Zhongguancun Avenue, Beijing 100872, China

[2] School of Economics and Resource Management, Beijing Normal University, 19 Xinjiekou Outer Street, Beijing 100875, China

* Correspondence: qmh@ruc.edu.cn; Tel.: +86-139-1031-3638

Received: 13 August 2019; Accepted: 26 August 2019; Published: 4 September 2019

check for updates

Abstract: Clinical named entity recognition is an essential task for humans to analyze large-scale electronic medical records efficiently. Traditional rule-based solutions need considerable human effort to build rules and dictionaries; machine learning-based solutions need laborious feature engineering. For the moment, deep learning solutions like Long Short-term Memory with Conditional Random Field (LSTM–CRF) achieved considerable performance in many datasets. In this paper, we developed a multitask attention-based bidirectional LSTM–CRF (Att-biLSTM–CRF) model with pretrained Embeddings from Language Models (ELMo) in order to achieve better performance. In the multitask system, an additional task named entity discovery was designed to enhance the model's perception of unknown entities. Experiments were conducted on the 2010 Informatics for Integrating Biology & the Bedside/Veterans Affairs (I2B2/VA) dataset. Experimental results show that our model outperforms the state-of-the-art solution both on the single model and ensemble model. Our work proposes an approach to improve the recall in the clinical named entity recognition task based on the multitask mechanism.

Keywords: clinical named entity recognition; information extraction; multitask model; long short-term memory; conditional random field

1. Introduction

Along with the popularization of medical information systems, more and more electronic medical records (EMR) are produced. As most of the content in EMRs involves unstructured texts, interpretation from specialists is needed to acquire relevant information in EMRs. However, in the face of large-scale EMRs, automated solutions are indispensable. Clinical named entity recognition (CNER) is a particular case in natural language processing (NLP) information extraction tasks, and it aims to extract specific conceptions from unstructured texts, such as problems, medical tests, and treatments [1], which is an essential process for transforming unstructured EMR texts into structured medical data. A highly effective CNER solution will help improve the efficiency of analyzing large-scale EMRs, thus supporting extensive medical research and the development of medical information systems.

Traditional clinical named entity solutions are rule-based, for example, Medical Language Extraction and Encoding System (MedLEE) [2], MetaMap [3], clinical Text Analysis and Knowledge Extraction System (cTAKES) [4], and KnowledgeMap [5]. Rule-based systems need considerable human effort to build basic rules and sometimes a specialized sub-field dictionary, which is specific to the existing entities, with a weak ability to recognize new entities and misspellings [6]. Rule-based systems have a high precision score; however, due to the limited rules, they have a low recall in general [7]. Given the disadvantages of rule-based systems, systems based on machine learning were proposed for implementing clinical information extraction to reduce the reliance on human-built rules and dictionaries. Furthermore, an increasing number of public medical sequence labeling datasets such as the Center for Informatics for Integrating Biology and the Bedside (I2B2) 2010 [8] and Semantic Evaluation (SemEval) 2014 [9] offer data fundamentals for training machine learning models. Machine learning models like support vector machine (SVM), conditional random field (CRF), and hidden Markov model (HMM) achieved superior results [10–12]. Among these models, linear chain CRF [13] could be one of the most widely used algorithms on account of its strong ability to model the state transition in the token sequence and tag sequence. Compared to rule-based systems, systems based on machine learning let the system learn rules based on the clinical records instead of prior defined rules, which enhances the system's ability to identify unknown entities.

Solutions based on machine learning usually contain two main processes: feature engineering and classification [7]. However, the feature engineering process is a weak link in machine learning systems. Researchers need to manually select possible features and design feature combinations, which is time-consuming, and the features selected could be exclusive to the given task. Being limited by the cognitive differences and deficiencies of humans, manually identified features are incomplete [14]. With the development of deep learning, more researchers focused on deep learning models to implement named entity recognition. Compared to machine learning-based solutions, the advantage of deep learning can free the feature engineering part by changing it to an automated process in the training process. Among deep NLP studies, one branch of recurrent neural networks, the long short-term memory network (LSTM), is a prevalent model for feature extraction due to its ability to keep memories of preceding contents of each token [15]. Recent works used bi-directional LSTM (biLSTM) to extract features and a CRF model to infer sequence labels, called the biLSTM–CRF hybrid model. Compared to machine learning algorithms, the biLSTM–CRF model achieved considerable performance compared to previous machine learning models [16–19]. Furthermore, biLSTM–CRF models use unsupervised pretrained word embeddings as features instead of manually engineered features, which reduces human factors in the system.

Recently, contextual word embeddings such as Embeddings from Language Models (ELMo) [20] and Bidirectional Encoder Representations from Transformers (BERT) [21] brought new improvements to the named entity recognition (NER) task. Embedding algorithms like Word2vec [22] and GloVe [23] are based on the meanings of words (the meaning is speculated by word co-occurrences), and they map each word to a vector using a language model. However, in different contexts, the same word may have different meanings. For instance, the meaning of the word "bank" in "go to the riverbank" and "go to the bank to deposit" is different. Contextual word embedding algorithms solve this problem by giving various embedding vectors of the same word in various contexts [20,21]. In CNER studies, models with pretrained contextual word embeddings in medical corpora outperformed those with Word2vec and GloVe [24].

With the proposal of contextual word embedding, methods like character combined embeddings and attention mechanism are yet to be tested, and prior studies based on deep learning did not pay enough attention to the system's perception of unknown entities, as the recall of those systems is relatively low. Our study hypothesizes that, through combining contextual word embedding, multitask, and attention mechanisms, the system can achieve better performance than previous works and recognize more unknown entities. Thus, we propose a multitask biLSTM–CRF model with pretrained ELMo contextual word embeddings to extract clinical named entities. The multitask mechanism separates the NER task into two parts: named entity discovery and named entity classification, in which the classification task is the primary task, and the discovery task is the secondary task. Usually, the secondary task in a multitask model can be seen as a regularizer [25]; this mechanism was implemented to reduce noise in the social media named entity recognition task [26]. We constructed the multitask mechanism to enhance the model's perception of unknown entities to improve recall.

In addition, we drew a self-attention mechanism into the model. Experiments were done on the I2B2 2010/VA [8] dataset. The results show that our model outperforms the typical LSTM–CRF models with ELMo contextual word embeddings. Our work provides an approach to improve the system's performance and perception of unknown entities based on multitask mechanism.

The paper is organized as follows: Section 2 summarizes previous studies on methods of clinical named entity recognition and describes the multitask recurrent neural network model and the ELMo pretrained contextual word embedding. Section 3 presents the experimental setting and results. Section 4 discusses the experimental results. At last, Section 5 concludes the findings of this study and describes some possible future directions based on this work.

2. Materials and Methods

In this section, we describe related work on clinical named entity recognition and how our model was designed. Section 2.1 describes related work on clinical named entity recognition. Section 2.2 describes the algorithm of ELMo contextual word embedding and its pretraining corpus. Section 2.3 describes the structure of the bi-directional LSTM with attention layers. Section 2.4 describes the multitask mechanism, which consists of the sequential inference task and the entity discovery task.

2.1. Related Work on Clinical Named Entity Recognition

The development of clinical named entity recognition systems approximately goes through three stages, which are rule-based systems (also known as knowledge-based or lexicon-based), feature engineered machine learning systems, and deep learning systems. Rule-based systems mainly rely on search patterns in the form of characters and symbols which contain the content information of some specific entity. Once the search patterns are built, the rule-based system searches records based on these pre-defined patterns. Prior works commonly built regular expressions to express the recognizing rules for named entities, and those regular expressions contained names or part of the names of target entities. Savova et al. extracted peripheral arterial disease (PAD) if the phrase in medical notes matched the pre-defined regular expressions. [27]; Bedmar et al. used similar methods to extract drug entities [28]. The rule-based system works like a retrieval system, and it compares every phrase to its regular expressions to check if the phrase is a named entity. However, the system can recognize an entity only if it fits some regular expressions; in other words, if the system has more regular expressions, it would recognize more entities. A well-performed rule-based system needs abundant lexicon resources to pre-define search patterns [6,29]. Knowledge bases like Unified Medical Language System (UMLS) [30] and DrugBank [31] are commonly used for the pre-definition work. Furthermore, a rule-based system can accurately identify a named entity that appears in its lexicon but becomes helpless for named entities not in its lexicon. Hence, the rule-based system usually has high precision but low recall [7]. Also, building and maintaining a domain-specific lexicon with regular expressions needs many resources. In the face of those shortcomings, machine learning-based systems were put forward.

Machine learning-based systems allow the system itself to learn rules and patterns from clinical records, which decreases the manual work in constructing them. SVM, logistic regression (LR), and CRF are the most commonly implemented algorithms in these systems [29]. These systems achieved relatively excellent performance [32–35]. For implementing algorithms like SVM, LR, and CRF, pivotal content and structural information should be provided and converted into particular forms to allow the learning model to understand sequences and learn patterns. Therefore, feature engineering becomes essential in machine learning-based systems [14]. For example, Roberts et al. implemented SVM to recognize anatomic locations from medical reports, and nine features including lemmas of words, grammatical dependency structure information, and path along the syntactic parse tree were engineered [32]. Sarker et al. implemented three classification approaches including SVM, naïve Bayes, and maximum entropy to extract adverse drug reaction entities; n-grams features, UMLS semantic types, sentiment scores, and topic-based features were engineered [33]. Rochefort et al.

identified geriatric competency exposures from students' clinical notes with LR; features including the number of notes, bag of words features, concept code features, Term Frequency–Inverse Document Frequency (TF-IDF) features, and semantic type features were engineered [34]. Deleger et al. recognized pediatric appendicitis score (PAS) from clinical records with CRF, and more than 20 features were engineered [35]. Manual feature selection is time-consuming, and, as there is no general standard, the manually identified features are usually incomplete [14]. Moreover, some of the features are also based on the medical knowledge base [33,34], which indicates that feature engineering processes also need abundant knowledge resources. Feature engineered machine learning systems can learn rules and patterns through a training process, which dramatically improves efficiency, and studies showed that these systems achieve considerable performance [32–35]. However, the system's performance highly relies on the features that humans selected, which decreases the robustness of the system.

Along with the development of deep learning in NLP, systems based on deep learning methods were proposed. Compared to systems based on machine learning algorithms, one of the best advantages of deep learning systems is the avoidance of manual work. It benefits from unsupervised pretrained embeddings like Word2vec [22] and GloVe [23]. Actually, the first neural network architecture proposed by Collobert et al. for NER tasks constructed features by orthographic information and lexicons which also contained manual work [25], whereas Collobert et al. improved his model by replacing those manually built features with word embedding, which converts a word into an N-dimension vector through an unsupervised training process [36]. Studies on CNER based on deep learning methods mainly follow two directions. One is to optimize the learning model, and the other is to construct or pre-train better embeddings, which can provide more information for the learning model.

For the studies on learning models, Collobert et al. firstly proposed a model with convolution layers to capture local information in the sequence [36]. Models based on Recurrent Neural Network (RNN) were proposed due to its superior ability in sequence learning. Huang et al. proposed a bi-directional LSTM model for sequence labeling and showed that assembling a CRF layer on top of an LSTM could improve performance [37]. Lample et al. proposed the biLSTM–CRF model for NER [38]. The biLSTM with CRF-based model showed its success in many CNER studies. Chalapathy et al. extracted clinical concepts with a biLSTM–CRF architecture, and achieved 83.88% F1 score, 84.36% precision score, and 83.41% recall score on the 2010 I2B2/VA dataset (the version with 170 training notes), which was better than all prior work [7]. Xu et al. extracted disease named entities with the same architecture, and achieved 80.22% F1, which was also better than prior work [39]. Wu et al. compared CRFs, Structured Support Vector Machines (SSVMs), semi-Markov, Convolutional Neural Network (CNN), and biLSTM–CRF on the 2010 I2B2/VA dataset (the version with 349 training notes) and found that biLSTM–CRF achieved the best performance among all learning models with 85.94% F1 score [14]. Xu et al. combined biLSTM–CRF with a global attention mechanism, and conducted experiments on the 2010 I2B2/VA dataset (the version with 170 training notes). They achieved 85.71% F1 score, 86.27% precision score, and 85.15% recall score, which performed the best compared to prior work [19]. At present, biLSTM–CRF is the most approved learning architecture for CNER tasks.

For studies on embeddings, word embedding was widely used in NER tasks. Chalapathy et al. compared random embedding, Word2vec, and GloVe in biLSTM–CRF, and found that the system with GloVe outperformed others [7]. Habibi et al. showed that the pre-training process of word embedding is crucial for NER systems, and, for domain-specific NER tasks, domain-specific embeddings could improve the system's performance [40]. Liu et al. used pretrained Word2vec embeddings on (Medical Literature Analysis and Retrieval System Online) MEDLINE and Wikipedia corpus and achieved considerable performance compared to other studies [41]. As a word can be seen as a sequence of characters, and characters in a word contain parts of a word's meaning and orthographic information, character-level embedding is quite useful for NER tasks. Normally, character-level embeddings are not pretrained; they are initialized randomly and trained by a sub-CNN or sub-RNN in the whole architecture. Liu et al. combined Word2vec embedding and an LSTM-trained character-level embedding as features of a word, which performed much better than only word embeddings [41].

Zeng et al. combined a Word2vec embedding, and a CNN-trained character-level embedding as features of a word, which performed better for some indicators [18]. Along with the development of contextual word embeddings, studies [20,21] showed that contextual embeddings achieved better performance than previous work [7,14,19]. Just like Word2vec embedding, a domain-specific pretrained contextual embedding model performs better in the domain-specific NER task. Zhu et al. compared general pretrained ELMo and clinical pretrained ELMo, and found that clinical ELMo performed much better than general ELMo. They achieved 88.60% F1 score, 89.34% precision score, and 87.87% recall score on the 2010 I2B2/VA dataset (the version with 170 training notes) [24]. Si et al. compared Word2vec, GloVe, fastText, ELMo, BERT-base, BERT-large, and Bio-BERT, and found that BERT-large achieved the best performance [42].

In general, systems with biLSTM–CRF architecture and contextual word embedding set the new state-of-the-art record in many CNER datasets at present [24,42]. However, methods that combine character-level embedding and attention mechanisms, such as in References [41] and [19], with contextual word embeddings are yet to be tested. Moreover, among the existing studies [7,14,19,40–42], the systems all had a relatively low recall, which indicates that those systems were not sensitive enough to unknown entities. Aguilar et al. proposed a multitask system for NER in social media in order to reduce noise, and their system achieved the highest F1 score and decent precision score compared to other systems [26]. Aguilar et al.'s work indicates that we can try to introduce the multitask mechanism in CNER tasks to make the system more sensitive to emerging clinical concepts.

In our work, we design a multitask attention-based biLSTM–CRF model (Att-biLSTM–CRF) to test the effect of the multitask mechanism in the CNER task. Compared with rule-based systems and machine learning-based systems, our work is based on deep learning, whereby we do not rely on human-designed rules and manually engineered features, which significantly improves our system's robustness and usability. Compared to prior work based on deep learning, we combine the biLSTM–CRF architecture, clinical pretrained context embedding, attention mechanism, and multitask mechanism in order to achieve better performance than prior work. Specifically, we test whether the multitask mechanism can improve the system's recall.

2.2. ELMo Contextual Word Embedding

ELMo is a pretrained contextual word embedding model. It is a bidirectional *LSTM* (biLSTM) language model which can generate context-dependent word embeddings [20]. The prediction process of the biLSTM language model is to maximize the log-likelihood of the probability of token *i* from both directions.

$$\sum_{i=1}^{N} (log\, p(t_i \mid t_1, t_2 \ldots, t_{i-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + log\, p(t_i \mid t_{i+1}, t_{i+2} \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s)), \tag{1}$$

where t_i is token *i*, $(t_1, t_2 \ldots t_{i-1})$ is the forward context of token *i*, $(t_{i+1}, t_{i+2} \ldots, t_N)$ is the backward context of token *i*, Θ_x represents the parameters of the token's representations, $\overrightarrow{\Theta}_{LSTM}$ represents the *LSTM* parameters in the forward direction, $\overleftarrow{\Theta}_{LSTM}$ represents the *LSTM* parameters in the backward direction, and Θ_s represents the parameters of the Softmax layer. Then, ELMo combines the representation from each layer of token *i* as follows:

$$R_i = \left\{ h_{i,j}^{LSTM} \middle| j = 0, 1, \ldots, L \right\}, \tag{2}$$

where R_i is the representation of token i, and $h_{i,j}^{LSTM}$ is the hidden layer which is equal to $[\overrightarrow{h}_{i,j}^{LSTM}; \overleftarrow{h}_{i,j}^{LSTM}]$. ELMo collapses the representations from all layers into one single vector. For a specific task, the ELMo representation of token i is calculated by

$$ELMo_i^{task} = \gamma^{task} \sum_{j=0}^{L} s_j^{task} h_{k,j}^{LSTM}, \tag{3}$$

where γ^{task} is the scalar factor to adjust the scale of vector based on the feature of a specific task, and s_j^{task} is the normalized weight of each layer.

ELMo showed better performance in several NLP tasks compared to other context-independent embedding models like Word2vec and GloVe [20]. For some specific domains, a domain-pretrained ELMo model had better performance than generalized ELMo [43,44]. In the clinical NER task, the LSTM–CRF model with ELMo pretrained on the medical corpus Multiparameter Intelligent Monitoring in Intensive Care III (MIMIC III) [45] significantly outperformed the same model with generalized ELMo [24]. In our work, we use the MIMIC III medical corpus pretrained ELMo to produce word embeddings as input variables. Thus, the main comparison is between our work and previous work with clinical ELMo embeddings.

2.3. The Att-biLSTM Model

A recurrent neural network (RNN) is a type of neural network designed to handle sequential data. For sequential data such as stock price data within a period and every token in one sentence, the data in step t typically have some relationships with the previous step. In a language model, the RNN can "remember" the information before the current step, which makes it suitable for sequence prediction [46]. Particularly, for a sequential data series $x_{task} = \{x_0, x_1, x_2, \ldots, x_t, \ldots, x_n\}$ where x_t is the t step of x_{task}, the model calculates the hidden cell output $\mathrm{h_t}$ by x_t and $\mathrm{h_{t-1}}$ for each step at first; then, computing hidden state outputs of all the steps, the model computes its output o_{task}, and each $\mathrm{o_t}$ is calculated by $\mathrm{h_t}$. The mathematical expression of the forward propagation process is as follows:

$$i_t = \tan\mathrm{h}(Ux_t), \tag{4}$$

$$\mathrm{h_t} = Wh_{t-1} + i_t + bias_h, \tag{5}$$

$$\mathrm{o_t} = Vh_t + bias_o, \tag{6}$$

where U is the weight of the input layer, W is the weight in the hidden cell, and V is the weight of the output layer. U, W, and V are shared for all steps. Commonly, after computing the outputs, another layer is added based on the task. For a classification task, a softmax function is usually used to normalize the probability of each class.

Theoretically, a naïve RNN model can handle the previous information for each step. However, in practice, the problems of vanishing gradient and exploding gradient in backpropagation through time (BPTT) result in it failing to learn enough information from previous steps and handle long-term dependencies [47]. Facing this dilemma, the LSTM model was implemented. The LSTM model combats the vanishing gradient and exploding gradient problem by its gating and cell state mechanism [15]. The mechanism includes a forget gate f, an input gate I, and a cell state C. The forward propagation process in an LSTM cell is as follows:

$$f_t = \sigma\Big(W_{fh}h_{t-1} + W_{fx}x_t + bias_f\Big), \tag{7}$$

$$I_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + bias_i), \tag{8}$$

$$\mathrm{C_t} = f_tC_{t-1} + i_t\tanh(W_{Ch}h_{t-1} + W_{cx}x_t + bias_C), \tag{9}$$

$$z_t = \sigma(W_{zh}h_{t-1} + W_{zx}x_t + bias_o), \tag{10}$$

$$h_t = z_t \tanh(C_t), \tag{11}$$

where f_t is the forget weight, and σ is the sigmoid function which restricts f_t between [0, 1]. In the range, 0 means to completely forget previous information, and 1 means to completely remember the previous information. I_t is the input weight of cell t, and it decides how much information should enter the cell. C_t is the value of the current cell t. It consists of previous information adjusted by the forget gate and current information by the input gate. At last, the output of the LSTM cell h_t is the cell state value normalized by a tanh function and then adjusted by z_t, in which z_t decides how much information should be added to the output.

In our work, the model needs to decide a token's label not only by the previous tokens but also by the tokens behind it; thus, we use a two-layer biLSTM to gather information on each token from both directions (shown in Figure 1). As described above, a single bidirectional LSTM generates an output $\overrightarrow{h}_t$ and, for a biLSTM, it uses two independent single LSTM layers to generate an output $[\overrightarrow{h}_t, \overleftarrow{h}_t]$. $[\overrightarrow{h}_t, \overleftarrow{h}_t]$ is the final representation of token t in our model.

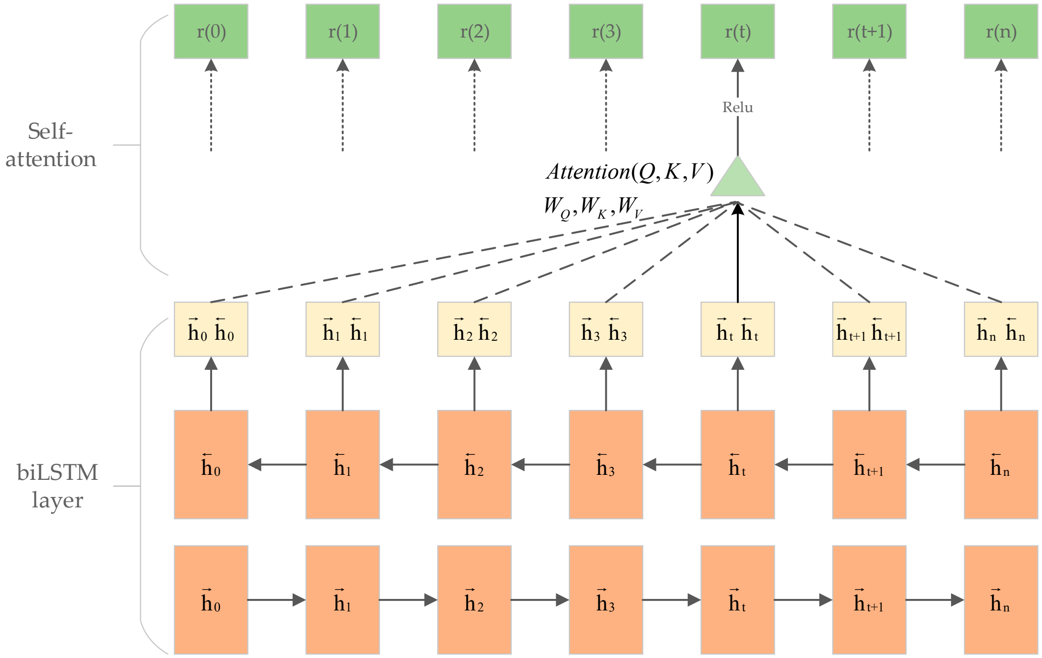
r(0)
r(1)
r(2)
r(3)
r(t)
r(t+1)
r(n)
Self-attention
Relu
$Attention(Q,K,V)$
W_Q,W_K,W_V
$\overrightarrow{h}_0 \overleftarrow{h}_0$
$\overrightarrow{h}_1 \overleftarrow{h}_1$
$\overrightarrow{h}_2 \overleftarrow{h}_2$
$\overrightarrow{h}_3 \overleftarrow{h}_3$
$\overrightarrow{h}_t \overleftarrow{h}_t$
$\overrightarrow{h}_{t+1} \overleftarrow{h}_{t+1}$
$\overrightarrow{h}_n \overleftarrow{h}_n$
biLSTM layer
$\overleftarrow{h}_0$
$\overleftarrow{h}_1$
$\overleftarrow{h}_2$
$\overleftarrow{h}_3$
$\overleftarrow{h}_t$
$\overleftarrow{h}_{t+1}$
$\overleftarrow{h}_n$
$\overrightarrow{h}_0$
$\overrightarrow{h}_1$
$\overrightarrow{h}_2$
$\overrightarrow{h}_3$
$\overrightarrow{h}_t$
$\overrightarrow{h}_{t+1}$
$\overrightarrow{h}_n$

Figure 1. The architecture of attention-based bidirectional Long Short-term Memory (Att-biLSTM). The biLSTM layer captures input data and generates embeddings from two directions. The embedding of each token from two directions is concatenated into one vector as the output of the biLSTM layer. Then, the vectors go through a multi-head self-attention layer. A relu function activates the outputs of the attention layer. Dropout is applied in the biLSTM layer. This part is a part of the encoder in our model.

Considering that, when human beings classify a token into some kind of entity, they may rely on some similar representations around it, we add a multi-head self-attention layer [48] after the biLSTM layer. In this layer, we compute the attention score by a query vector (Q), a key vector (K), and a value vector (V).

$$\text{Attention}(\text{Q}, \text{K}, \text{V}) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)\text{V}, \tag{12}$$

where d_k is the number of dimensions in K to scale the dot product of Q and K. Based on the self-attention mechanism, Q, K, and V are computed from the same input, which is the biLSTM representation in our model. Specifically, $\mathrm{Q} = [\overrightarrow{h}_t, \overleftarrow{h}_t]\mathrm{W}_\mathrm{Q}$, $\mathrm{K} = [\overrightarrow{h}_t, \overleftarrow{h}_t]\mathrm{W}_K$, and $\mathrm{V} = [\overrightarrow{h}_t, \overleftarrow{h}_t]\mathrm{W}_V$. We use two heads to capture information from different perspectives; then, we concatenate the attention matrices and send them into a dense layer to obtain the final representation. The architecture of the Att-biLSTM part in our work is shown in Figure 1.

2.4. Multitask Mechanism

After the model obtains the representations from the Att-biLSTM, it separates the NER task into a primary task and a secondary task. The primary task is a sequential inference task, and the secondary task is an entity discovery task. The two tasks are conducted simultaneously.

In the sequential inference task, the representation vectors are sent to a CRF model to decide which label should be assigned on each token. The reason that we do not use a dense layer and softmax function to estimate the class of each token is that the label of each token has sequential dependence, and the softmax function can capture the dependence information. For instance, it is impossible that a label representing the beginning of an entity follows another beginning label in one entity in the real data, but it may happen in the prediction if we use a dense layer and a softmax function. The CRF model can infer the dependence of token t with token $t-1$ and token t + 1 in a sequence by its state transition algorithm [13]. Thus, a CRF layer is used to infer the sequence in our model. Specifically, giving a sequence $x = \{x_0, \ldots, x_t, \ldots, x_n\}$ and its label sequence $y_{task} = \{y_0, \ldots, y_t, \ldots, y_n\}$, it complies with the following Markov property:

$$\mathrm{P}\Big(\mathrm{y}_t \Big| x, y_0, \ldots y_{t-1}, y_{t+1}, \ldots, y_n\Big) = \mathrm{P}\Big(\mathrm{y}_t \Big| x, y_{t-1}, y_{t+1}\Big). \tag{13}$$

Then, $\mathrm{P}(y_{task}|x)$ is a chain conditional random field, and the conditional probability of y_{task} is

$$\mathrm{P}(y_{task1}|x) = \frac{exp\Big(\sum_{i,k}^{n,K_1} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l}^{n,K_2} \mu_l s_l(y_i, x, i)\Big)}{Z(x)}, \tag{14}$$

where t_k is a transition eigenfunction, s_l is a state feature function, λ_k and μ_l are weight parameters, K_1 is the number of transition features, and K_2 is the number of state features. $Z(x)$ is a normalization function to normalize the probability. t_k and λ_k can be united into one feature function: $f_k(y, x) = \sum\limits_{i=1}^{n} f_k(y_{i-1}, y_i, x, i)$. Then, the probability of given sequence x with label sequences y_{task1} is expressed by the following equation:

$$\mathrm{P}(y_{task1}|x; w) = \frac{exp\Big(\sum_{k=1}^{m} \sum_{i=1}^{n} w_k f_k(y_{i-1}, y_i, x, i)\Big)}{\sum_{y'=0}^{z} exp\Big(\sum_{k=1}^{m} \sum_{i=1}^{n} w_k f_k\Big(y'_{i-1}, y'_i, x, i\Big)\Big)}, \tag{15}$$

where w is the weight matrix of f_k, and z is the set of labels in the label sequence. To maximize $\mathrm{P}(y_{task1}|x; w)$, we optimize w by the log-likelihood estimation algorithm. To obtain the label sequence y_{task1}, we use the Viterbi algorithm to decode the label sequence solved by the CRF layer. Before the representations enter the CRF layer, they go through a dense layer first, and this dense layer has 13 output neural cells which represent all 13 labels.

In the entity discovery task, the representations generated from the Att-biLSTM are sent into a binary classifier; this classifier classifies a token as being an entity or not. A dense layer with two output neural cells is used to represent the two classes, and a softmax function is used to normalize the probability. The following equation expresses the probability of given sequence x with label sequences y_{task2}:

$$\mathrm{P}(y_{task2}|x; w) = \frac{\sum_{i=1}^{n} exp\Big(x^T w_i\Big)}{\sum_{k}^{K} \sum_{i=1}^{n} exp\Big(x^T w_i^k\Big)}. \tag{16}$$

In the backpropagation process, the loss values of two tasks are combined by a linear process, which is

$$loss_{total} = \gamma_1 loss_{t1} + \gamma_2 loss_{t2}, \tag{17}$$

where γ_1 and γ_2 are factors of two loss values, and they represent the priority of the two tasks. In every backpropagation process, the model computes the total loss and conducts gradient descent. The entire architecture of our model is shown in Figure 2. The source code of our model was published online in the Supplementary Materials.

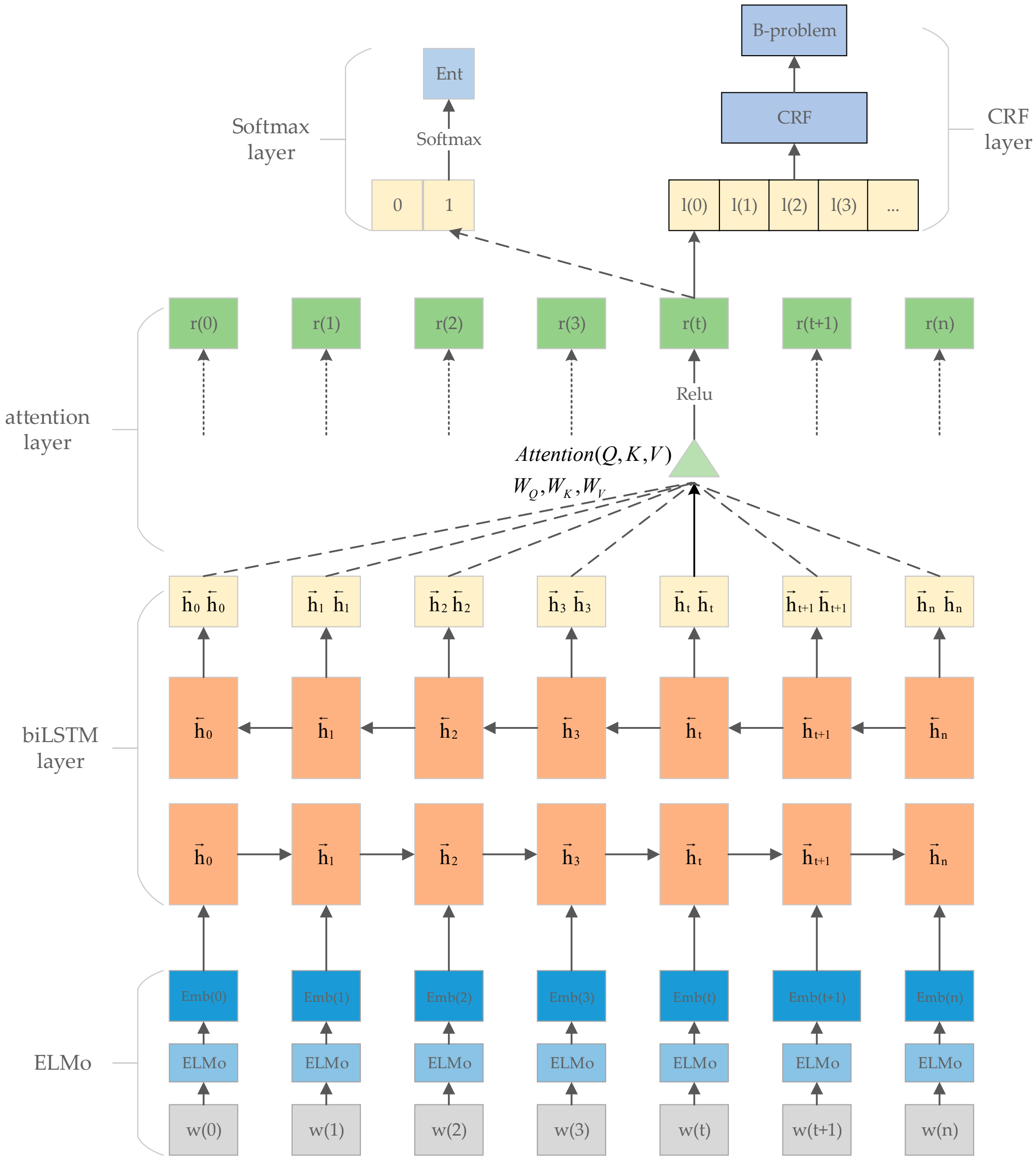

Figure 2. The architecture of our model. Firstly, the tokens in a sequence enter the pretrained Embeddings from Language Models (ELMo) model, and ELMo outputs the contextual embeddings of each token. Then, the Att-biLSTM layer receives the contextual embeddings and outputs the encoded vector of each token. At last, the encoded vectors are sent to the softmax layer to conduct the entity discovery task and to the Conditional Random Field (CRF) layer to decode the sequential labels synchronously.

3. Results

To test the model's performance, in this section, we describe the process of experiments and the experimental results. Section 3.1 describes the dataset we use. The dataset 2010 I2B2/VA is a public dataset for the CNER task, and several studies conducted experiments on this dataset [7,19,24]. Section 3.2 describes the experimental setting such as hyperparameters and the learning optimizer. Section 3.3 describes the evaluation metrics, for which exact precision, recall, and F1 are used. Section 3.4 describes the results of our experiments.

3.1. Dataset

To examine our model, we used the 2010 I2B2/VA dataset from the 2010 I2B2 challenge. This dataset is a set of medical records contributed by Partners Healthcare, Beth Israel Deaconess Medical Center, and the University of Pittsburgh Medical Center [8]. The records in the dataset are in text format, and the dataset is already separated into a training set and testing set. There are three different entities in the dataset: problem, test, and treatment. The descriptions of records, sentences, and tokens are shown in Table 1. In the training set, there are 7073 problem entities, 4844 test entities, and 4606 entities. In the testing set, there are 12,592 problem entities, 9225 test entities, and 9344 treatment entities.

Table 1. Descriptions of training set and testing set in the 2010 Informatics for Integrating Biology & the Bedside/ Veterans Affairs (i2b2/VA) dataset.

Dataset	Records	Sentences	Tokens	Tokens Per Sentence
Training set	170	16,315	149,666	9.17
Testing set	256	27,626	267,758	9.69

3.2. Experimental Setting

In our experiment, we labeled the data with BIEOS format (label prefix B is the token in the beginning of an entity, label prefix I is the token inside an entity, label prefix E is the token at the end of an entity, label prefix O is the token outside any entity, and label prefix S is a single entity). For the training progress, we used the Adam optimizer [49] to train the model and tune the hyperparameters by random search [50]. The early stopping strategy was used to prevent overfitting. The final hyperparameters are shown in Table 2. We implemented our model on the pytorch library on Python 3.7.

Table 2. Hyperparameters chosen in our work.

Hyperparameters	Value
Dimension of Embeddings from Language Models (ELMo)	1024
Bidirectional Long Short-term Memory (biLSTM) hidden size	256
Number of biLSTM layers	2
Number of attention heads	2
Dropout rate	0.5
Learning rate	0.001
Batch size	64
Epochs	100

3.3. Evaluation Metrics

The evaluation metric followed the regulation "Evaluation Methods and Procedures for 2010 I2B2/VA Challenge" [51]. We used the exact F1, exact precision, and exact recall score to evaluate the performance of our model as most works using this dataset did. "Exact" means that the concept entities extracted must match the ground-truth entities exactly both in terms of boundaries and concept type. The definitions of precision, recall, and F1 are shown below.

$$Precision_c = \frac{TP_c}{TP_c + FP_c}, \tag{18}$$

$$Recall_c = \frac{TP_c}{TP_c + FN_c}, \tag{19}$$

$$\mathrm{F1}_c = \frac{2 * (Recall_c * Precision_c)}{Recall_c + Precision_c}, \tag{20}$$

where c is the tag of an entity; TP_c stands for the true positives of entity c, which means that the actual tag of this entity is c, and the predicted tag is also c; FP_c stands for the false positives, which means that the actual tag of this entity is not c, but the predicted tag is *c*; and FN_c stands for false negatives, which means that the actual tag of this entity is c, but the predicted tag is not c. Specifically, as we used the exact metrics, we treated a predicted entity as a true positive only if it matched both the boundary and tag type of the corresponding actual entity. Micro F1 was used to integrate all $\mathrm{F1}_c$.

3.4. Results

We trained the model with different ransom seeds 10 times, and the mean and standard deviations of metrics are reported. Table 3 shows the performance of our model, and the performances from other models which experimented on the same dataset. Our solution in single-model mode achieved an exact F1 score of 87.53 ± 0.11%, exact precision of 87.75 ± 0.18%, and exact recall of 87.32 ± 0.26%. Before our work, the best performing single model was the "ELMo (clinical) + BiLSTM–CRF (single) model" [24], which also used clinical pretrained ELMo word embeddings. We obtained an improvement of 0.69% in mean F1, an improvement of 0.31% in mean precision, and an improvement of 1.06% in mean recall compared with the best performing model.

Table 3. Results of experiments on the 2010 i2b2/VA dataset.

Solutions	F1	Precision	Recall
GloVe-biLSTM–CRF [7]	83.88	84.36	83.41
Clinical Named Entity Recognition system (CliNER) 2.0 [52]	83.8	84.0	83.6
Att-biLSTM–CRF + Transfer [19]	85.71	86.27	85.15
ELMo (General) + BiLSTM–CRF (Single) [24]	82.54 ± 0.14	83.26 ± 0.25	81.84 ± 0.22
Word2vec + multitask-Att-biLSTM–CRF	78.70	79.98	77.47
ELMo (General) + multitask-Att-biLSTM–CRF	83.00	82.91	83.09
ELMo (Clinical) + BiLSTM–CRF (Single) [24]	86.84 ± 0.16	87.44 ± 0.27	86.25 ± 0.26
Our model (Single)	**87.53 ± 0.11**	**87.75± 0.18**	**87.32 ± 0.26**
ELMo (Clinical) + BiLSTM–CRF (Ensemble) [24]	88.60	89.34	87.87
Our model (Ensemble)	**88.78**	89.11	**88.46**

Additionally, we build an ensemble model based on 10 single models trained on different random seeds [53]. The ensemble model classified tokens based on a voter mechanism that chose the most voted label by the 10 single models. The results of our ensemble model are also shown in Table 3. We can see that our ensemble model achieved an F1 score of 88.78, precision of 89.11, and recall of 88.46. Compared to the previous best solution "ELMo (clinical) + BiLSTM–CRF (ensemble)", our model improved by 0.18% in F1 and 0.59% in recall, but had a lower performance for precision (−0.23%). The F1 and precision had a slight variation between our model and the previous best solution, but we can see a noticeable improvement in recall, just like the comparison between the single models. The improvement in recall agrees with the aim of the multi-task mechanism in our model, which was to enhance the model's sensibility to unknown tokens. To see how the multitask mechanism performed if different embedding methods were implemented, we changed the embedding part of our system to general pretrained Word2vec and general pretrained ELMo, and the results are shown in Table 3. The result shows that the system with Word2vec embeddings performs not as good as we expected. The reason may be that the hyperparameters and label format of our system were adjusted for the contextual word embedding. However, the system with general ELMo performed better compared to the result in Reference [24]. The result indicates that the multitask mechanism may have better performance with contextual word embeddings.

The evaluation of the prediction for each type of medical entities is shown in Table 4. We can see that the performance of our system on predicting problem entities was better than the other two kinds of entities in all three indicators, and the prediction of medical test was a little worse than problem and treatment entities. The reason for this difference may be the imbalance of the training dataset, in which there were 7073 problem entities, 4844 test entities, and 4606 entities. However, the prediction results of the three entities were quite close with tiny standard deviations (0.21 F1 score, 0.25 precision, and 0.16 recall), which indicates that our system is stable when predicting the different types of entities.

Table 4. Evaluation of each type of entity.

Entity Type	F1	Precision	Recall
Medical Test	88.37	88.61	88.13
Problem	89.03	89.40	88.66
Treatment	88.96	89.34	88.58
SD	0.21	0.25	0.16

Some works using the 2010 I2B2/VA original dataset are not reported in Table 3, because those works used the original larger dataset of 2010 I2B2/VA which contained 349 records in its training set and 477 records in its testing set [41,42]. For some reason, I2B2 now only provides a smaller dataset with 170 records in the training set and 256 records in the testing set. Theoretically, the same model trained on the larger dataset should perform better than on the smaller dataset. However, our model performed even better than most models trained on the original larger dataset [12,41]. For the works conducted on the smaller dataset, the solution "ELMo (clinical) + BiLSTM–CRF (ensemble)" [24] was the previous state-of-the-art model, and the result shows that our model significantly outperformed the state-of-the-art model in recall and slightly outperformed it in F1 score.

4. Discussion

The experimental results both on the single model and ensemble model showed the ability to improve the system's recall using an additional entity discovery task. According to the results from previous CNER models [7,12,19,24,41,42], those models with a single task always had a relatively lower recall compared to their precision; however, it could be used to discover more entities in practice. In our model, the multitask mechanism was used to balance discovering more entities and correctly identifying entities, and the use of the additional task can be seen as a process to add extra weights to discovering clinical entities. For the models with a single task, the model optimizes parameters only by the loss of the ground-truth tags and estimated tags by cross-entropy. Compared to the models with a single task, the multitask model tends to optimize parameters based primarily on if a token is an entity, and correspondingly reduces the reliability on a token being assigned the right tag. In the backpropagation process, gradients are independent of each other in the softmax and CRF parts. Then, by backpropagating to the encoder parts, the gradients from the two parts are merged; thus, the multitask mechanism mainly changes the way of encoding.

5. Conclusions

In this paper, we firstly discussed recent work on clinical named entity recognition and highlighted the new improvements brought by contextual embeddings. Then, we proposed the multitask Att-biLSTM–CRF model with contextual embeddings. The multitask mechanism separates the entity recognition task into two simultaneous sub-tasks, entity discovery and sequential inference. Our experiment conducted on the 2010 I2B2/VA dataset showed that our model achieved better performance than the previous state-of-the-art solution. Notably, our model improved the recall significantly, which agrees with what we expected.

Our algorithm improved the perception of unknown entities just as we hypothesized, which means that the system should have a better capability to deal with emerging medical concepts without extra training resources. This idea could not only be applied in medical concept extraction, but also other medical named entity recognition applications such as drug names and adverse drug reactions, as well as named entity recognition tasks in other fields. For future studies, we want to put forward two ideas. One is a transfer learning idea. We can already see the improvements brought by the multitask mechanism in this paper, and the multitask mechanism can be seen as a task-oriented regularizer. Therefore, it could be meaningful to train the model for the entity discovery task so as to regularize the model first, and then implement transfer learning to train the same model for the sequential inference task. Another idea is that, in previous work, character-level embedding was very useful for improving the system's performance; thus, it would be worthwhile to build a model with combined character-level embedding and contextual word embedding.

Supplementary Materials: The source code is available at https://github.com/jeffy129/multi_cner.

Author Contributions: Conceptualization, J.Y. and Y.L.; data curation, J.Y. and C.G.; methodology, J.Y. and M.Q.; resources, J.Y.; software, J.Y. and X.Y.; investigation, J.Y. and Y.L.; supervision, M.Q.; validation, J.Y. and M.Q.; visualization, J.Y. and Y.L.

Funding: This research was supported by the China Scholarship Council (CSC) (grant number: 201806360227), also was partially supported the Major Project for the Research and Application of Generic Technologies of National Quality Infrastructure (NQI) (grant number:2018YFF0215803-3) and the Ministry of Education in China (MOE) Project of Humanities and Social Sciences (grant number:18YJA630087).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Meystre, S.M.; Savova, G.K.; Kipper-Schuler, K.C.; Hurdle, J.F. Extracting information from textual documents in the electronic health record: A review of recent research. *Yearb. Med. Inform.* **2008**, *17*, 128–144.
2. Friedman, C.; Alderson, P.O.; Austin, J.H.; Cimino, J.J.; Johnson, S.B. A general natural-language text processor for clinical radiology. *J. Am. Med. Inform. Assoc.* **1994**, *1*, 161–174. [CrossRef] [PubMed]
3. Aronson, A.R.; Lang, F.-M. An overview of MetaMap: Historical perspective and recent advances. *J. Am. Med. Inform. Assoc.* **2010**, *17*, 229–236. [CrossRef] [PubMed]
4. Savova, G.K.; Masanz, J.J.; Ogren, P.V.; Zheng, J.; Sohn, S.; Kipper-Schuler, K.C.; Chute, C.G. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, component evaluation and applications. *J. Am. Med. Inform. Assoc.* **2010**, *17*, 507–513. [CrossRef] [PubMed]
5. Denny, J.C.; Irani, P.R.; Wehbe, F.H.; Smithers, J.D.; Spickard, A., III. The KnowledgeMap project: Development of a concept-based medical school curriculum database. In Proceedings of the AMIA Annual Symposium Proceedings, Washington, DC, USA, 8–12 November 2003; pp. 195–199.
6. Liu, S.; Tang, B.; Chen, Q.; Wang, X. Drug name recognition: Approaches and resources. *Information* **2015**, *6*, 790–810. [CrossRef]
7. Chalapathy, R.; Borzeshi, E.Z.; Piccardi, M. Bidirectional LSTM–CRF for clinical concept extraction. *arXiv* **2016**, arXiv:1611.08373.
8. Uzuner, Ö.; South, B.R.; Shen, S.; DuVall, S.L. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *J. Am. Med. Inform. Assoc.* **2011**, *18*, 552–556. [CrossRef] [PubMed]

9. Pradhan, S.; Elhadad, N.; Chapman, W.; Manandhar, S.; Savova, G. Semeval-2014 task 7: Analysis of clinical text. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 23–24 August 2014; pp. 54–62.
10. Boag, W.; Wacome, K.; Naumann, T.; Rumshisky, A. CliNER: A lightweight tool for clinical named entity recognition. In Proceedings of the AMIA Joint Summits on Clinical Research Informatics, San Francisco, CA, USA, 23–25 March 2015.
11. Wang, Y.; Patrick, J. Cascading classifiers for named entity recognition in clinical notes. In Proceedings of the Workshop on Biomedical Information Extraction, Association for Computational Linguistics, Borovets, Bulgaria, 14–16 September 2009; pp. 42–49.
12. DeBruijn, B.; Cherry, C.; Kiritchenko, S.; Martin, J.; Zhu, X. Machine-learned solutions for three stages of clinical information extraction: The state of the art at i2b2 2010. *J. Am. Med. Inform. Assoc.* **2011**, *18*, 557–562. [CrossRef]
13. Lafferty, J.; McCallum, A.; Pereira, F.C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the 18th International Conference on Machine Learning (ICML 2001), Williamstown, MA, USA, 28 June–July 1 2001; pp. 282–289.
14. Wu, Y.; Jiang, M.; Xu, J.; Zhi, D.; Xu, H. Clinical named entity recognition using deep learning models. In Proceedings of the AMIA Annual Symposium Proceedings, Washington, DC, USA, 4–8 November 2017; pp. 1812–1819.
15. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
16. Unanue, I.J.; Borzeshi, E.Z.; Piccardi, M. Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition. *J. Biomed. Inform.* **2017**, *76*, 102–109. [CrossRef]
17. 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*, 1381–1388. [CrossRef] [PubMed]
18. Zeng, D.; Sun, C.; Lin, L.; Liu, B. LSTM–CRF for drug-named entity recognition. *Entropy* **2017**, *19*, 283. [CrossRef]
19. Xu, G.; Wang, C.; He, X. Improving clinical named entity recognition with global neural attention. In Proceedings of the Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, Macau, China, 23–25 July 2018; pp. 264–279.
20. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.
21. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
22. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
23. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
24. Zhu, H.; Paschalidis, I.C.; Tahmasebi, A. Clinical Concept Extraction with Contextual Word Embedding. *arXiv* **2018**, arXiv:1810.10566.
25. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning (ICML 2008), Helsinki, Finland, 5–9 July 2008; pp. 160–167.
26. Aguilar, G.; Maharjan, S.; López-Monroy, A.P.; Solorio, T. A Multi-task Approach for Named Entity Recognition in Social Media Data. In Proceedings of the Third Workshop on Noisy User-generated Text of Association for Computational Linguistics, Copenhagen, Denmark, 7 September 2017; pp. 148–153.
27. Savova, G.K.; Fan, J.; Ye, Z.; Murphy, S.P.; Zheng, J.; Chute, C.G.; Kullo, I.J. Discovering peripheral arterial disease cases from radiology notes using natural language processing. In Proceedings of the AMIA Annual Symposium Proceedings, Washington, DC, USA, 13–17 November 2010; pp. 722–726.
28. Bedmar, I.S.; Martínez, P.; Herrero Zazo, M. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). In Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013), Atlanta, GA, USA, 13–14 June 2013; pp. 341–350.

29. Wang, Y.; Wang, L.; Rastegar-Mojarad, M.; Moon, S.; Shen, F.; Afzal, N.; Liu, S.; Zeng, Y.; Mehrabi, S.; Sohn, S. Clinical information extraction applications: A literature review. *J. Biomed. Inform.* **2018**, *77*, 34–49. [CrossRef]
30. Hebbring, S.J. The challenges, advantages and future of phenome—Wide association studies. *Immunology* **2014**, *141*, 157–165. [CrossRef]
31. Law, V.; Knox, C.; Djoumbou, Y.; Jewison, T.; Guo, A.C.; Liu, Y.; Maciejewski, A.; Arndt, D.; Wilson, M.; Neveu, V. DrugBank 4.0: Shedding new light on drug metabolism. *Nucleic Acids Res.* **2013**, *42*, D1091–D1097. [CrossRef]
32. Roberts, K.; Rink, B.; Harabagiu, S.M.; Scheuermann, R.H.; Toomay, S.; Browning, T.; Bosler, T.; Peshock, R. A machine learning approach for identifying anatomical locations of actionable findings in radiology reports. In Proceedings of the AMIA Annual Symposium Proceedings, Chicago, IL, USA, 3–7 November 2012; pp. 779–788.
33. Sarker, A.; Gonzalez, G. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *J. Biomed. Inform.* **2015**, *53*, 196–207. [CrossRef]
34. Rochefort, C.M.; Buckeridge, D.L.; Forster, A.J. Accuracy of using automated methods for detecting adverse events from electronic health record data: A research protocol. *Implement. Sci.* **2015**, *10*, 5. [CrossRef]
35. Deleger, L.; Brodzinski, H.; Zhai, H.; Li, Q.; Lingren, T.; Kirkendall, E.S.; Alessandrini, E.; Solti, I. Developing and evaluating an automated appendicitis risk stratification algorithm for pediatric patients in the emergency department. *J. Am. Med. Inform. Assoc.* **2013**, *20*, e212–e220. [CrossRef]
36. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
37. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM–CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
38. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural architectures for named entity recognition. *arXiv* **2016**, arXiv:1603.01360.
39. Xu, K.; Zhou, Z.; Hao, T.; Liu, W. A bidirectional LSTM and conditional random fields approach to medical named entity recognition. In Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2015 (AISI 2015), Beni Suef, Egypt, 28–30 November 2015; pp. 355–365.
40. Habibi, M.; Weber, L.; Neves, M.; Wiegandt, D.L.; Leser, U. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics* **2017**, *33*, i37–i48. [CrossRef] [PubMed]
41. Liu, Z.; Yang, M.; Wang, X.; Chen, Q.; Tang, B.; Wang, Z.; Xu, H. Entity recognition from clinical texts via recurrent neural network. *BMC Med. Inform. Decis. Mak.* **2017**, *17*, 67. [CrossRef]
42. Si, Y.; Wang, J.; Xu, H.; Roberts, K. Enhancing Clinical Concept Extraction with Contextual Embedding. *arXiv* **2019**, arXiv:1902.08691. [CrossRef] [PubMed]
43. Jin, Q.; Liu, J.; Lu, X. Deep Contextualized Biomedical Abbreviation Expansion. *arXiv* **2019**, arXiv:1906.03360.
44. Jin, Q.; Dhingra, B.; Cohen, W.W.; Lu, X. Probing biomedical embeddings from language models. *arXiv* **2019**, arXiv:1904.02181.
45. Johnson, A.E.; Pollard, T.J.; Shen, L.; Li-wei, H.L.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L.A.; Mark, R.G. MIMIC-III, a freely accessible critical care database. *Sci. Data* **2016**, *3*, 160035. [CrossRef] [PubMed]
46. Mikolov, T.; Karafiát, M.; Burget, L.; Černocký, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the 17th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, 26–30 September 2010; pp. 1045–1048.
47. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on Machine Learning (ICML 2013), Atlanta, GA, USA, 16–21 June 2013; pp. 1310–1318.
48. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
49. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
50. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
51. I2B2. Evaluation Methods and Procedures for 2010 i2b2/VA Challenge. Available online: https://www.i2b2.org/NLP/Relations/assets/Evaluation%20methods%20for%202010%20Challenge.pdf (accessed on 25 May 2019).

52. Boag, W.; Sergeeva, E.; Kulshreshtha, S.; Szolovits, P.; Rumshisky, A.; Naumann, T. CliNER 2.0: Accessible and Accurate Clinical Concept Extraction. *arXiv* **2018**, arXiv:1803.02245.

53. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the 1995 International Joint Conference on AI, Montreal, QC, Canada, 20–25 August 1995; pp. 1137–1145.