Information Sciences 602 (2022) 184–200

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

BINER: A low-cost biomedical named entity recognition

Mohsen Asghari [a,*], Daniel Sierra-Sosa [b], Adel S. Elmaghraby [a]

[a] Department of Computer Science and Engineering, University of Louisville, KY, USA
[b] Department of Computer Science and Information Technology, Hood College, Frederick, MD, USA

ARTICLE INFO

Article history:
Received 28 March 2021
Received in revised form 24 February 2022
Accepted 19 April 2022
Available online 22 April 2022

Keywords:
Natural Language Processing
Named entity recognition
Deep learning
Biomedical text
Transfer Learning
Computational efficiency

ABSTRACT

A primary focus of the healthcare industry is to improve patient experience and quality of service. Practitioners and health workers are generating large volumes of text that are captured in Electronic Medical Records, clinical reports, and publications. Additionally, patients post millions of comments on social media related to healthcare, on diverse topics such as hospital services, disease symptoms, and drugs effects. Unifying various data sources can guide physicians and healthcare workers to avoid unnecessary, irrelevant information and expedite access to helpful information. The main challenge to creating Biomedical Natural Language Understanding is the lack of standard datasets and the extensive computational resources needed to develop different models. This paper proposes a model trained on low-tier GPU computers, producing comparable results to larger models like BioBERT. We propose BINER, a Biomedical Named Entity Recognition architecture using limited data and computational resources.

1. Introduction

A Natural Language Processing (NLP) system processes the structure of a text, for many purposes such as Information Extraction, Question Answering [1] Topic modeling and Trend analysis [2]. Using these algorithms for healthcare will contribute to understanding the correlation between disease and chemicals in Electronic Health Record (EHR), studying adverse drug reactions, capturing health related trends in noisy social media [3], and perform automatic summarising [4]. Before getting to these tasks, we need to solve and improve techniques such as Part of Speech (POS) detection and Named Entity Recognition (NER) in healthcare, which are challenging due to lack of access to reasonable-sized and high-quality of annotated databases, along with the lack of tools and resource to build practical NLP solutions. Some of the traditional techniques (Conditional Random Field and Hidden Markov Models) require human intervention, including exact word spelling and external resource generation like dictionaries and lexicons. For example, using Lexicons to enhance the embedding layer [5], or simply using a lookup table and a gazetteer for NER [6]. Using experts in healthcare to generate high-quality solutions is expensive and time consuming. By using modern techniques such as Deep Learning to improve POS and NER tasks in healthcare will reduces the cost of human labeling and improves model performance [7].

Bidirectional Encoder Representation from Transformers (BERT) [8] is the most recent technique for many autonomous tasks, such as text classification, question answering, and NER. According to BERT developers, training using the Base version of BERT with 12 Layers, 768 Hidden sizes, and 12 Self Attention heads (110 Million Total parameters) is accomplished in four days using four cloud Tensor Processing Units (TPUs). The Large version of BERT, 24 layers and 1024 hidden layers and 16

* Corresponding author.
E-mail addresses: m0asgh02@louisville.edu (M. Asghari), sierra-sosa@hood.edu (D. Sierra-Sosa), adel@louisville.edu (A.S. Elmaghraby).

https://doi.org/10.1016/j.ins.2022.04.037
0020-0255/

self-attention heads (i.e., 340 Million total parameters) accomplishes training in four days using 16 cloud TPUs. BERT needs one GPU for a Base and one TPU for a Large BERT in deployment. This model is a general language model and not a specific domain model, such as health care, which opens an area of research and application to train domain-specific BERT. BioBert [9] has been proposed as a pre-trained biomedical model. According to researchers, it takes approximately 23 days to train using eight NVIDIA V100 GPU.

In this research, we identified two challenges to training. First, the current state-of-the-art technique, BERT, requires the use of extensive computational resources [8,9]. Second, the original BERT implementation is not domain-specific, and therefore must be modified to model target specific domains, e.g., the biomedical field. We propose that addressing these challenges will lead to improved results in the health care domain.

In the present study, we evaluate an alternative implementation of different architecture by using various Embedding Layers combined with Bidirectional LSTM and Conditional Random Fields to achieve domain-specific NER in the Biomedical field. Our results compare favorably with BioBERT when testing using six different standard healthcare datasets. We surpassed the BioBert F1-Score by 3% on Disease detection, 8% on Protein detection (e.g., contains DNA, RNA, and Cells) and 9% on Gene detection. In addition, we used the proposed architecture to transfer the knowledge and extend the model for more challenging environment to detect chemicals in clinical notes, which these notes have more noise and ambiguity compare to the well written and structured publications.

2. Related Work

This study evaluates state-of-the-art NER models. We found several NER models that require attention. The majority of new state-of-the-art models that use deep learning are computationally expensive and typically operate offline. Designing accurate models without using high-end cloud services or expensive graphical processing units (GPU) is necessary. Our research focused on biomedical textual data, and we designed a computationally low-cost deep learning solution.

BioBERT is often used for research related to our work applying NER to deep learning [9]. It is designed for multi-task NER, relation extraction, and question answering problems. BioBERT reports 0.62% improvement for medical NER. This benchmark uses over nine annotated datasets. The lowest F1-Score is 71.11% (JNLPBA dataset) for detecting Gene/Protein, and 93.44% for the highest F1-Score for BC5CDR when searching for Drug/Chem names. We compare our proposed architecture with BioBert results using over five standard datasets for NER tasks.

Crichton et al. studied the efficiency of using Multi-output multi-task models compared to a single task model [10]. They show on average, the multi-task model produces better NER results for biomedical textual data. Their research utilizes Look up Embedding combinations with convolutional neural networks (CNNs), which need to be extended and experimented with other types of architecture and embedding layers. Wang et al. studied NER based multi-task learning of biomedical text by combining the character and word levels [11]. They proposed three architectures, Multi-Task Model Character (MTM-C), Multi-Task Model Word (MTM-W), and Multi-Task Model Character-Word (MTM-CW). MTM-C uses character level embedding and MTM-W uses word-level embedding. The MTM-CW merges Character-Level embedding and Word-Level embedding. All these models will end up in a CRF layer to learn the word sequence and identify them in a sentence.

S.K. Hong et al. proposed a deep learning label-label transition model [12]; their work is concerned with correctly segmenting entities. This is because entities are usually a complex and long combination of several words. The proposed DTranNER model combines two deep-learning-based networks capable of detecting label sequences in the prediction stage. Their design used two CRF layers while we used a unified deep learning-based architecture with two CRF targets to separate the labeling with word segmentation. Our model surpasses their results by an average 2.8% higher accuracy.

Marcove et al. [13] focus on multi-task learning (MTL) for Opinion Role Labeling (ORL) and Semantic Role Labeling (SRL) by introducing the SRL4ORL model. Their research approach uses learning techniques for SRL and ORL. They used opinion labeling on a dataset that suffers from labeled data scarcity; they used Semantic Role labeling as input to the model to improve the model; as a result, they introduce a fully shared and hierarchical model.

Ma and Hovey At Carnegie Mellon University [14] research the same neural network proposed by Chiu and Nichols [15] with an additional Conditional Random Fields (CRF) layer. They proposed a model without feature engineering and reduced prepossessing cost. They claim that this model is useful for an extensive range of sequence labeling. This model evaluates two tasks; Part of Speech tagging and NER with 97.55% and 91.21% F1-Scores, respectively. Their research focus is on general non-domain-specific NER. They combine the conditional random fields with BiLSTM layers, making it a relevant case to compare with our work. We train their network over the standard biomedical datasets and report their F1-Scores to compare with our results.

The other aspect of our research focuses on different embedding systems over Biomedical text sources. Several research studies show that studying word distribution is not enough for NLP tasks such as POS and NER. As an example, at IBM, Santos and Zadrozeny use CNNs at the character level to improve the POS tagging problem [16]. They develop a language-dependent model and experiment with it on English and Portuguese with a reported accuracy of 97%. Chui and Nicholas at British Columbia present an NER model [15] using BiLSTM on top of a character level CNN; their model obtains an F1-Score of 91.62%. In this research, we compare four different Embedding layers, RNN-Character level, CNN-Character level, LookUp, and Self-attended Encoder (all of these will be discussed in Section 3), and evaluate the effect of these layers on Biomedical text sources.

Huang et al. proposed a low-cost NER model [17]. However; they focus on reducing the cost of manually labeling datasets. Their contribution is designing a model that requires less manual labeling while maintaining recognition ability. Our research focus is developing a computationally lower-cost method by using less resources to train our model, and we measure the training size effect on transfer learning by using medical literature to train and retrain the same model to recognize clinical notes.

In this work, we focus on building a deep learning model by concentrate on multi-task learning at the CRF layer and learn word segmentation and entities to combine the learning values for better outcomes. In addition, we study the effect of different embedding layers on different proposed architectures and compare our results across six standard healthcare datasets.

3. Research Methods

This section focuses on the architecture's design to do multi-task learning (MTL). The first task is word segmentation, targeting complex entities in a sentence composed of multiple words. To train this task, we used the Begin-Inside-Outside (BIO) annotation format. This format's advantage is that we can train multiple models to group words together. The BIO format design helps each model to have a better understanding of complex entities. The second task is to identify the relevant entities. We will discuss the necessary components needed to design our proposed neural network. We will discuss three different architectures to address multi-task learning to improve the NER task in medical text.

3.1. Bidirectional Long Short Term Memory

Detecting the entity of a word or a phrase depends on context. To find a correlation between words in a sentence using a traditional neural network would be ineffective because of the complexity of a sentence. A sentence is defined as a sequence of words; as we know, Recurrent Neural Network (RNN) architectures are designed for learning sequences, which is a good choice for sentence analysis. RNN is often utilized in language modeling [18,19]; however, one of the drawbacks to using a simple RNN is the difficulty of this network in capturing long-distance correlations in a sequence. This problem is known as the vanishing gradient problem. Thus, researchers introduced the Long Short Term Memory (LSTM) model to overcome the RNN limitation. In the LSTM network, processed sequences can be transferred to an additional stage if needed, providing a time frame for representing word locations in a sequence. The LSTM saves the information to be used later in the sequence. A sequence will move forward in time; however, we need to go backward and forward in some cases; this means that information must be transformed to the past and future to discover the correlation in NER. An entity is a set of related words. A relevant word or words will identify each entity in a sentence. Therefore we used a Bidirectional LSTM (BiLSTM) model to use the information in two directions.

3.2. Embedding Layer

The embedding layer creates a mapping between words and dense numerical matrices in Natural Language Processing (NLP) to generate proper inputs to an Artificial Neural Network (ANN). We study this word and character layer. First, Word-Level embedding, which represents each word as a vector, requires a vocabulary of words, V, gathered from all the documents with size $|V|$, and embedding size D, which represents the size of a vector defining a word. Therefore, we have a matrix with size $|V| * D$, where each row refers to an individual word, behaving like a lookup table. At this level, a predefined word-level embedding such as Stanford's GLOVE or Google's Word2Vector [20,21] can be used. Kitaev and Klein did valuable research on the effectiveness of the encoder on the performance of the parser and presented a Self-Attentive Encoder (SAE) [22]. Later, Peters et al. introduced a model that enhanced the SAE to create Elmo (Embedding from Language Models) [23].

Researchers demonstrated that studying morphological features of words such as word suffix and prefix would bring added value to the analysis [14,16]. This type of text representation is known as character level embedding. This research utilized two types of character level embedding, first a recurrent neural network (RNN) and second a convolution neural network (CNN). Fig. 1 illustrates how a tokenization level was connected to a Word-Level or Character-Level in our preprocessing stage. Our embedding layer performs the word and character level tokenization and prepares the model's input.

3.3. Conditional Random Field Layer

A Conditional Random Field (CRF) [24] is used for segmenting and labeling sequence data. Researches show that the CRF technique at the sentence level achieves better results compared to individual word analysis methods like the Maximum Entropy Markov Model (MEMM) or the Hidden Markov Model (HMM) [25,26]. This CRF characteristic draws a researcher's attention to using this technique for NER tasks. CRF combined with BiLSTM could improve the outcome of sequence analysis [27]. Fig. 2 explains how these two layers can contribute to each other. The figure illustrates how a sentence at the bottom of the figure goes through the LSTM blocks and then the CRF layer to predict the output, which is the entity type of each word.

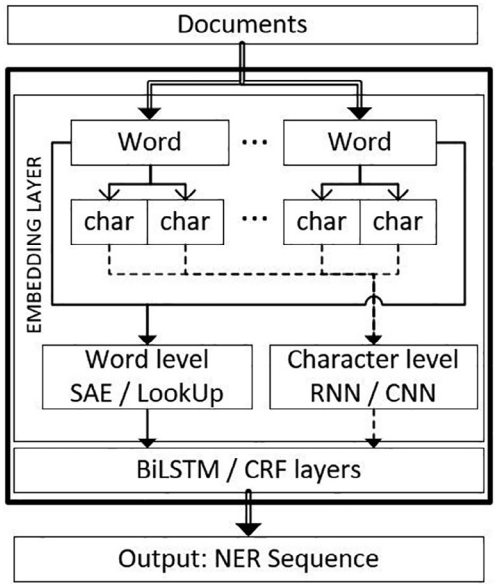

Fig. 1. Character level implemented by CNN and RNN, Word level implemented by Self-Attentive Encoder (SAE) and Lookup.

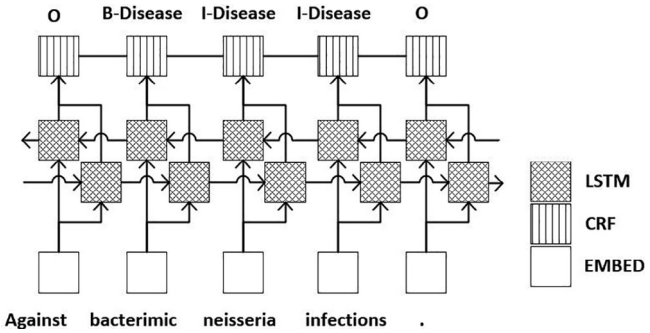

Fig. 2. Transformation of a sentence to output by connecting Embed, Bidirectional LSTM,and CRF.

The CRF loss function is defined as a matrix of scores, where A is a sentence formed as a sequence defined by $\left[[x]_1^T\right]$, then we have a function to calculate the scores $f_\theta\left([x]_1^T\right)$. In the scoring matrix we defined each element by $[f_\theta]_{i,t}$ where i is index of tag outputs and t addresses each word. This matrix represents a position-independent scoring system. Eq. 1 defines the scoring formula [27].

We enhance this level of architecture by separating the output into two sections: BIO (Begin, Inside, Outside) to segment words, and Entity level, which classifies words (Fig. 3). This network efficiently uses segmentation results to enhance the labeling of a sequence, and we call this approach BINER. In the next section, we will discuss three implementations of our proposed approach.

$$s\left([x]_1^T, [i]_1^T, \theta\right) = \sum_{t=1}^{T}\left([A]_{[i]_{t-1},[i]_t} + [f_\theta]_{[i]_t,t}\right) \tag{1}$$

3.4. Proposed Approach

We introduce three different BINER implementations using three ANN architectures. We compare and analyze each architecture's results and find the model with the best NER results on Biomedical text.

3.4.1. Base Implementation

The first implementation seen in Fig. 4 uses a shared BiLSTM layer to train the model and implement two CRFs. One for word segmentation, described by CRF_{iob}, and the other for sequence labeling CRF_{ner}. CRF layers use the Log-Sum-Exp (LSE) as

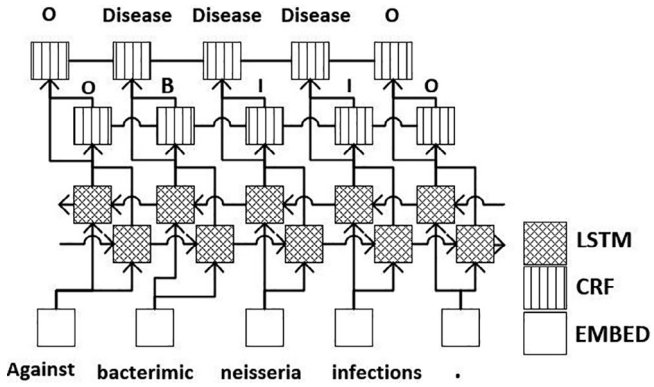

Fig. 3. Transformation of a sentence to output by Connecting Embed, Bidirectional ILSTM and CRF in BINER Approach.

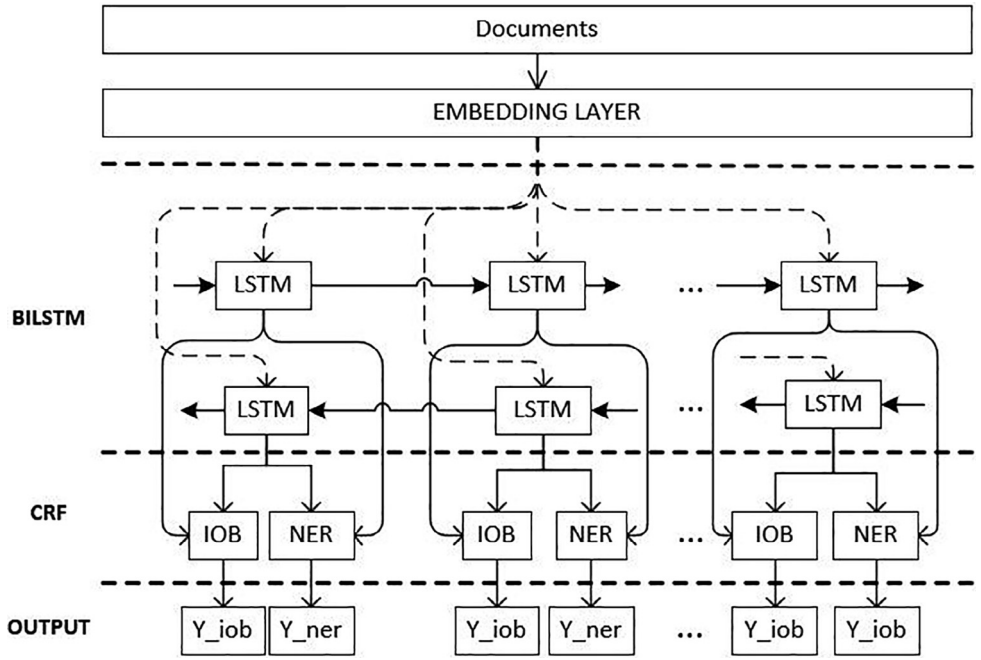

Fig. 4. Connection between layers in Base-BINER Implementation.

the loss function, where x is the sequence and t is sequence length shown in Eq. 2. The equation will calculate the true sequence and predicted sequence, then the loss function will calculate the mean square error. To aggregate the loss function for both word segmenting and labeling, we calculate the Mean Square Error (MSE) using LSE. Prediction results for individual CRFs are calculated using the LSE, e.g., S'_{iob} for CRF_{iob} and S'_{ner} for CRF_{ner}. Using S as LSE and n as the number of entities in a document, we calculate the MSE via Eq. 3.

$$S = x^* + log\left(\sum_{1}^{t} exp(x_i - x^*)\right) where \quad x^* = max_1^t\{x_i\} \tag{2}$$

$$MSE = \frac{1}{n}\sum_{1}^{n}\left[\left[(S_{iob} + S_{ner}) - (S'_{iob} + S'_{ner})\right]^2\right] \tag{3}$$

3.4.2. Parallel Implementation

The second architecture is very similar to the basic implementation with the difference that it contains individual BiLSTM and CRF for word segmentation and Labeling, respectively. In this architecture we add an extra layer as the concat layer. The

two sections of the learning/training is done in parallel and combined in concat layer; this architecture is the Parallel BINER step. Fig. 5 shows the architectures in detail. This architecture follows the same loss function equation as represented in Eq. 3.

3.4.3. Sequential Implementation

In contrast to the parallel architecture, the third architecture integrates the two sections in sequence for learning. The first section is word segmenting, and feeds the output into the labeling section. In our experiment, we test the reverse as well. This means that we use labeling first and then segmentation. Based on our experiments we observed that there was no difference in learning. The sequential BINER step of our implementation performs a sequential ordering. Fig. 6 illustrates how the layers are connected. To induce propagation in the neural network, we change the loss function using the Mean of Errors (ME). The ME is calculated via Eq. 4, where the LSE, S', refers to the last CRF score calculated for CRF_{iob} and CRF_{ner}, where n represents the length of a given sequence.

$$ME = \frac{1}{n}\sum_{1}^{n}[(S - S')] \tag{4}$$

Sequence architecture requires multiple loss functions, so we used *retain-graph* property in PyTorch v1.8 framework to back-propagate each loss function individually.

4. Material and methods

In this section, we describe the datasets, evaluation metrics and methodology used to conduct the experiments. The experiments will test the proposed approach. We will use the results to compare our method to existing methods.

4.1. Datasets

There are several publicly available standard datasets to train and evaluate NER tasks. Table 1 shows a list of datasets utilized in this research with some of their features. We selected six different datasets in the biomedical domain in addition to the LINNAEUS dataset, which focuses on species, to challenge our approaches and evaluate them for cross-domain adaptation. The BIO format (Begin, Inside, Outside) is used to prepare the datasets and tag Entities. All the datasets are preprocessed and separated into three files *Train* includes 70%, *Validation* includes 10%, and *Test* includes 20% of the dataset.

The JNLPBA [28] dataset is driven from GENIA corpus and annotated to be used as a ground truth to detect *Protein*, *RNA*, *Cell Line*, *Cell Type*, and *DNA*. BioCreative II Gene Mention (BC2GM) [29] is a benchmark dataset to train for NER tasks and has been utilized by several researchers to detect gene names such as BANNER, GLIMI, and BioBERT created by [30,31,9], respectively. Pathway Curation was the main task in BioNLP 2013 [32]; this dataset was designed to tackle the event extraction in

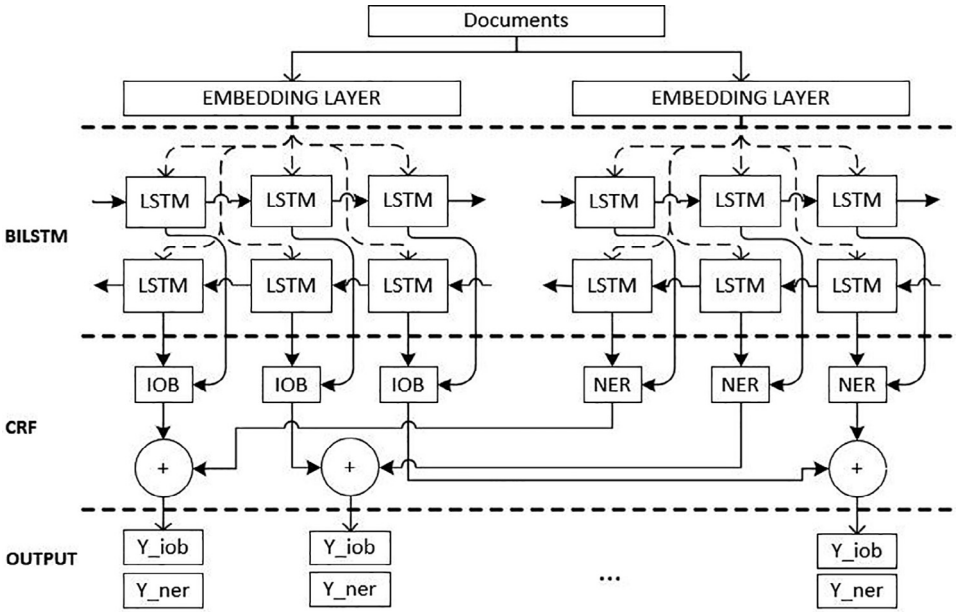

Fig. 5. Connection between layers in parallel BINER implementation.

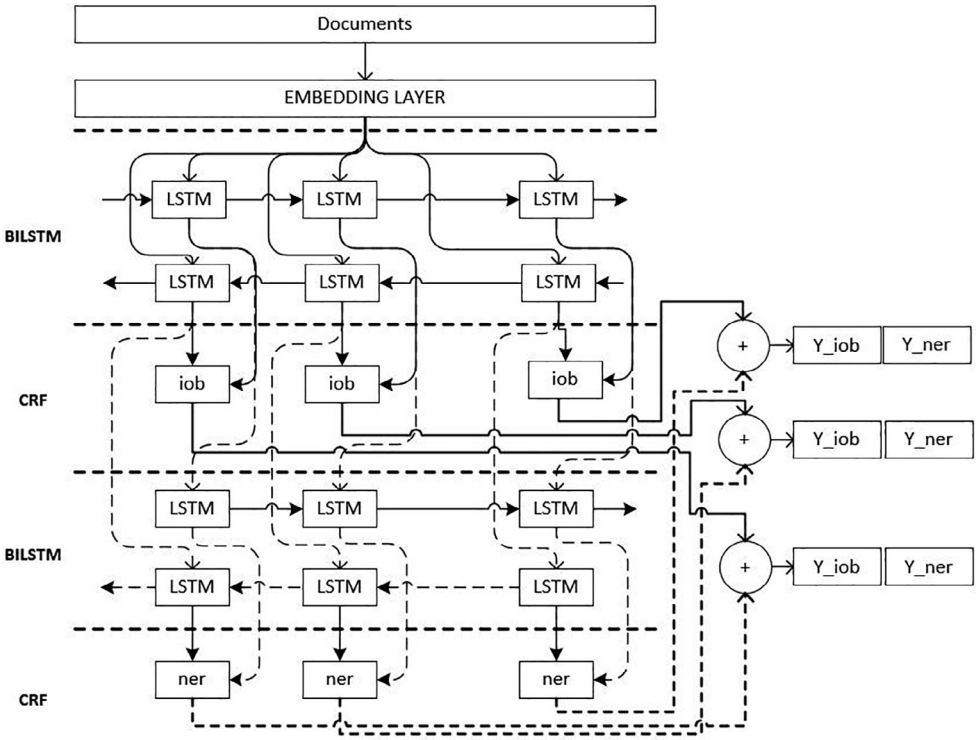

Fig. 6. Connection between layers in Sequential BINER implementation.

Table 1
Type and Frequency of entities in each Dataset.

Dataset	Type	Entity Frequency
JNLPBA	Gene/Protein	dna: 8,392, protein: 27,032, cell_type: 6,177, cell_line: 3,380, rna: 837
BIONLP13PC	Gene	gene: 5,399, complex: 719, cellular: 464, chemical: 1,155
BC2GM	Gene	gene: 15,047
LINNAEUS	Species	species: 2,103
NCBI-DISEASE	Disease	disease: 5,118
BC5CDR	Disease/Chemical	chemical: 5,185 disease: 4,098
ADEs	Chemical	chemical: 19,100

medical text to support curation. Before conducting any event extraction, it is necessary to identify the entities so we used this dataset to detect *Gene or Gene Product*, *Complex*, *Cellular-Component*, and *Simple-Chemical*.

The LINNAEUS [33] dataset normalized and recognized the species name mentioned in a medical text, the version used in this research contains 100 random full-text documents converted to standoff format. One of the leading entities in the medical text is Disease names. We selected the NCBI-disease [34] Test to evaluate our models for this task. This dataset contains 793 PubMed abstracts and contains 790 unique disease names. The BioCreative V Chemical Disease Relation (BC5CDR) [35] dataset has a combination of two entities *Chemical* and *Disease* extracted by humans from 1,500 PubMed articles. More datasets are available to be addressed, but we selected these six datasets to cover different types of entities and evaluate our neural network architectures in different areas. The summarization of all the datasets are shown in Table 1. The Adverse Drug Events (ADEs) [36] collection is another dataset that creates a set of medical documents where physicians document drug-related information, including drug names, dosage, strength, duration, frequency, form, and reason. This dataset detects medication and creates two types of relationships: drugs used for specific symptoms and disease and drug with adverse events. The dataset includes a total of 505 documents; 202 are used for testing, and 303 are used for training.

4.2. Evaluation Metrics

To train and evaluate the models, we split the data into three different sets: training, validation, and test. We use the validation dataset to assess the model while training. This dataset helps us have an indicator to select the best state of the model during training. To be more accurate and remove the bias, we did not use the best-achieved F1-Score in training; instead, we used the Macro F1-Score as our metric. Optiz and Burst discuss two types of Macro F1-Scores, "Average F1" and "F1 of Averages" [37]. Eq. 7 expresses the formula for the F1-Score, which is the harmonic (H) mean of Precision (P) and Recall (R) calculated by Eqs. 5 and 6, respectively. where $TP = TruePositive, TN = TrueNegative, FP = FalsePositive$, and $FN = FalseNegative$.

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

$$F1 = H(P, R) = \frac{2 * P * R}{P + R} \tag{7}$$

If x is the number of times that we repeat the experiment, then we can calculate the "Average F1" via Eq. 8 and "F1 of Average" via Eq. 9, defined as follows:

$$\mathcal{F}_1 = 1/n \sum_x F1_x = 1/n \sum_x \frac{2 * P * R}{P + R} \tag{8}$$

$$\mathbb{F}_1 = H(\overline{P}, \overline{R}) = \frac{2 * \overline{P} * \overline{R}}{\overline{P} + \overline{R}} = 2 * \frac{\left(\frac{1}{n}\sum_x P_x\right)\left(\frac{1}{n}\sum_x R_x\right)}{\frac{1}{n}\sum_x P_x + \frac{1}{n}\sum_x R_x} \tag{9}$$

Optiz and Burst state the "F1 of Average" [37] is a heavily biased metric, and in some cases, it can be misleading as an evaluation metric. This situation is more likely to happen when the dataset is imbalanced. In this research, all the datasets are imbalanced, therefore, we report the result based on "Average F1." For simplicity, we call the "Average F1" an F1-Score.

4.3. Training and Hyper Parameters Tuning

After defining the model's architecture, we use all the databases described in Section 3.4 and evaluate the models based on evaluation metrics defined in Section 4.2. Fig. 7 shows how we perform the experiment. We split each database randomly, without repetition, into three sections, training (70%), evaluation (10%), and testing (20%). Each experiment is defined by using different hyper-parameters; for example, Learning rate 0.0001, batch size 8, Hidden Layer Size 100, Drop out 0.3, Embed size 100, and Adam Optimization Function is a description of an experiment. We train each model by 100 epochs. The first purpose of the evaluation metrics is to select the best model using the evaluation dataset. According to the combination of selected ranges for each hyperparameter, each model implementation for each database has 3,600 experiments. The output of each experiment becomes a model. In the test section, we use the test dataset and evaluation metrics to select the winning model of the 3,600 models tested. In this process, we carefully follow the training process to avoid the trap of overfitting our model; therefore, we report all the diagrams for each parameter that we tuned. In the end, we compared our result to two state-of-art techniques BioBERT and DTranNER and conclude the best model with the values of the hyperparameters.

In this section, we discuss the hyperparameter tuning procedure and use The NCBI-Disease dataset as an example. Epochs, Embedding Size, Learning Rate, Hidden Layer Size, and Batch Size are the parameters we discuss individually. We report the best values for each implementation via Table 2.

4.3.1. Epochs

'Epoch' is the number of iterations of the training set for a model. Choosing too many epochs will increase the risk of overfitting, and the model would end up losing generality. Therefore, choosing the correct number of epochs has a high impact on the model. Figs. 8 show the F1-Score over the number of epochs for different Embedding layers for the three proposed architectures using the NCBI-Disease dataset as an example. In this example, we can conclude that Parallel BINER performs better than the other implementations. With this implementation, we can decide that lookup word-level embedding has the better output. We can see that for the parallel and lookup tables, we need to stop before 100 epochs. This example explains the effect the number of epochs has when keeping the other parameters fixed. However, we cannot make the final decision just by the epoch of an embedding layer.

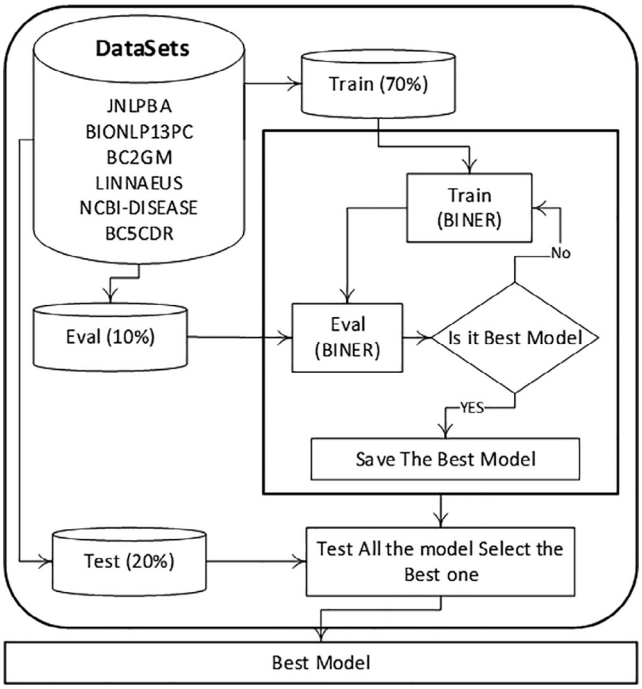

Fig. 7. Process of Training, Evaluation, and Test a Model to Select the Best Model With Proper Hyper-parameters.

Table 2
Shows the Hyper-parameters list for each Implementation.

	Models		
Hyper-parameters	BINER sequential	BINER parallel	BINER basic
Epochs	60	60	100
Learning Rate	0.0002	0.001	0.0001
Batch Size	16	64	8
Hidden Layer Size	500	400	400
Drop Out	0.5	0.2	0.4
Embed Size	200	300	150
Optimization Function	Adam	Adam	Adam

4.3.2. Embedding Size

'Embedding Size' is the maximum length of a sentence in our model, explaining how much padding or cropping we might have to perform to fit the sentences to our model. A parameter that is too small will cause us to drop words from the sentences, and a parameter that is too large will require adding padding to some sentences. Thus, finding the best number for this hyper-parameter can play a significant role in the model performance. In Fig. 9 we present the results when we select LookUp Word level and we simply train the model for 100 epochs from the three proposed implementations for NCBI based on F1-Score. The chart shows 200 for Sequential BINER, 150 for Basic BINER, and 200 for Parallel BINER. In conclusion, the experiment on the NCBI dataset shows 200 Embed size for Sequential BINER implementation performs more solidly.

4.3.3. Learning Rate

The learning rate parameter is used for optimizing the weights in a neural network. It can change depending on the type of optimization function and architecture we use. For example, selecting a large learning rate might pass the optimum point, making the system unstable to converge. Choosing a small learning rate might require more epochs and iterations for training. So, we need to select the correct learning rate to achieve weight optimization. We train our models using different learning rates and compare them using F1-Score. Fig. 10 shows that the model will converge and have better results when choosing 0.0001 or 0.0002 as the learning rate for the NCBI-Disease dataset. In this figure, we keep the best parameters for each model. For instance, for Sequential BINER we used 200 as Embedding Size, used lookup as the embedding, and we train the model for 100 epochs.

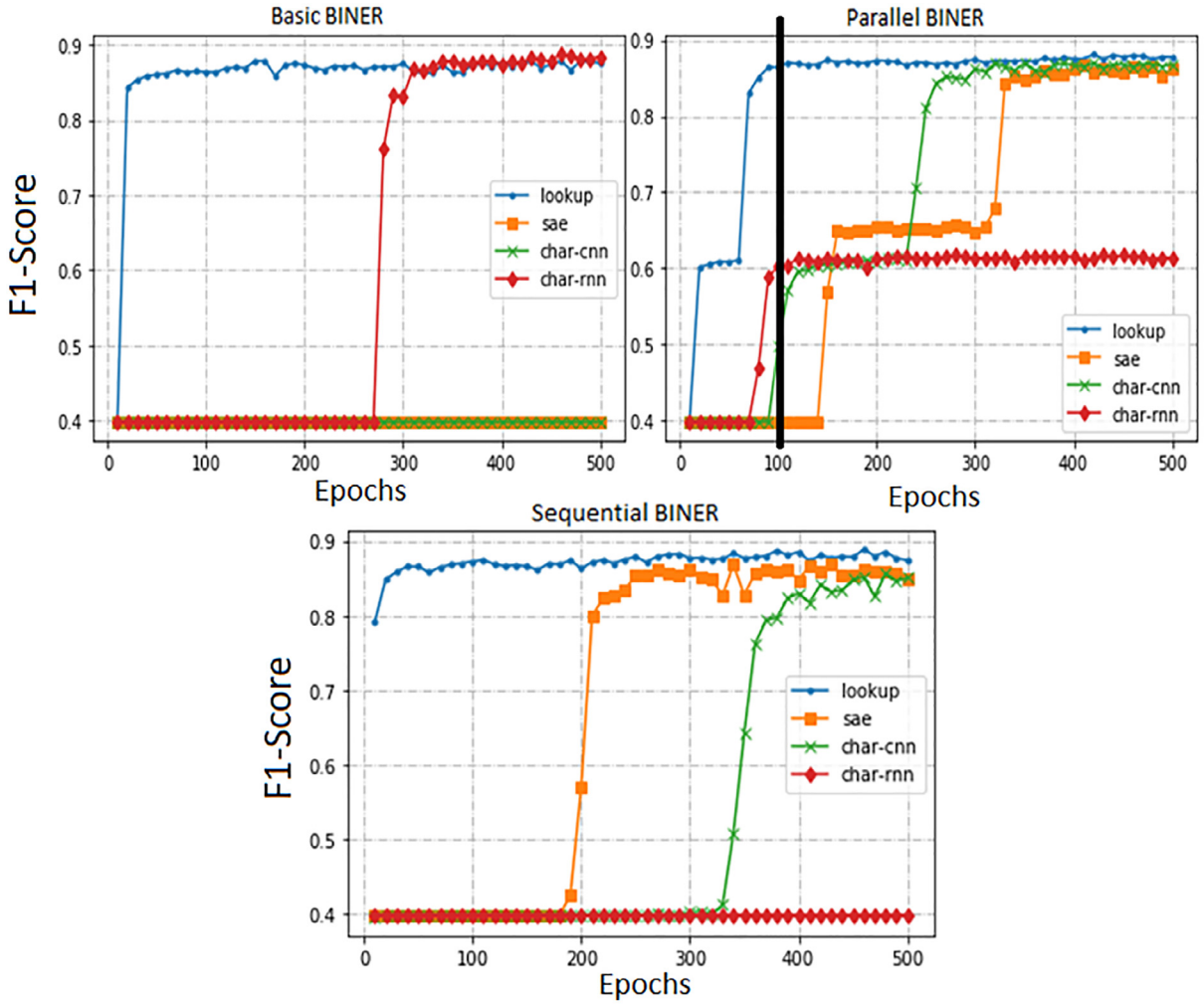

Fig. 8. Epochs and F1-Score for Basic, Parallel, and Sequential BINER implementations.

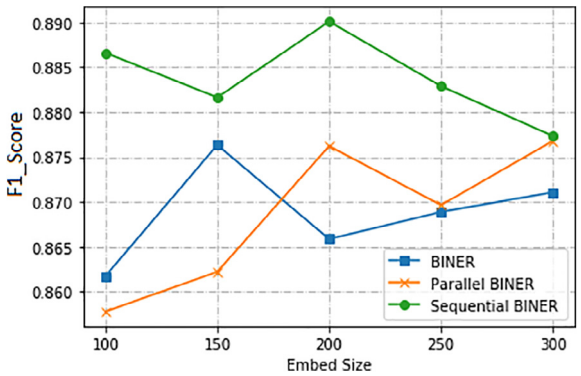

Fig. 9. Relationship between Embedding Size and F1-Score.

4.3.4. Hidden Layers Size

The hidden layer in a neural model defines the level of complexity. To choose the best number of nodes, we need to train a variety of models to study the effect of nodes on model performance. Selecting too few nodes will cause a limitation in learning all the patterns, inducing information bottlenecks. However, selecting too many nodes will increase the model's com-

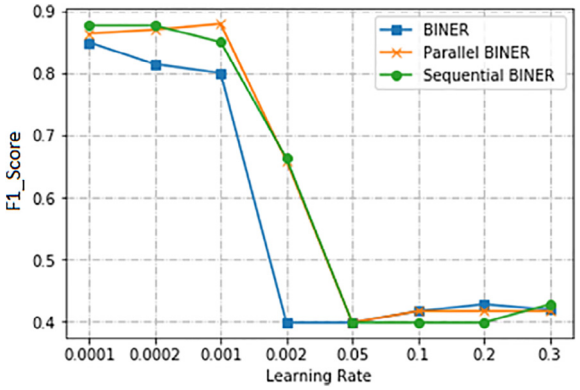

Fig. 10. Relationship between Learning Rate and F1-Score.

plexity and slow the convergence time, leading to unnecessary iterations and increasing the risk of falling into a local minimum. Fig. 11 shows F1-Score over various number of hidden layer nodes. This figure shows that 500 is the best choice for Hidden Size for Sequential BINER and 400 for Basic and parallel BINER. Basic BINER implementation shows that if we increase the Hidden size to be more than 700, the model accuracy will drop to a 40% F1-Score. The parallelization begins to decline after 400. The sequential architecture is affected more than other architectures because this implementation has different loss functions; a connection between layers makes the error propagation faster and merging many hidden layers for this architecture requires more than 100 epochs, which means more computational time.

4.3.5. Batch Size

Batch size determines how many samples in each epoch we will feed to the neural network. Choosing the best number for this hyper-parameter may affect the model's performance in two ways. First, the largest possible batch size will affect the model's running time, which will have an effect on memory limitations. However, the model will end up training faster. Second, choosing a more significant batch size will send more data to the model, and each epoch will perform more calculations, which might decrease the generality of the learning. Therefore, choosing the correct batch size plays a significant role in the network. We select different batch size ranges and trained the model based on them to see the effect on F1-Score. Fig. 12 shows the relationship between batch size and F1-Score for each lookup layer on the NCBI-Disease dataset.

We obtain similar hyper-parameters when using the different datasets. For this reason, we only present the results evaluated for the NBCI-Disease dataset. Finally, Table 2 shows the selected hyper-parameters to train every three implementations of the BINER approach.

5. Results and Analysis

In this section, experiments are conducted to test the proposed approach on its effectiveness in improving the sensitivity and specificity of NER for medical context and compare the results to the existing methods. We also conduct performance tests to determine how well the existing and proposed methods perform when using transfer learning.

5.1. Model results for Medical Publication

In this section, we report our experimental results for our NER model over publication corpus. Each implementation is compared with three state-of-the-art techniques, BioBERT [9] and BLSTM-CNN-CRF [14], and DtranNer [12]. All the state-of-the-art techniques that we compared with our proposed model rely on CRF as a top layer except for BioBERT. F1-Score (F1), Precision (P), and Recall (R) are reported to compare the three implementations of BINER. Table 3 shows the Base-BINER results; our implementation outperforms the others in two out of six selected datasets. Base-BINER used two separate CRF layers as top layer and shows that if a dataset is designed for one purpose, e.g., BC2GM focused on gene and NCBI-DISEASE, then only the disease datasets perform better. Table 4 demonstrates that our implementation out performs the state-of-the-art algorithms in five databases. The table shows 9% improvement in detecting genes (BC2GM), 4% enhancement in detecting a combination of Disease and chemicals (BC5CDR), and an improvement of 1% in a large dataset that contains more than 40,000 entities (JNLPBA). Our architecture targeted those entities that combined two or more word together as a entity. The ability to detect these entities resulted from learning word segmentation and labeling separately. Table 5 shows that Sequence-BINER performed similar to Base-BINER, while we saw an improvement in three datasets out of six using our implementation. In all of the implementations, we saw that RNN embedding layer shows more improvement compare to other embedding layers. We propose using RNN character level embedding with the parallel BINER implementation as the best model. We summarize the reasons for our recommended implementation in Fig. 13 We conclude that parallel BINER

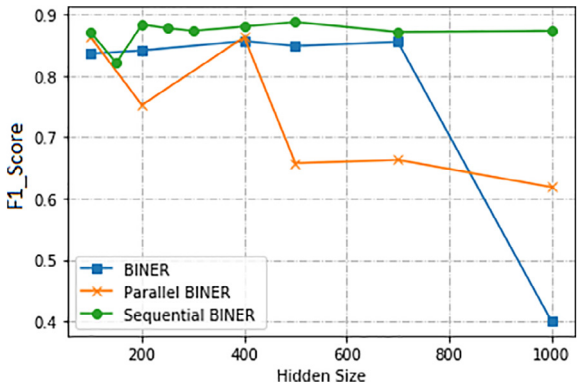

Fig. 11. Relationship between Hidden Size and F1-Score for lookup layer.

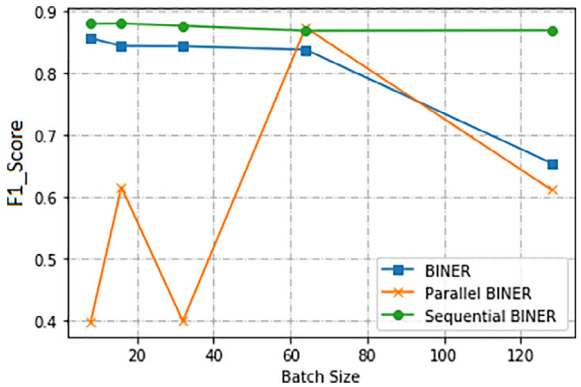

Fig. 12. Relationship between Batch Size and F1-Score for lookup layer on NCBI-Disease dataset.

Table 3
Performance value in terms of Precision, Recall and F1-Score for the proposed Base-BINER Implementation using different word-embedding techniques and state-of-the-art methods.

Corpus	BC2GM			BC5CDR			Linnaeus			NCBI-DISEASE			JNLPBA			BioNLP13PC		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Base-BINER-RNN	**0.92**	**0.89**	**0.89**	0.53	0.33	0.41	0.9	0.52	0.54	**0.94**	**0.91**	**0.91**	0.86	0.87	0.86	**0.92**	**0.9**	**0.9**
Base-BINER-CNN	0.91	0.86	0.87	0.85	0.77	0.77	**0.97**	0.58	0.64	0.45	0.5	0.47	0.84	0.85	0.84	0.9	0.88	0.88
Base-BINER-Lookup	0.91	0.86	0.87	0.9	0.76	0.81	0.9	0.76	0.82	0.93	0.9	0.9	0.85	0.83	0.83	0.86	0.89	0.86
Base-BINER-SAE	0.44	0.5	0.47	0.62	0.33	0.31	0.49	0.5	0.49	0.45	0.5	0.47	0.83	0.86	0.83	0.92	0.87	0.88
DTranNER(2020)	0.84	0.84	0.84	0.89	0.9	0.9	-	-	-	-	-	-	**0.88**	**0.89**	**0.88**	-	-	-
BLSTM-CNNs-CRF (2018)	0.87	0.81	0.82	0.79	0.73	0.73	0.9	0.51	0.59	0.9	0.85	0.85	0.72	0.75	0.72	0.71	0.59	0.63
BioBERT (2019)	0.81	0.81	0.81	**0.89**	**0.83**	**0.86**	0.92	**0.94**	**0.93**	0.88	0.89	0.88	0.74	0.83	0.78	-	-	-

performance is better or equivalent in almost all the cases unless using the Linnaeus dataset, where the BioBERT model performed best. The Linnaeus dataset specifically focused on species, though it is a relatively smaller dataset than the other five datasets. When we apply our approach, we achieved a 0.85 F1-Score which is smaller than the BioBERT F1-score.

Table 6 shows the results for Base-, Sequence-, and Parallel-BINER reporting P-value for Precision, Recall and F1-Score compare to BioBERT results. The statistical significance is 0.05 ($p < 0.05$), thus the results of the T-Test according to reported p-values shows there is sufficient evidence to make sure the result of classification compare to BioBERT have same mean, therefore any observed difference in model results is likely due to a difference in the models architects and hyper-parameters tuning.

Table 4
Performance value in terms of Precision, Recall and F1-Score for the proposed Parallel-BINER Implementation using different word-embedding techniques and state-of-the-art methods.

Corpus	BC2GM			BC5CDR			Linnaeus			NCBI-DISEASE			JNLPBA			BioNLP13PC		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Parallel-BINER-RNN	**0.92**	**0.9**	**0.9**	**0.91**	**0.91**	**0.91**	**0.92**	0.79	0.85	**0.95**	**0.91**	**0.91**	**0.88**	**0.90**	**0.89**	**0.92**	**0.89**	**0.9**
Parallel-BINER-CNN	0.91	0.87	0.87	0.87	0.79	0.8	0.97	0.58	0.64	0.94	0.9	0.9	0.85	0.86	0.85	0.91	0.88	0.89
Parallel-BINER-Lookup	0.86	0.89	0.84	0.89	0.77	0.81	0.93	0.76	0.83	0.94	0.89	0.91	0.83	0.84	0.83	0.89	0.87	0.87
Parallel-BINER-SAE	0.9	0.89	0.88	0.92	0.77	0.82	0.92	0.72	0.79	0.94	0.87	0.89	0.85	0.83	0.83	0.91	0.84	0.86
DTranNER	0.84	0.84	0.84	0.89	0.9	0.9	-	-	-	-	-	-	0.88	0.89	0.88	-	-	-
BLSTM-CNNs-CRF	0.87	0.81	0.82	0.79	0.73	0.73	0.9	0.51	0.59	0.9	0.85	0.85	0.72	0.75	0.72	0.71	0.59	0.63
BioBERT	0.81	0.81	0.81	0.89	0.83	0.86	0.92	**0.94**	**0.93**	0.88	0.89	0.88	0.74	0.83	0.78	-	-	-

Table 5
Performance value in terms of Precision, Recall and F1-Score for the proposed Sequence-BINER Implementation using different word-embedding techniques and state-of-the-art methods.

Corpus	BC2GM			BC5CDR			Linnaeus			NCBI-DISEASE			JNLPBA			BioNLP13PC		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Sequence-BINER-RNN	**0.92**	**0.87**	**0.89**	0.55	0.33	0.41	0.74	0.5	0.49	**0.94**	**0.91**	**0.92**	0.86	0.88	0.87	**0.93**	**0.9**	**0.91**
Sequence-BINER-CNN	0.9	0.84	0.86	0.87	0.79	0.81	0.92	0.74	0.82	0.94	0.89	0.9	0.84	0.86	0.85	0.9	0.86	0.87
Sequence-BINER-Lookup	0.9	0.87	0.87	0.89	0.75	0.81	0.85	0.78	0.81	0.94	0.88	0.9	0.85	0.84	0.84	0.9	0.85	0.86
Sequence-BINER-SAE	0.84	0.89	0.86	0.8	0.81	0.77	0.96	0.59	0.65	0.52	0.5	0.47	0.78	0.86	0.81	0.91	0.86	0.87
DTranNER	0.84	0.84	0.84	0.89	0.9	0.9	-	-	-	-	-	-	**0.88**	**0.89**	**0.88**	-	-	-
BLSTM-CNNs-CRF	0.87	0.81	0.82	0.79	0.73	0.73	0.9	0.51	0.59	0.9	0.85	0.85	0.72	0.75	0.72	0.71	0.59	0.63
BioBERT	0.81	0.81	0.81	**0.89**	**0.83**	**0.86**	**0.92**	**0.94**	**0.93**	0.88	0.89	0.88	0.74	0.83	0.78	-	-	-

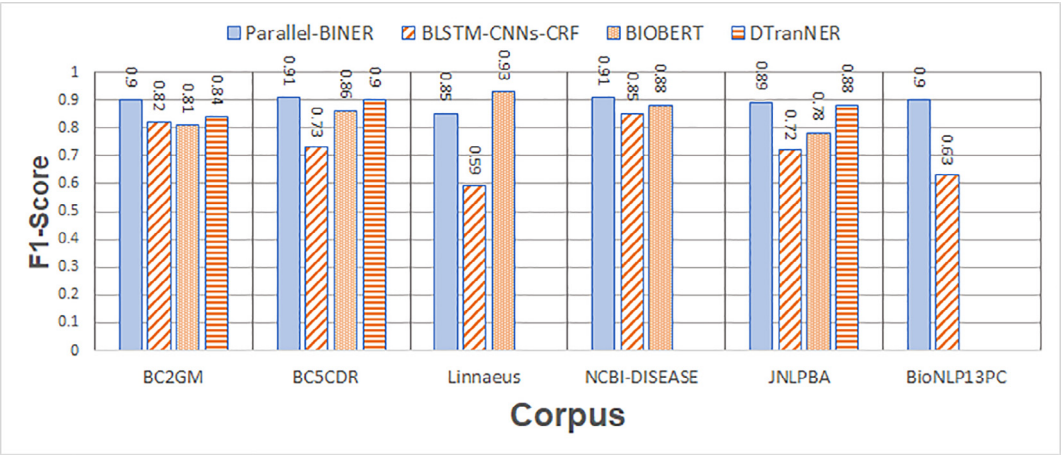

Fig. 13. Comparison of Parallel BINER model with BioBERT, BLSTM-CNN-CRF and DTranNER.

5.2. Model results for Clinical Note

Many pre-trained models are created by large datasets and powerful machines, such as ELMO, BERT, GPT3, which few companies or research centers can build, maintain, and deploy. On top of accessing enormous resources, the cost of data labeling and keeping the models updated to create a bottleneck for expanding the models in the real world is high. This case study shows the effect of shifting data in a single same domain, such as training data for detecting drug names on academic publications and using the same model to detect the drug names in clinical notes. We also study the effect of the data labeling size for training on deep learning architectures. Experiments are designed by bringing unseen data, e.g., a set of documents that are clinical notes have physician annotations for adverse drug effects. That suggested a focus on chemical and drug entities. The Adverse Drug Events (ADEs) [36] dataset proposed creating a set of medical documents that physicians can annotate with drug-related information, including drug names, dosage, strength, duration, frequency, form, and reason.

Table 6
P-Values with Confidence of 0.05 compare to Bio-BERT as base.

	Base-BINER				Sequence-BINER				Parallel-BINER			
	RNN	CNN	Lookup	SAE	RNN	CNN	Lookup	SAE	RNN	CNN	Lookup	SAE
Precision	0.0016	0.0033	0.008	0.0099	0.0042	0.0075	0.0067	0.0085	0.0089	0.0083	0.0069	0.0085
Recall	0.0076	0.0091	0.0067	0.007	0.0078	0.0065	0.0066	0.0085	0.0096	0.0063	0.0055	0.007
F1-Score	0.0074	0.0087	0.0014	0.0099	0.0073	0.001	0.0014	0.009	0.0064	0.0052	0.0019	0.0023

This dataset detects the medicine and creates two types of relationships: drugs used for specific symptoms and disease and drugs with adverse events. We select one state-of-the-art technique named BERT, which is a model trained on PubMed and Mimic dataset [88]. To compare with our proposed BINER model, in the first step, we use both models without tuning or retraining to detect the drug names. The result shows that both model need retraining and tuning again to perform on the selected dataset. Retraining the model might sound easy, however, preparing and annotating new data is a time-consuming and expensive process. We design an experiment to study the training size effect in transfer learning. Fig. 14 shows how we split the dataset into separate sections for Training and Testing. We randomly select 20% for the Test group and 80% for the Train group in the first split. In the second split, we create ten small datasets from the 80% training set. We randomly select 10% of the training set and incrementally add it to the dataset. The first dataset contains 10%, the second dataset 20% until the 10th dataset, representing the 80% training set. In this experiment, we consider two factors. First, we consider the size of the training set (ten datasets). Second, we consider the number of epochs (10,20, and 30 Epochs). The BERT model is the state-of-art model selected to compare with all models. We compare Parallel BINER with BERT and the results show that BERT performs better. On the other hand, the BINER Sequence shows better accuracy compared to BERT. The F1-Score for each experiment based on the training data size is documented; the x-axis shows the size of the training dataset. In the last experiment, we selected Sequential and Parallel implementation as superb models, so we re-trained these two models with 10, 20, and 30 epochs. The BERT model was also chosen as the state-of-art model to com-

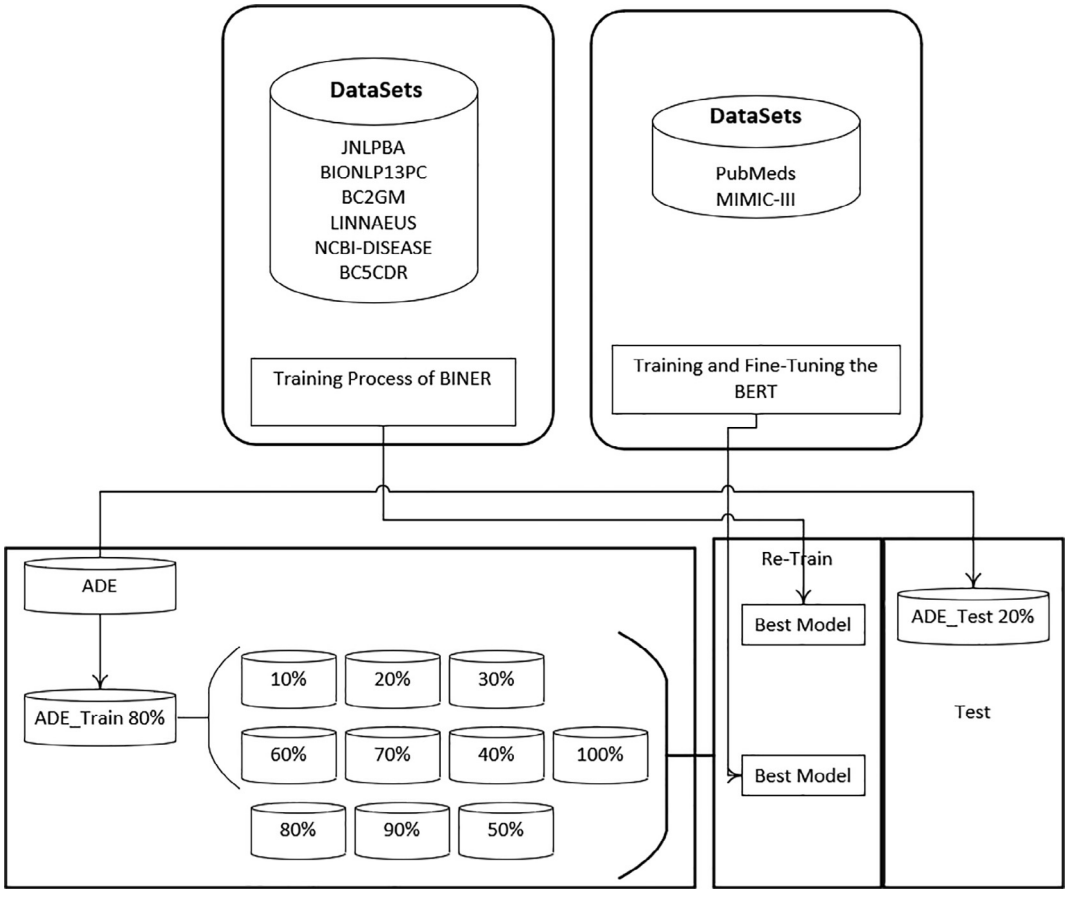

Fig. 14. Shows the Process of Retraining the Models for the ADE dataset.

pare with our models. The BERT model is appropriate because it was trained on MIMIC and Pubmed datasets. Fig. 15 shows that the BERT model performs better than the BINER parallel implementation. On the other hand, Fig. 16 shows that the BINER Sequential implementation has better performance than BERT. If we retrain the model with ten epochs, it will have similar result as BERT, and if we retrain with 20 or 30 epochs, it outperforms the BERT model.

5.3. Summary of Analysis

Performance measures for the different BINER implementations, BioBERT, and DTraNER included precision, recall and macro f1-score defined in Eq. 8. Because we performed multiple comparisons in this study to assess the models performance against state-of-the-art techniques, we report the p-value.

For publication datasets, we compared the statistical metrics for multiple state-of-the-art models. We demonstrated, via a bar chart (Fig. 13), that we outperformed BioBert in four out of the five datasets, with an improved accuracy of 2.8% on average. The BIONLP13PC was a different dataset that did not report using the BioBert model; we choose this dataset because of the diversity of entities that cover many new hot topics in biology close to biologists' needs. The DTraNER model uses this dataset, and our model exceeds this model by 27% improvement.

We observe how error propagation and layers are linked and how the combination will affect the accuracy and generality of a model. To challenge our models, we add a clinical note dataset, which are structured differently than academic publications. As a physician generates these notes while they interview patients, the possibility of errors, like misspellings, or the high number of abbreviations makes this dataset more challenging and noisy. In biomedical publication analysis, we conclude that Parallel and Sequential BINER are the best implementations for these datasets; with further analysis, using the sequential and parallel architecture for detecting drug names in clinical notes, the sequential model outperforms BERT. We observed two factors in sequential implementation that make this model more general. The first factor is backpropagation in the Segmentation layer (where we learn about the IOB structure of the sentences) and entity detection (where we detect NER). The second factor is propagating the error of the segmenting layer to the entity layer. These factors help the model to find a correlation between segmenting the words and labeling them.

The first experiment shows that we can use a lower number of parameters in the neural network while improving or maintaining the quality and accuracy of the model. The second experiment also studies how many annotations we need

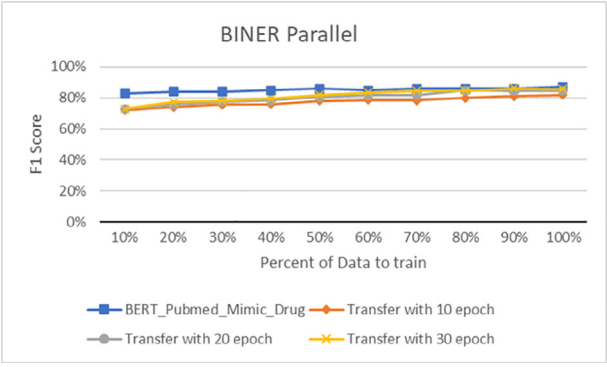

Fig. 15. Comparison between BINER Parallel and BERT on ADE dataset.

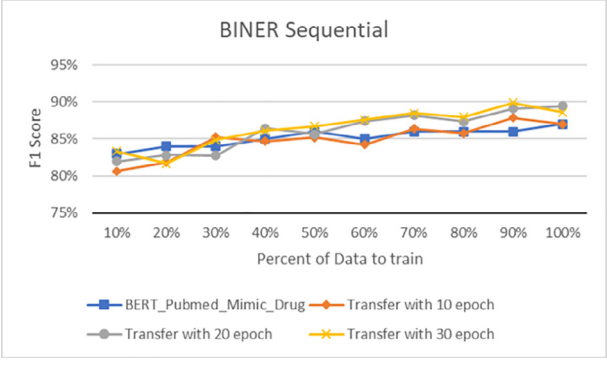

Fig. 16. Comparison between BINER Sequential and BERT on ADE dataset.

to have an acceptable accuracy range. Fig. 16 shows that using 50% of data and training for 20 epochs, we can surpass the BERT model by an average 4% improvement. As we expected, the trend shows that more annotation will improve the accuracy and this increased accuracy needs more resources and time that what is normally used in healthcare, since the method is so expensive and limited.

6. Conclusion

The NER task is a foundation for advanced text analysis. In domain-specific applications such as medical text, context plays a significant role in improving performance. Implementing applicable and accessible models is another critical feature for creating domain-specific models. We developed a practical and reliable Biomedical NER model named BINER. In this approach, we implemented and evaluated three architectures. These were designed by combining different Deep Learning layers to assess biomedical text sources. The best performance was achieved by comparing the implementations via parallel BINER with RNN character level embedding. Our proposed model has achieved promising results for Named Entity Recognition in biomedical text. The model can be expanded, via additional training, to include other application domains.

Based on the experimental results, we successfully responded to the challenges that we identified. Our contribution can be summarized as follows:

1. By experimenting with different Embedding Layers combined with Bidirectional LSTM and CRF to do NER in the Biomedical domain, we show different embedding layer and architecture effects. We demonstrated and tested different architectures and identified a Parallel BINER model that outperforms other architectures. Our proposed model requires fewer layers than Bert and BioBERT. The model has 2 major layers, each of them has 10 Layers with a Hidden size of 400 (32 Million Total Parameters). This model is trained in 7 days using NVIDIA 2080 GTX;
2. Our results compare favorably with BioBert when tested on five different standard datasets. We improved the F1-Score over BioBert by 3% on Disease detection, 8% on detecting Protein (Contains DNA, RNA, and Cell), and 9% for Gene detection. All the code and experiments are available in our Github at https://github.com/moasgh/BumbleBee. The models are available to be tested on this website, http://catanas.org/BINER.

This model can be used and implemented in hospitals and research centers to detect disease, drug names, genes, and species. The retrained model can be used for clinical notes and biomedical literature. We deployed this model at the University of Louisville hospital to unlock the power of unstructured data; this model analyzed more than 100,000 radiology and pathology notes.

The BINER focused on NER biomedical literature and clinical notes; in contrast to other models, that they performing entity relation extraction and biomedical question answering. We create a foundation and train the model using three different biomedical sources, medical literature, physician notes, and clinical notes. In future work, we will enhance and expand the model for other tasks as well.

CRediT authorship contribution statement

Mohsen Asghari: Conceptualization, Methodology, Software, Data curation, Writing - original draft. **Daniel Sierra-Sosa:** Methodology, Software, Investigation, Writing - review & editing. **Adel S. Elmaghraby:** Supervision, Validation, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S.-J. Yen, Y.-C. Wu, J.-C. Yang, Y.-S. Lee, C.-J. Lee, J.-J. Liu, A support vector machine-based context-ranking model for question answering, Information Sciences 224 (2013) 77–87.
[2] M. Asghari, D. Sierra-Sosa, A. Elmaghraby, Trends on health in social media: Analysis using twitter topic modeling, in: 2018 IEEE international symposium on signal processing and information technology (ISSPIT), IEEE, 2018, pp. 558–563.
[3] M. Asghari, D. Sierra-Sosa, A.S. Elmaghraby, A topic modeling framework for spatio-temporal information management, Information Processing & Management 57 (6) (2020) 102340.
[4] N. Vanetik, M. Litvak, E. Churkin, M. Last, An unsupervised constrained optimization approach to compressive summarization, Information Sciences 509 (2020) 22–35.
[5] A. Passos, V. Kumar, A. McCallum, Lexicon infused phrase embeddings for named entity resolution, arXiv preprint arXiv:1404.5367..
[6] L. Ratinov, D. Roth, Design challenges and misconceptions in named entity recognition, in: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), 2009, pp. 147–155.
[7] H. Fei, Y. Ren, D. Ji, Dispatched attention with multi-task learning for nested mention recognition, Information Sciences 513 (2020) 241–251.
[8] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805..

[9] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C.H. So, J. Kang, Biobert: pre-trained biomedical language representation model for biomedical text mining, arXiv preprint arXiv:1901.08746..
[10] G. Crichton, S. Pyysalo, B. Chiu, A. Korhonen, A neural network multi-task learning approach to biomedical named entity recognition, BMC bioinformatics 18 (1) (2017) 1–14.
[11] X. Wang, Y. Zhang, X. Ren, Y. Zhang, M. Zitnik, J. Shang, C. Langlotz, J. Han, Cross-type biomedical named entity recognition with deep multi-task learning, Bioinformatics 35 (10) (2019) 1745–1752.
[12] S. Hong, J.-G. Lee, Dtranner: biomedical named entity recognition with deep learning-based label-label transition model, BMC bioinformatics 21 (1) (2020) 1–11.
[13] A. Marasović, A. Frank, Srl4orl: Improving opinion role labeling using multi-task learning with semantic role labeling, arXiv preprint arXiv:1711.00768..
[14] X. Ma, E. Hovy, End-to-end sequence labeling via bi-directional lstm-cnns-crf, arXiv preprint arXiv:1603.01354..
[15] J.P. Chiu, E. Nichols, Named entity recognition with bidirectional lstm-cnns, Transactions of the Association for, Computational Linguistics 4 (2016) 357–370.
[16] C.D. Santos, B. Zadrozny, Learning character-level representations for part-of-speech tagging, in: Proceedings of the 31st international conference on machine learning (ICML-14), 2014, pp. 1818–1826.
[17] H. Huang, H. Wang, D. Jin, A low-cost named entity recognition research based on active learning, Scientific Programming (2018).
[18] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, S. Khudanpur, Recurrent neural network based language model, in: Eleventh annual conference of the international speech communication association, 2010.
[19] W. Zaremba, I. Sutskever, O. Vinyals, Recurrent neural network regularization, arXiv preprint arXiv:1409.2329..
[20] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
[21] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781..
[22] N. Kitaev, D. Klein, Constituency parsing with a self-attentive encoder, arXiv preprint arXiv:1805.01052..
[23] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, arXiv preprint arXiv:1802.05365..
[24] J. Lafferty, A. McCallum, F.C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data..
[25] A. McCallum, D. Freitag, F.C. Pereira, Maximum entropy markov models for information extraction and segmentation., in: Icml, Vol. 17, 2000, pp. 591–598..
[26] A. Ratnaparkhi, A maximum entropy model for part-of-speech tagging, in: Conference on Empirical Methods in Natural Language Processing, 1996.
[27] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, arXiv preprint arXiv:1508.01991..
[28] J.-D. Kim, T. Ohta, Y. Tsuruoka, Y. Tateisi, N. Collier, Introduction to the bio-entity recognition task at jnlpba, in: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications, Citeseer, 2004, pp. 70–75..
[29] L. Smith, L.K. Tanabe, R.J. nee Ando, C.-J. Kuo, I.-F. Chung, C.-N. Hsu, Y.-S. Lin, R. Klinger, C.M. Friedrich, K. Ganchev, et al., Overview of biocreative ii gene mention recognition, Genome biology 9 (2) (2008) S2..
[30] R. Leaman, G. Gonzalez, Banner: an executable survey of advances in biomedical named entity recognition, in: Biocomputing 2008, World Scientific, 2008, pp. 652–663..
[31] D. Campos, S. Matos, J.L. Oliveira, Gimli: open source and high-performance biomedical name recognition, BMC bioinformatics 14 (1) (2013) 54.
[32] H. Mi, P. Thomas, Panther pathway: an ontology-based pathway database coupled with data analysis tools, in: Protein Networks and Pathway Analysis, Springer, 2009, pp. 123–140..
[33] M. Gerner, G. Nenadic, C.M. Bergman, Linnaeus: a species name identification system for biomedical literature, BMC bioinformatics 11 (1) (2010) 85.
[34] R.I. Doğan, R. Leaman, Z. Lu, Ncbi disease corpus: a resource for disease name recognition and concept normalization, Journal of biomedical informatics 47 (2014) 1–10.
[35] J. Li, Y. Sun, R.J. Johnson, D. Sciaky, C.-H. Wei, R. Leaman, A.P. Davis, C.J. Mattingly, T.C. Wiegers, Z. Lu, Biocreative v cdr task corpus: a resource for chemical disease relation extraction, Database (2016).
[36] K. Buchan, Annotation guidelines for the adverse drug event (ade) and medication extraction challenge, n2c2, US..
[37] J. Opitz, S. Burst, Macro f1 and macro f1, arXiv preprint arXiv:1911.03347..