Revue d'Intelligence Artificielle
Vol. 37, No. 1, February, 2023, pp. 239-247
Journal homepage: http://iieta.org/journals/ria

NeRBERT- A Biomedical Named Entity Recognition Tagger

Manish Bali*, Anandaraj Shanthi Pichandi

Department of Computer Science and Engineering, Presidency University, Bengaluru 560064, India

Corresponding Author Email: balimanish0@gmail.com

https://doi.org/10.18280/ria.370130

Received: 21 January 2023
Accepted: 9 February 2023

Keywords:
Bi-directional LSTM, bidirectional encoder representations from transformers, deep learning, ensemble model, graphical processing unit, machine learning, named entity recognition

ABSTRACT

Biomedical named entity recognition is a popular research topic in the Biosciences domain as number of biomedical articles getting published are increasing rapidly. Generic models using machine learning and deep learning techniques have been proposed for extracting these entities in the past, however there is no clear verdict on which techniques are better and how these generic models perform in a domain-specific big data scenario. In this paper, we evaluate three baseline models using the most complex BioNLP 2013 cancer genetics dataset addressing the cancer domain. A classifier ensemble, bidirectional long short-term memory (Bi-LSTM) model and the bidirectional encoder representations from transformers (BERT) model are implemented. We propose NeRBERT, a domain-specific, graphical processing unit (GPU) pre-trained language model using extra biomedical corpora extending BERT$_{\text{BASE}}$. Experimental results prove the efficacy of NeRBERT as it outperforms the other three models with an F1-score gain of 12.18 pp, 8.59 pp and 5.43 pp over the ensemble, Bi-LSTM and BERT models respectively. GPUs reduce the model training time to less than half. Comparing it to existing state-of-the-art models, it performs 1.57 pp higher than the next best existing model compared, emerging as a robust biomedical and cancer phenotyping NER tagger.

1. INTRODUCTION

Given the exponential growth of documents, particularly in the biomedical field, there is a critical need for techniques that can aid in the automatic extraction of reliable biological information. Identification of biomedical entities such as proteins, cells, Deoxyribonucleic acid (DNA), Ribonucleic acid (RNA), and many more is part of this process. The goal of generic named entity recognition (NER) is to locate a word or phrase that correlates to a specific occurrence, such as a person, location, organization, or any other variable. The extraction and classification of biological named entities in a corpus with pre-defined entity tags is the primary focus of the Biomedical NER task. *BIO* tagging is commonly used in the corpus to represent the beginning (*B*), inside (*I*), and outside (*O*) or non-entity tokens.

Even though BioNER is a prominent topic of research, the best models in literature for classifying biomedical named entities and general text still have a significant (~10%) performance gap between them. The following factors are the main causes of this task's difficulty for bio-medical entities:

(1) The creation of new biomedical named entities is happening rapidly, and a complete lexicon for these entities is missing;

(2) Biomedical entities comprise words or phrases that, despite being the same, can mean contextually different entities. Additionally, a lot of terms are with different spellings;

(3) Some entities are also far too lengthy with some modifiers used before any basic named entity. It is therefore very difficult to determine where their boundaries lie;

(4) Biomedical designated entities may also be embedded into one another. Because of this, it takes a lot of effort to identify these entities;

(5) It is also noted that abbreviations are widely employed in this field. The challenge of classifying them becomes that much more challenging because there is no proof that such abbreviations exist.

Therefore, directly applying the advancements in NLP to generic NER models returns unsatisfactory results. There exists a need to develop disease-specific efficient models to handle complex biomedical entities in that domain. Hence, the key contributions of this paper towards a robust, domain-specific biomedical NER tagger are:

(1) NeRBERT, an optimized cancer phenotyping NER tagger developed by extensive pre-training using extended biomedical corpora on the most complex among available (and accessible) BioNLP datasets, the BioNLP2013 cancer genetics dataset addressing the cancer domain. In intra-model comparison, it outperforms the baseline models with an F1-score gain of 12.18 pp, 8.59 pp and 5.43 pp over the ensemble, Bi-LSTM and BERT$_{\text{BASE}}$ models respectively. In inter-model comparison with existing state-of-the-art models, it performs 1.57 pp better than the next best existing model compared.

Optimizing NeRBERT pre-training leveraging GPU computing. Since training such a complex model with large corpora is extremely time consuming, we leverage the many-core architecture of the 8 x NVidia V100 (32GB) GPUs and fit all mini-batch training into its large memory. With this, the time taken to pre-train NeRBERT is reduced to less than half (~11 days) compared to training it on an equivalent 8-socket CPU-only system.

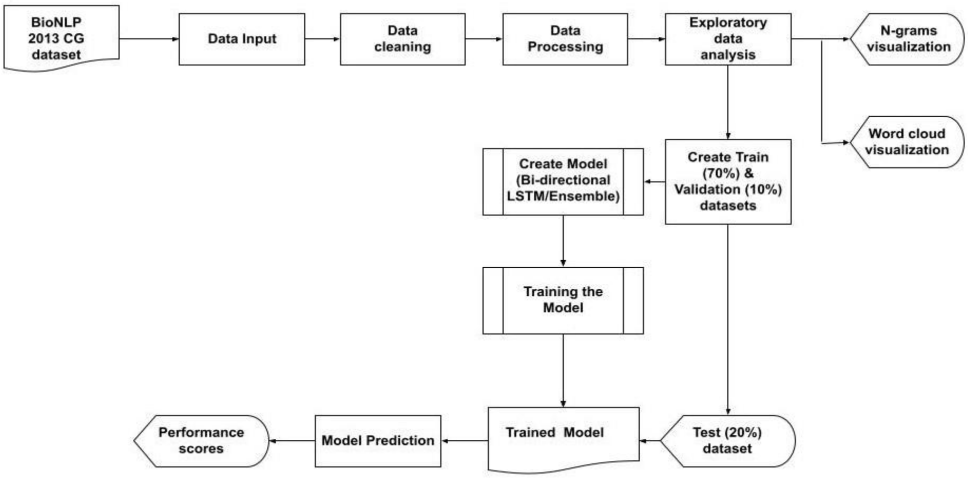


Figure 1. Process flow in baseline models

The manuscript is organized as follows: Section 1 presents an introduction to the topic of named entity recognition and the challenges in biomedical entity recognition. In Section 2, existing methods in biomedical named entity recognition are discussed under related works. Section 3 explains the materials and methods proposed for named entity recognition. Results and discussion are detailed in section 4. Finally, Section 5 draws conclusions and ends with references.

2. RELATED WORKS

In literature, NER techniques have been categorized as rule-based heuristic, dictionary-based, and statistical ML techniques. However, it has been found that these techniques do not produce desired results. Additionally, they fall short in terms of entity representation coverage with sufficiently rich features, and no one algorithm/technique can achieve higher performance. In recent years, classifier ensemble [1-5], which is an amalgamation of various base classifiers has emerged as a viable machine learning technique. Instead of individual base classifier performance, it functions as an aggregator, combining the results of base classifiers to address any classifier weaknesses and provide a comprehensively well-balanced performance.

Deep Learning algorithms for NER tasks, particularly Long Short-Term Memory and Bi-directional LSTM [6-10] have demonstrated leadership since they have significantly outperformed current ML techniques. Deep learning, as opposed to feature-based techniques that are curated by humans, can automatically identify hidden traits from unlabeled data. Word embeddings, which represent word meanings in *n*-dimensional space, are learnt using these unlabeled data. One significant advantage of these approaches is the ability to design training algorithms that forego task-specific engineering in favor of relying on large, unlabeled datasets to uncover internal word representations useful in the overall NER objective.

From the time Google released the BERT model, researchers have been able to determine how effective this strategy is for downstream NLP tasks, such as NER [11]. By utilizing a masked model that pretrains on a sizable amount of textual unstructured data in an unsupervised or self-supervised way, this language model increases the impact of the fine-tuning technique. Despite having a multilanguage model, researchers found that BERT works better when it has been pretrained on domain-specific language [12-16]. The results of the fine-tuning performed better on several NLP tasks, including sentiment analysis, NER, and question-answering, compared to the initial BERT multilanguage model. However, to get the model to understand biomedical content better and with an eye on improving performance, the developers of BioBERT [17] pretrained it using the original BERT data and additional biomedical corpora. This helped to increase model's performance for BioNLP workloads tested. Although developers of BioBERT have pre-trained the model on biomedical corpora and tested few task-specific datasets, the most complex BioNLP 2013 Cancer Genetics (CG) dataset covering multiple sub-domains of cancer biology was not evaluated and there are very few references to develop and evaluate any cancer-specific NER taggers in literature.

3. MATERIALS AND METHODS

The choice of NER models selected in this paper are based on literature study. A classifier ensemble may be used to improve classification performance, as is discussed in Section 2. The classification performance can also be improved by employing a Bi-LSTM network. It is required to recall the long-range reliance of the entity from the first instance to the next instance when the sentences are complex and lengthy, for instance, "*TGF-beta mediates RUNX induction and FOXP3 is efficiently up-regulated by RUNX1 and RUNX3"*, where neural networks are required. And BERT has emerged as a choice in many NLP areas as it uses a masked model that predicts randomly masked words in a sequence, and hence can be used for learning bidirectional representations.

Figure 1 describes a high-level process flow for the baseline models considered. We briefly discuss these models and NeRBERT, which is same in structure as $\text{BERT}_{\text{BASE}}$, and then its pre-training and fine-tuning process.

3.1 Ensemble model

Machine Learning algorithms perform poorly in standalone mode whereas ensemble techniques have performed well [2]. Figure 2 illustrates the architecture of the ensemble model used. Five base classifiers with word2vec CBOW embedding [18] are used both in standalone and ensemble mode which are briefly explained below.

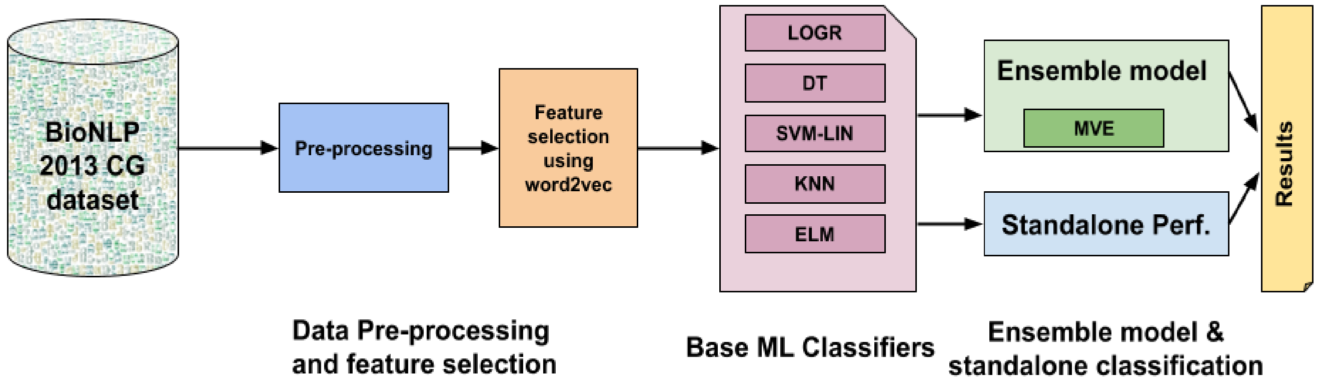

BioNLP 2013 CG dataset
Pre-processing
Feature selection using word2vec
LOGR
DT
SVM-LIN
KNN
ELM
Ensemble model
MVE
Standalone Perf.
Results
Data Pre-processing and feature selection
Base ML Classifiers
Ensemble model & standalone classification

Figure 2. Proposed architecture of the ensemble model

(1) Logistic regression (LOGR): To achieve the categorization of the dependent variable, regression is performed to the independent and dependent variables. In this work, LOGR defines a prediction method that verifies the named entity's semantic correspondence to the context. Mathematically, it is shown as in Eq. (1).

$$logit[\pi(x)] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots \ldots . + \beta_m X_m \quad (1)$$

LOGR returns π(x) as in Eq. (2):

$$\pi(x) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots\ldots+\beta_m X_m}}{1 + e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots\ldots+\beta_m X_m}} \quad (2)$$

(2) Support Vector Machine (SVM): The pattern in the data is used by SVM, which serves as a non-probabilistic binary linear classifier. SVM utilizes the function in Eq. (3) to predict.

$$Y' = w * \phi(x) + b \quad (3)$$

In Eq. (3), Y' is retrieved by reducing the risk of regression as in Eq. (4).

$$R_{reg}(Y') = C * \sum_{i=0}^{l} \gamma(Y_i' - Y_i) + \frac{1}{2} * \|w\|^2 \quad (4)$$

Here,

$$w = \sum_{j=1}^{l} (\alpha_j - \alpha_j^*) \phi(x_j) \quad (5)$$

In Eq. (5), the parameters α and α^* state the relaxation parameter called Lagrange multiplier. The output obtained is,

$$Y' = \sum_{j=1}^{l} (\alpha_j - \alpha_j^*) \phi(x_j) * \phi(x) + b \quad (6)$$

$$Y' = \sum_{j=1}^{l} (\alpha_j - \alpha_j^*) * K(x_j, x) + b \quad (7)$$

In Eq. (6) and Eq. (7), $K(x_j, x)$ states the kernel function. In our study, SVM algorithm with Linear kernel functions is used.

(3) Extreme Learning Machine (ELM): One of the main problems with some models is local minima and convergence, which gets worse with large training datasets and reduces overall learning and classification efficiency. ELM with a linear kernel function is suggested as the base classifier for NE classification to address this. The proposed ELM base classifier's output is defined as in Eq. (8).

$$y(t + k) = f(X) = \sum_{i=1}^{L} \beta_i G(a_i, b_i, X) \quad (8)$$

(4) Decision Tree (DT): In this study, C5.0 decision tree classifier model is applied that performs recursive partitioning over the dataset to predict named entity for the input. Originating at the root node, each node of the tree splits the feature vector into different branches based on association rule between the split criteria.

(5) K-Nearest neighbors (KNN): Since the full data set is used to categorize new data, K-nearest neighbor is employed to solve a classification problem. This approach does not require any training data. It determines the separation between a new data point and each existing point in the dataset when one is provided. Then, based on the *K* value, it determines how many nearest neighbors there are in the data set. If *K=1*, it calculates the minimal distance between each point and categorizes them all as belonging to the same class. If *K* is greater than 1, a list of *K* minimum distances between each data point is used.

The ensemble model uses majority voting to arrive at the result. In majority voting, the class label that represents the majority (mode) of the class labels predicted by each individual classifier is the class label that is projected for a specific sample.

3.2 Bi-directional LSTM model

The concept of a bidirectional RNN, which operates in both forward and reverse directions and has separate hidden layers for each direction, is the foundation of the bidirectional LSTM. As depicted in Figure 3, these hidden layers are connected to a common output layer. We employ this model to assess its BioNER effectiveness in comparison to other models.

For all models, the dataset is split into train set (70%), validation (10%) and test set (20%). The holdout evaluation technique is used to assess how well the implemented model predicts. Also, we use word2vec continuous bag of words (CBOW) embedding. CBOW architecture [18] is used to learn word representations which predicts the middle word based on the terms in the surrounding context. As seen in Figure 4, the context consists of a few words before and after the present (middle) word.

3.3 BERT model

As shown in Figure 4, BERT is typically an encoder stack of the Transformer architecture. It's an embedding layer which receives a string of words as input and sends it up to the subsequent encoder unit in a way like the standard encoder in the transformer. It applies self-awareness to each encoder layer. The results are then disseminated via a feed-forward network. The output of the feedforward network is then passed on to the next encoder.

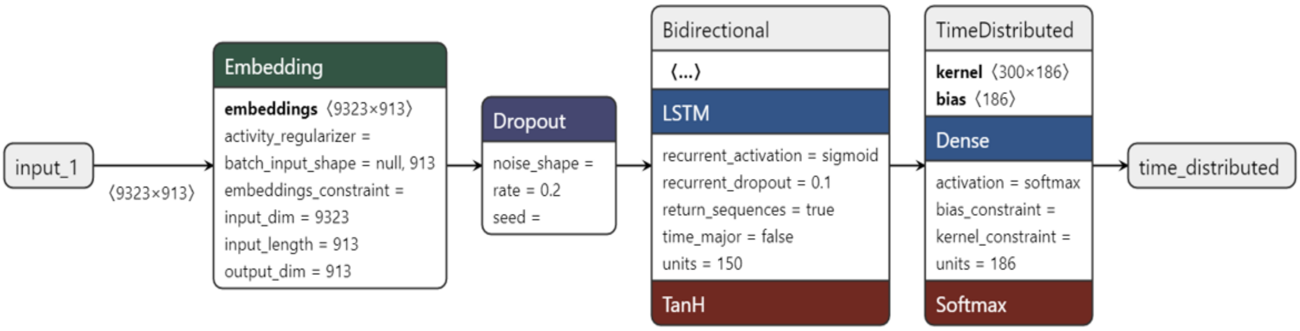


Figure 3. Proposed architecture of the Bi-directional LSTM model

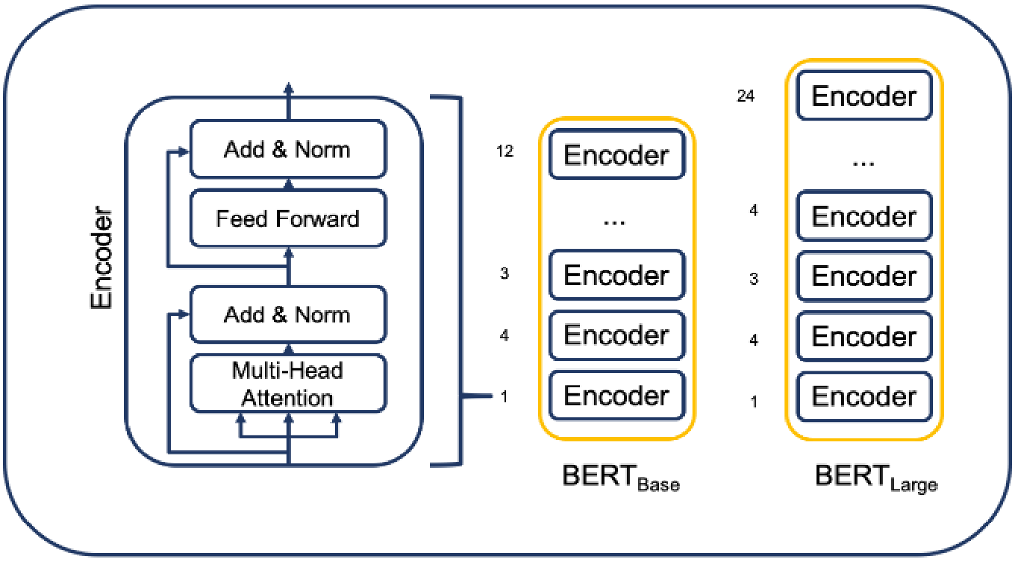


Figure 4. The BERT architecture

BERT uses a fine-tuning method for each activity that does not require a specific design. A pre-trained BERT model can be fine-tuned with just one additional layer to achieve cutting-edge performance. The data from the training set is used to fine-tune the BERT architecture. While organizing data, the required format is initially followed. Three input arrays are received by the BERT layers: *input_ids*, *attention_mask*, and *token_type ids*.

(1) *input_ids*: They are a list of integers that each have a unique connection to a word.

(2) *attention_mask*: The input IDs array are represented by this collection of 1s and 0s.

(3) *Token_type ids*: To categorize sequences or react to queries, this is employed. Since these need two distinct sequences to be kept in the same input IDs, special tokens like classifier [*CLS*] and separator [*SEP*] are used to separate the sequences.

The tokenizer class *encode_plus* function tokenizes the raw input, adds the special tokens, and pads the vector to the maximum length provided. To feed our raw data into the BERT model in the correct format, a helper function is used. The two variations, BERT$_{\text{BASE}}$ and BERT$_{\text{LARGE}}$ as seen have varying degrees of architecture complexity. The base model's encoder contains 12 layers and 110M parameters, whilst the large encoder version has 24 layers and 330M parameters.

3.4 NeRBERT model

In this study, we extend the BERT$_{\text{BASE}}$ model to propose NeRBERT. It is pre-trained using additional biomedical corpora like PubMed abstracts and PubMed Central articles and then fine-tune on task specific dataset which is the complex BioNLP 2013 CG dataset. Figure 5 provides an overview of the pre-training and fine-tuning of the NeRBERT model implemented.

3.4.1 Pre-training NeRBERT model

During the pre-training phase, the BERT model which is trained on general corpora (Wikipedia) will not be able to address complex domain-specific biomedical text like *TGF-beta, RUNX, FOXP3* etc. Therefore, BERT though good for general-purpose NLP tasks performs poorly for domain specific BioNER tasks. Hence, we pre-train it with extra biomedical corpora like PubMed abstracts and PubMed central articles. Large amounts of this unannotated corpora is used in pre-training in an unsupervised way.

Due to a risk of overfitting issues and longer training times associated with larger epoch numbers, we decided to limit the number of epochs to two. Since training NeRBERT model is extremely time consuming, we leverage GPUs for this task. Eight Nvidia V100 (32GB) GPUs are used and time to train is compared to an equivalent 8-socket CPU system

The NeRBERT fine-tuned weights and other hyper-parameters are shown in Figure 6. In brief, the model uses BERT$_{\text{BASE}}$, tokenizer used is BERT base-cased, fully connected classification layer with Adam optimizer, layers used-12 and hidden size is 768. We fine-tune the pre-trained model with task specific corpora for named entity recognition.

3.4.2 Fine-tuning NeRBERT model

For fine-tuning NeRBERT for named entity recognition task, an extra fully connected layer is added that further trains the model using biomedical-specific annotated corpora, the BioNLP 2013 CG dataset, using the pre-trained backbone weights. A single K40 GPU was used to fine-tune NeRBERT for this task.

We assess the performance of all models using standard relative performance parameters i.e., Accuracy, Precision, Recall and F-1 score.

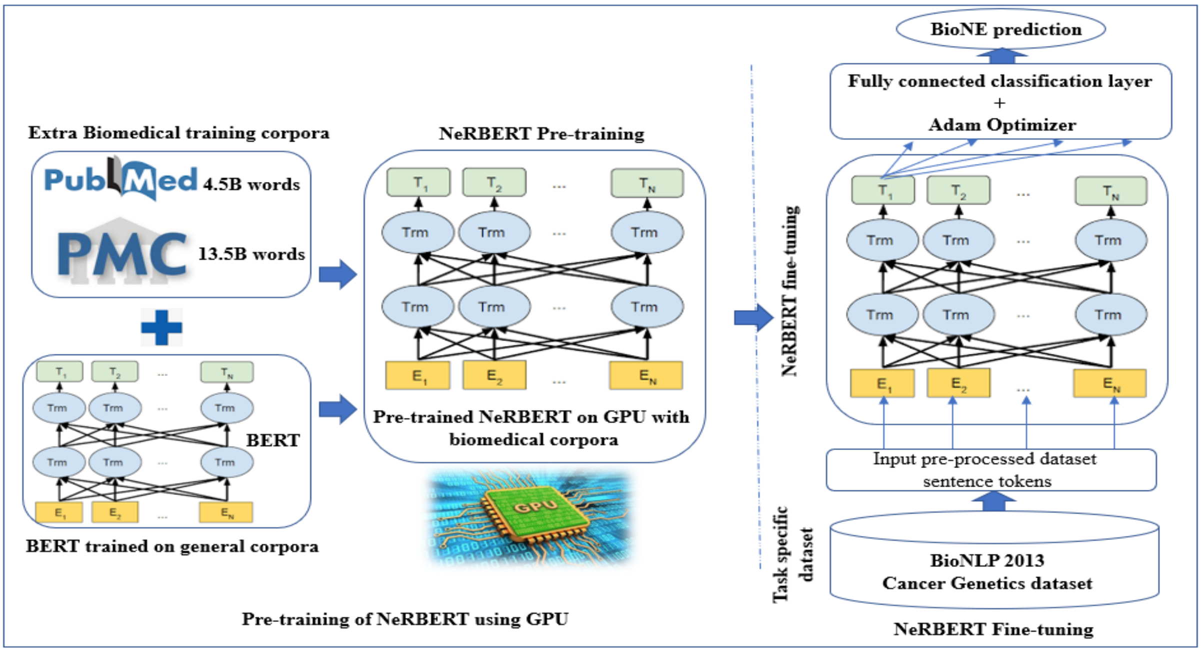
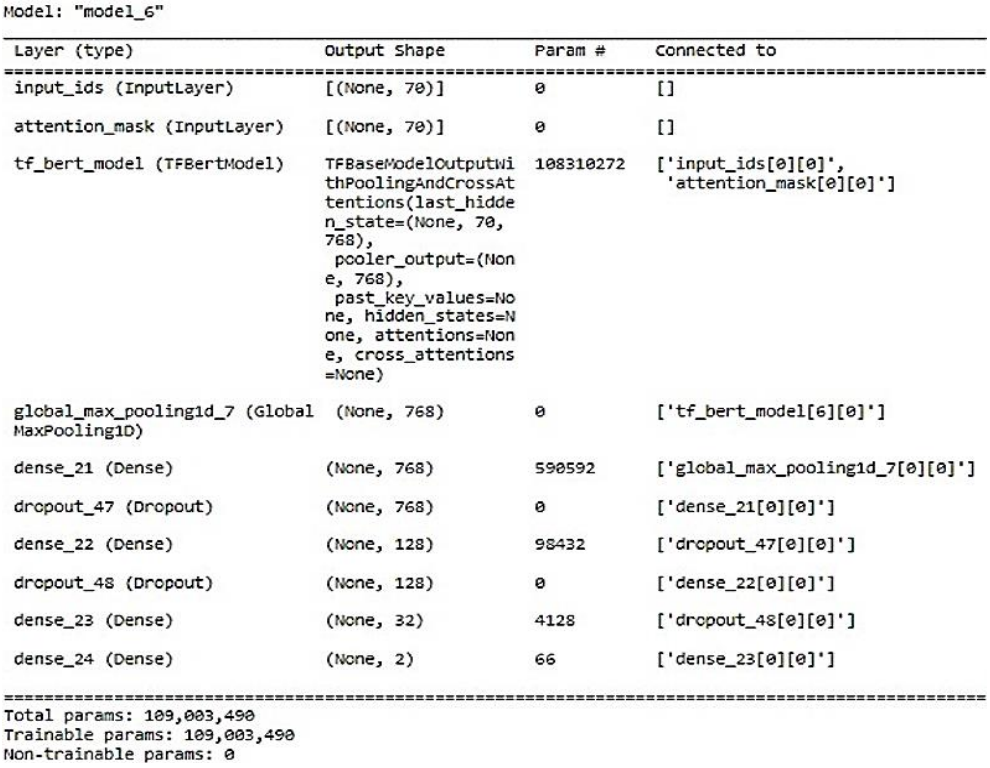

Extra Biomedical training corpora
PubMed 4.5B words
PMC 13.5B words
BERT
BERT trained on general corpora
NeRBERT Pre-training
Pre-trained NeRBERT on GPU with biomedical corpora
GPU
Pre-training of NeRBERT using GPU
NeRBERT fine-tuning
BioNE prediction
Fully connected classification layer + Adam Optimizer
Input pre-processed dataset sentence tokens
Task specific dataset
BioNLP 2013 Cancer Genetics dataset
NeRBERT Fine-tuning
T1 T2 ... TN
Trm
E1 E2 ... EN

Figure 5. Pre-training and fine-tuning of NeRBERT model



```
Model: "model_6"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_ids (InputLayer)         [(None, 70)]         0           []

 attention_mask (InputLayer)    [(None, 70)]         0           []

 tf_bert_model (TFBertModel)    TFBaseModelOutputWi  108310272   ['input_ids[0][0]',
                                thPoolingAndCrossAt               'attention_mask[0][0]']
                                tentions(last_hidde
                                n_state=(None, 70,
                                768),
                                 pooler_output=(Non
                                e, 768),
                                 past_key_values=No
                                ne, hidden_states=N
                                one, attentions=Non
                                e, cross_attentions
                                =None)

 global_max_pooling1d_7 (Global  (None, 768)         0           ['tf_bert_model[6][0]']
 MaxPooling1D)

 dense_21 (Dense)               (None, 768)          590592      ['global_max_pooling1d_7[0][0]']

 dropout_47 (Dropout)           (None, 768)          0           ['dense_21[0][0]']

 dense_22 (Dense)               (None, 128)          98432       ['dropout_47[0][0]']

 dropout_48 (Dropout)           (None, 128)          0           ['dense_22[0][0]']

 dense_23 (Dense)               (None, 32)           4128        ['dropout_48[0][0]']

 dense_24 (Dense)               (None, 2)            66          ['dense_23[0][0]']

==================================================================================================
Total params: 109,003,490
Trainable params: 109,003,490
Non-trainable params: 0
```

Figure 6. Fine-tuned weights and other hyper-parameters in NeRBET

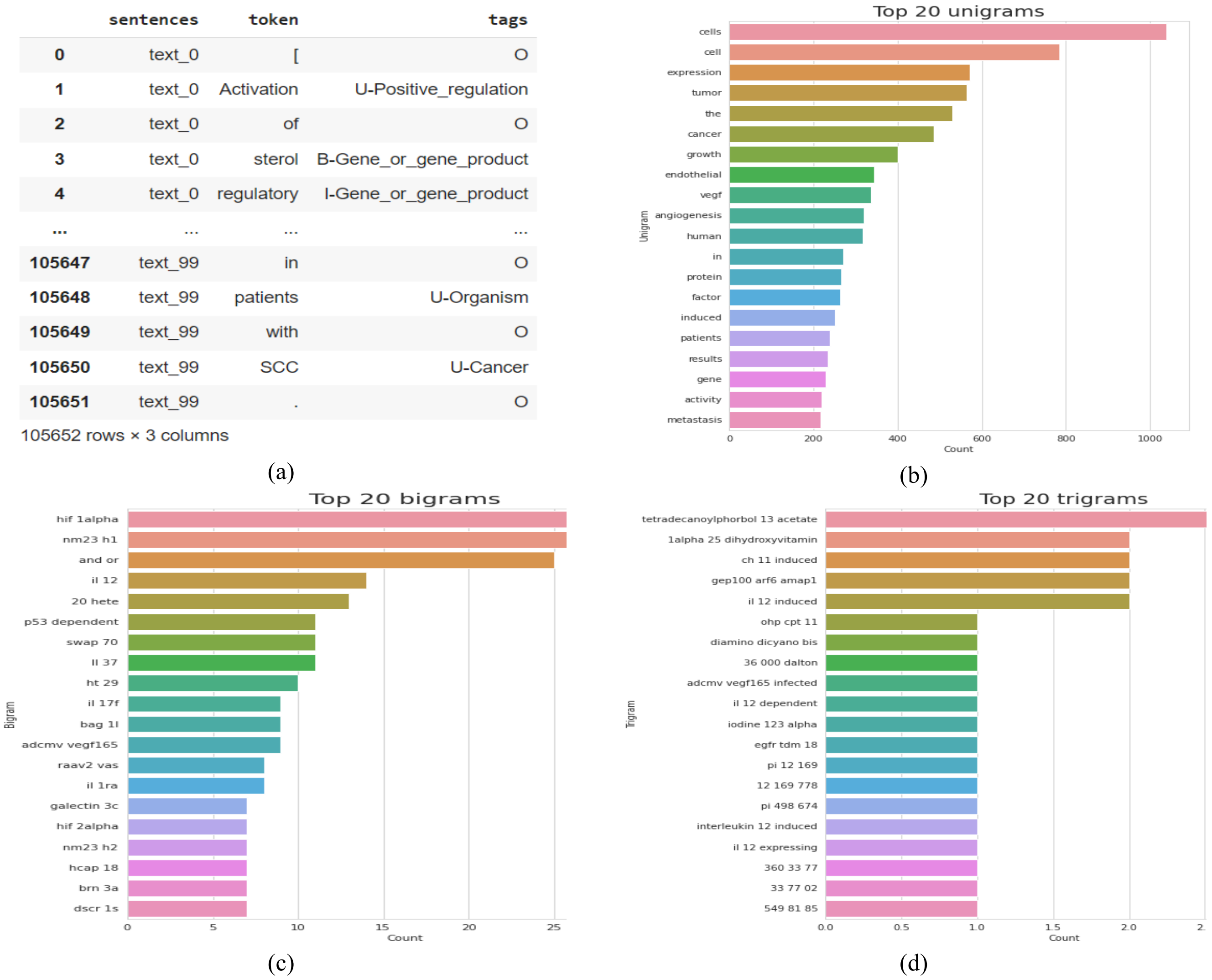
4. RESULTS AND DISCUSSION

The statistics of the complex BioNLP 2013 cancer genetics (CG) dataset is shown in Table 1. This dataset was chosen as it has the maximum number of entity and event types and thus is the most complex among the available (and accessible) BioNLP datasets.

Since the dataset is complex, visualization of its various characteristics can be useful for understanding the output of any NER system and for debugging and improving the system. Hence we carry out exploratory data analysis on the dataset. Figure 7(a) shows the Train Data Frame. This is formed from the 'train' dataset with analysis performed on the 'tokens' and 'tags' column. The n-gram visualization in the dataset is shown in Figure 7(b)-(f). Unigram, bigram and trigram of both tokens and tags present in the dataset have been captured from an exploratory understanding purpose. It shows the top 20 term frequencies for both tokens and tags in the train data frame. As seen from Figure 7(f) only one bigram exists in tags. The token and tag word clouds are shown in Figure 8.

Table 1. BioNLP 2013 CG dataset statistics

Parameter	Train_ data	Vali_data	Test_data
No. of lines	300	100	200
No. of entities	300	100	200
No. of entity values	18404	6085	6955
Avg. word count in lines	1334	1316	1272
Avg. entity value count per line	61	60	34



	sentences	token	tags
0	text_0	[	O
1	text_0	Activation	U-Positive_regulation
2	text_0	of	O
3	text_0	sterol	B-Gene_or_gene_product
4	text_0	regulatory	I-Gene_or_gene_product
...	...	...	...
105647	text_99	in	O
105648	text_99	patients	U-Organism
105649	text_99	with	O
105650	text_99	SCC	U-Cancer
105651	text_99	.	O

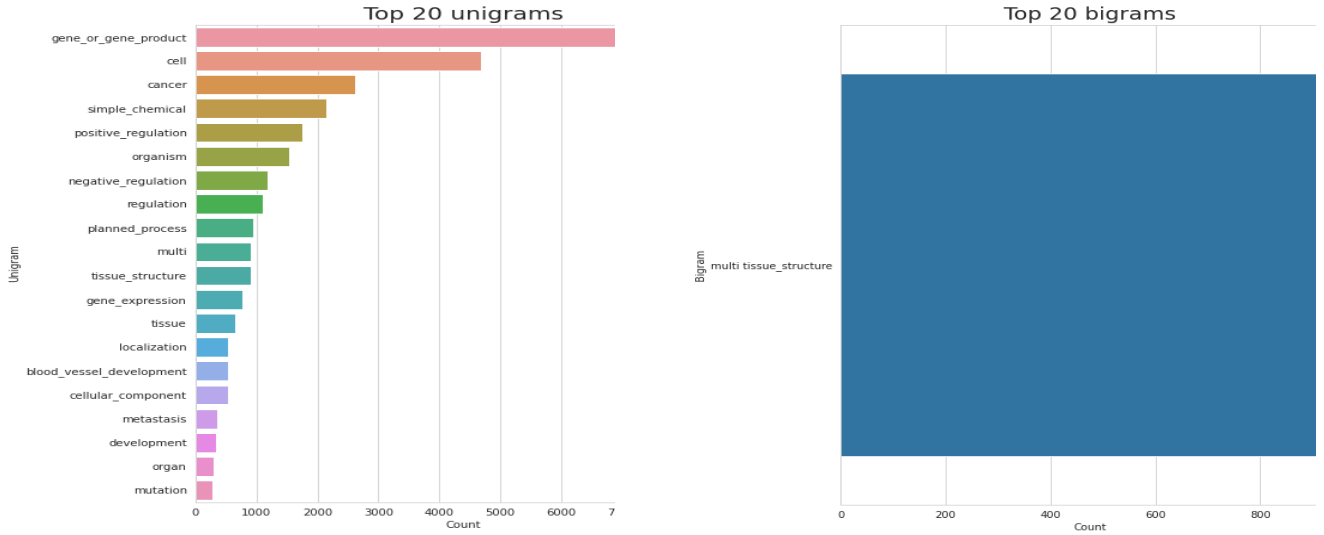
105652 rows × 3 columns

(a)

(b)

(c)

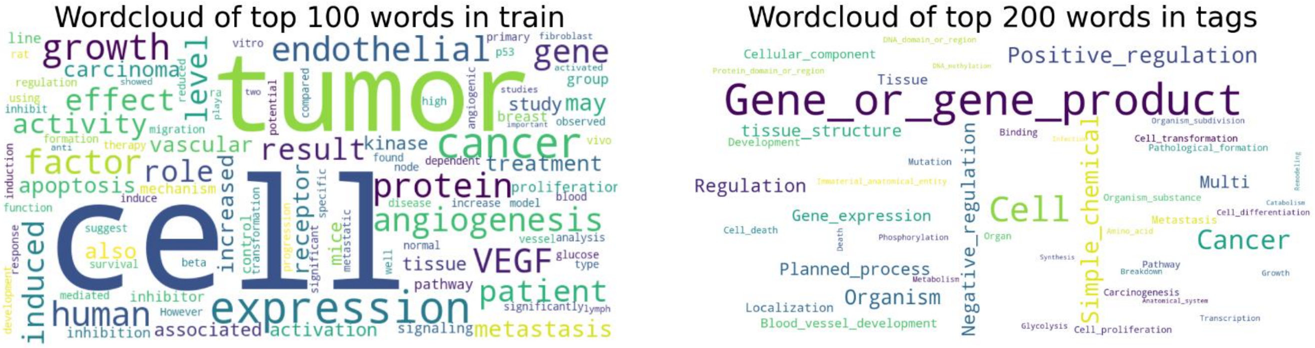
(d)



(e)

(f)

Figure 7. (a) Train Data Frame; (b) unigrams on tokens; (c) bigrams on tokens; (d) trigrams on tokens; (e) unigrams on tags; and (f) bigrams on tags



(a)

(b)

Figure 8. Word cloud for (a) tokens; and (b) tags in the dataset

Table 2. Experimental results of ML classifiers and ensemble model on BioNLP2013 dataset (in %)

Classifier/Model	Accuracy	Precision	Recall	F1-score
LOGR	77.91	73.28	71.98	72.62
SVM-LIN	78.38	72.48	69.45	70.93
DT	80.23	76.87	72.18	74.45
KNN	74.56	70.57	72.19	71.37
ELM	70.18	65.48	63.87	64.66
Ensemble Model	84.38	78.38	75.87	77.10

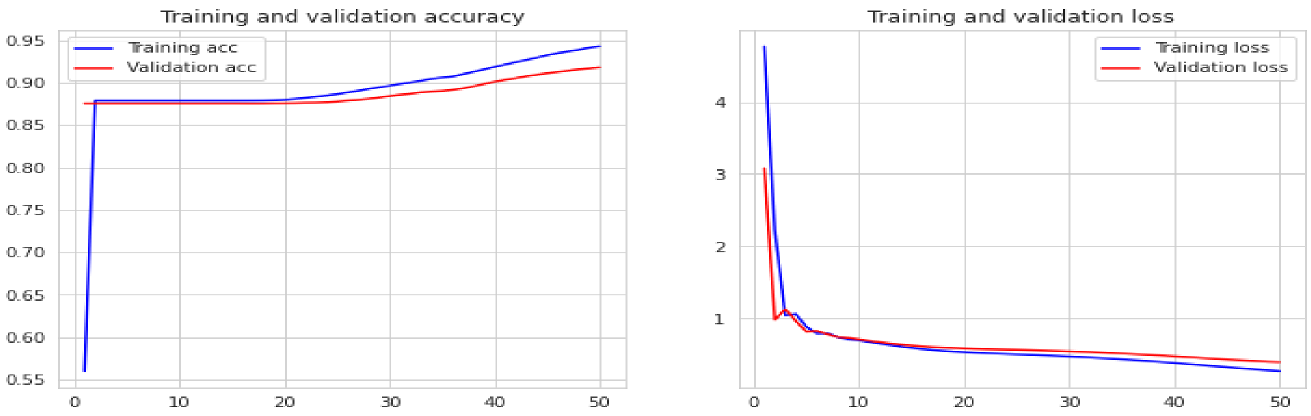


Figure 9. Training and validation plots for Bi-LSTM model

The results of NE classification using the ensemble model and individual base classifiers is shown in Table 2. All individual classifiers are outperformed by the ensemble model which provides generalization. With scores of 84.38% and 77.10% respectively, the implemented majority voting ensemble returns the highest performance in terms of accuracy and F1-score. The decision tree, which has an accuracy of 80.23%, outperformed all other individual classifiers, followed by SVM-LIN, which has an accuracy of 78.38%. With an F1-score of 64.66%, the ELM algorithm returned the lowest results. The training and validation accuracy plots of the Bi-LSTM model is illustrated in Figure 9. From the plots, it is observed that the accuracy of the model is a high 87% which is good. From training loss plot, it is observed that the model fits well to the training data and it fits new data also well as observed from validation loss plot.

Intra-model experimental results of all models are compared in Table 3. It can be observed that $\text{BERT}_{\text{BASE}}$ pre-trained on general corpora achieves an accuracy of 88.74% and an F1-score of 83.85%. It outperforms the ensemble model and Bi-LSTM model on F1-score by 6.75 pp and 3.16 pp respectively. However, the proposed NeRBERT model which is pre-trained on general corpora and additional biomedical corpora achieves the highest accuracy of 91.08% and F1-score of 89.28%. There is an F1-score gain of 12.18 pp, 8.59 pp and 5.43 pp over ensemble, Bi-LSTM and $\text{BERT}_{\text{BASE}}$ models respectively. NeRBERT performs better than other models due to extensive pre-training on relevant corpora and better ability to consider a word's context. NeRBERT returns different vectors for the same word based on the words around it as opposed to previous word embedding techniques like GloVe, word2vec, and TF-IDF that always return the same vector for the word regardless of the context.

With 8 x Nvidia Volta 100 GPUs, each with 32GB memory the NeRBERT model pre-training lasts for ~11 days which is less than half the time taken compared to a similar 8-socket CPU-only system. The many-core architecture of the GPUs and the fact that all mini-batch training assigned to it fits into its large memory helps to accelerate the pre-training task thereby providing a viable platform for scaling BERT training.

In Inter-model comparison shown in Table 4, we compare the experimental results of NeRBERT with existing state-of-the-art models like CNN, HunFlair, SciSpacy and Huner on the same dataset. The cross-comparison results are shown in Table 4. It is observed that NeRBERT outperforms all models with an F1-score gain of 1.57 pp above HunFlair, the next best model compared.

Table 3. Intra-model experimental results on BioNLP 2013 CG dataset (in %)

Models	Word Embedding	Accuracy	Precision	Recall	F1-score
Ensemble Model	word2vec	84.38	78.38	75.87	77.10
Bi-directional LSTM	word2vec	87.89	82.18	79.26	80.69
$\text{BERT}_{\text{BASE}}$	general corpora	88.74	83.59	84.12	83.85
Proposed NeRBERT model	general corpora + biomedical corpora	91.08	90.21	88.37	89.28

Table 4. Inter-model experimental results on BioNLP 2013 CG dataset

Reference	Model	F1-score(%)
Crichton et al. [19]	CNN	85.98
Weber et al. [20]	HunFlair	87.71
Neumann et al. [21]	SciSpacy	66.18
Weber et al. [22]	Huner	71.22
Proposed model	NeRBERT	89.28

5. CONCLUSIONS

In this work, we presented NeRBERT which is a biomedical cancer phenotyping NER tagger leveraging the BERT architecture. We pre-train NeRBERT with large biomedical corpora from two sources namely, PubMed abstracts and PubMed central full-text articles using Nvidia GPUs. To assess its performance, in intra-model evaluation, we compare its performance with three models, an ensemble classifier, a Bi-LSTM model and the plain BERT model. For ensemble and Bi-LSTM models, we use word2vec word embeddings and train using the most complex among the available (and accessible) BioNLP shared task datasets, the BioNLP 2013 CG dataset. The proposed NeRBERT model outperformed all the models with an F1-score gain of 12.18 pp, 8.59 pp and 5.43 pp over ensemble, Bi-LSTM and $\text{BERT}_{\text{BASE}}$ models respectively. The model pre-training leverages GPUs and helps to reduce the model pre-training time by less than half compared to an equivalent CPU-only system providing a viable platform for scaling BERT training. In inter-model comparison with existing state-of-the-art NER models, NeRBERT outperforms all models with an F1-score gain of 1.57 pp above HunFlair, the next best model compared, emerging as a robust biomedical and cancer phenotyping NER tagger. Encouraged by the results, the model will be extended to other disease specific vocabulary and for more complex, performance-lagging biomolecular event extraction tasks as part of future research for which NER is a sub-task to improve biomolecular event extraction performance.

REFERENCES

[1] Nayel, A., Hamada Shashirekha, H.L. (2017). Improving NER for clinical texts by ensemble approach using segment representations. In: Proceedings of the 14th International Conference on Natural Language Processing, Kolkata, pp. 197-204. https://aclanthology.org/W17-7525/, accessed on Sep. 12, 2022.

[2] Bali, M., Murthy, P.V.R. (2021). Bio-molecular event extraction using classifier ensemble-of-ensemble technique. Advances in Intelligent Systems and Computing, 1175: 445-462. http://dx.doi.org/10.1007/978-981-15-5619-7_32

[3] Copara, J., Naderi, N., Knafou, J., Ruch, P., Teodoro, D. (2020). Named entity recognition in chemical patents using ensemble of contextual language models. arXiv preprint arXiv:2007.12569. https://doi.org/10.48550/arXiv.2007.12569

[4] Doan, S., Collier, N., Xu, H., Duy, P.H., Phuong, T.M. (2012). Recognition of medication information from discharge summaries using ensembles of classifiers. BMC medical informatics and decision making, 12(1): 1-10. https://doi.org/10.1186/1472-6947-12-36

[5] Erdengasileng, A., Han, Q., Zhao, T., Tian, S., Sui, X., Li, K., Wang, W., Wang, J., Hu, T., Pan, F., Zhang, Y. (2022). Pre-trained models, data augmentation, and ensemble learning for biomedical information extraction and document classification. Database. https://doi.org/10.1093/database/baac066

[6] Dang, T.H., Le, H.Q., Nguyen, T.M., Vu, S.T. (2018). D3NER: biomedical named entity recognition using CRF-BiLSTM improved with fine-tuned embeddings of various linguistic information. Bioinformatics 34(20): 3539-3546. https://doi.org/10.1093/bioinformatics/bty356

[7] Saad, F., Aras, H., Hackl-Sommer, R. (2020). Improving named entity recognition for biomedical and patent data using bi-LSTM deep neural network models. International Conference on Applications of Natural Language to Information Systems, 12089: 25-36. https://doi.org/10.1007/978-3-030-51310-8_3

[8] Rivera-Zavala, R.M., Martínez, P. (2021). Analyzing transfer learning impact in biomedical cross-lingual named entity recognition and normalization. BMC bioinformatics, 22(1): 1-23. https://doi.org/10.1186/s12859-021-04247-9

[9] Wu, H., Ji, J., Tian, H., Chen, Y., Ge, W., Zhang, H., Yu, F., Zou, J., Nakamura, M., Liao, J. (2021). Chinese-named entity recognition from adverse drug event records: Radical embedding-combined dynamic embedding–based BERT in a bidirectional long short-term conditional random field (Bi-LSTM-CRF) model. JMIR Medical Informatics, 9(12): e26407. https://doi.org/10.2196/26407

[10] Cho, H., Lee, H. (2019). Biomedical named entity recognition using deep neural networks with contextual information. BMC bioinformatics, 20(1), pp.1-11. https://doi.org/10.1186/s12859-019-3321-4

[11] Devlin, J., Chang, MW., Lee, K., Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. https://doi.org/10.48550/arXiv.1810.04805

[12] Peng, Y., Yan, S., Lu, Z. (2019). Transfer learning in biomedical natural language processing: an evaluation of BERT and ELMo on ten benchmarking datasets. arXiv preprint arXiv:1906.05474. https://doi.org/10.48550/arXiv.1906.05474

[13] Peng, Y., Chen, Q., Lu, Z., (2020). An empirical study of multi-task learning on BERT for biomedical text mining. arXiv preprint arXiv:2005.02799. https://doi.org/10.48550/arXiv.2005.02799

[14] Alrowili, S., Vijay-Shanker, K. (2021). BioM-transformers: building large biomedical language models with BERT, ALBERT and ELECTRA. Proceedings of the 20th Workshop on Biomedical Language Processing, pp. 221-227. http://dx.doi.org/10.18653/v1/2021.bionlp-1.24

[15] Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. ACM Transactions on Computing for Healthcare (HEALTH), 3(1): 1-23. https://doi.org/10.48550/arXiv.2007.15779

[16] Zhu, R., Tu, X. and Huang, J.X. (2021). Utilizing BERT for biomedical and clinical text mining. Data Analytics in Biomedical Engineering and Healthcare, pp. 73-103. https://doi.org/10.1016/b978-0-12-819314-3.00005-7

[17] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4): 1234-1240. https://doi.org/10.1093/bioinformatics/btz682

[18] Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. https://doi.org/10.48550/arXiv.1301.3781

[19] Crichton, G., Pyysalo, S., Chiu, B., Korhonen, A. (2017). A neural network multi-task learning approach to biomedical named entity recognition. BMC Bioinformatics, 18(1): 368. https://doi.org/10.1186/s12859-017-1776-8

[20] Weber, L., Sanger, M., Munchmeyer, J., Habibi, M., Ulf, L., Akbik, A. (2021). HunFlair: an easy-to-use tool for state-of-the-art biomedical named entity recognition. Bioinformatics, 37(17): 2792-2794. https://doi.org/10.1093/bioinformatics/btab042

[21] Neumann, M., King, D., Beltagy, I., Ammar, W. (2019). ScispaCy: fast and robust models for biomedical natural language processing. 18th BioNLP Workshop and Shared Task. Association for Computational Linguistics, Florence, Italy, pp. 58-66. https://doi.org/10.18653/v1/W19-5034

[22] Weber, L., Münchmeyer, J., Rocktäschel, T., Habibi, M., Leser, U. (2020). HUNER: improving biomedical NER with pretraining. Bioinformatics, 36(1): 295-302. https://doi.org/10.1093/bioinformatics/btz528