TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 19, No. 6, December 2021, pp. 1884~1891
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v19i6.21667

A novel deep learning architecture for drug named entity recognition

T. Mathu, Kumudha Raimond
Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India

Article Info

Article history:

Received Mar 1, 2021
Revised Oct 10, 2021
Accepted Oct 18, 2021

Keywords:

Drug named entity recognition
Natural language processing
Residual LSTM
Sentence level embedding
Stacked Bi-LSTM

ABSTRACT

Drug named entity recognition (DNER) becomes the prerequisite of other medical relation extraction systems. Existing approaches to automatically recognize drug names includes rule-based, machine learning (ML) and deep learning (DL) techniques. DL techniques have been verified to be the state-of-the-art as it is independent of handcrafted features. The previous DL methods based on word embedding input representation uses the same vector representation for an entity irrespective of its context in different sentences and hence could not capture the context properly. Also, identification of the n-gram entity is a challenge. In this paper, a novel architecture is proposed that includes a sentence embedding layer that works on the entire sentence to efficiently capture the context of an entity. A hybrid model that comprises a stacked bidirectional long short-term memory (Bi-LSTM) with residual LSTM has been designed to overcome the limitations and to upgrade the performance of the model. We have contrasted the achievement of our proposed approach with other DNER models and the percentage of improvements of the proposed model over LSTM-conditional random field (CRF), LIU and WBI with respect to micro-average F1-score are 11.17, 8.8 and 17.64 respectively. The proposed model has also shown promising result in recognizing 2- and 3-gram entities.

Corresponding Author:

T. Mathu
Department of Computer Science and Engineering
Karunya Institute of Technology and Sciences
Karunya Nagar, Coimbatore 641114, TamilNadu, India
Email: mathu@karunya.edu

1. INTRODUCTION

Named entity recognition (NER) is an essential task of information extraction (IE), and is often utilized in natural language processing (NLP). The NER that defines and classifies the labels of drugs into predefined classes from unstructured medical texts is referred to as drug named entity recognition (DNER) [1]. The research on the DNER becomes prominent ever since the identification of drug-drug interactions (DDI) and adverse drug reaction (ADR) have become important in the branch of pharmacodynamics and pharmacokinetics. However, many studies [2]-[5] have shown that there is not much specific work for DNER in recent years. Techniques like rule-based framework, machine learning (ML) methods and deep learning (DL) techniques were employed for the DNER. The rule-based and ML techniques heavily depend on the field/subject knowledge of the human professionals to devise the features for designing the recognition model. DL uses more than one layer of artificial neural networks that recognizes the named entities [2], [6]–[8]. When compared to traditional approaches, DL is more

advantageous in automatically recognizing hidden features. The latest DL techniques do not require the intervention of the human experts in constructing the features from unstructured text. The most common DL model used is long short-term memory (LSTM) which helps to preserve the long-range dependency especially while dealing with sequential text. Bidirectional LSTM model (Bi-LSTM) which reads the text both in forward and reverse directions is used to capture the context of the word for better prediction. Word embedding models like Word2Vec, GloVe and FastText are usually used for word embedding in DL algorithms. We have observed that the major limitation of word embedding models is that they work with the same vector for all the mentions of an entity in the article and hence could not capture the context properly.

The previous research works for DNER models based on DL have used word2vec [9] and Glove [10] word embedding and character embedding models to represent the input. It was found that the word embedding model could not capture the semantic feature of the words in the sentence completely. Because in word2vec, every unique word throughout the corpus will have the same vector in the vector space. Consider the following sentences:

- Sentence 1: MAO inhibitors prolong and intensify the anticholinergic (drying) effects of antihistamines. (MAO – B-group, inhibitors – I-group).
- Sentence 2: In the absence of formal clinical drug interaction studies, caution should be exercised when administering TAXOL concomitantly with known substrates or inhibitors of the cytochrome P450 isoenzymes CYP2C8 and CYP3A4.

In sentence 1, the word "MAO inhibitors" is annotated as a class of drug 'group' and hence "inhibitors" is annotated with "I-group". But in sentence 2, the word "inhibitors" does not refer to any class of drug and hence should be identified as "O". If word embedding models like Word2Vec or Glove are used, the same word vector is used for both the mentions and hence would not be recognized correctly based on the context. Though recurrent neural networks (RNN) have been used for various NER models, the full potential is not realized when an LSTM or Bi-LSTM model alone is used. Also, in DNER, several drug names are n-gram entities. For instance, "albendazole sulfoxide" (2-gram entity), "central nervous system depressants" (4-gram entity).

In this paper, to overcome the challenges mentioned above, a novel architecture has been proposed by incorporating a sentence embedding model, stacked Bi-LSTM and residual LSTM. A sentence embedding model called ELMo [11], is used to deal with entire sentence to capture the context properly. Since the model is extremely contextualized, both syntax and semantic characteristics of the word are modelled. The main advantage is that it would be able to generate vectors for words that are not seen during training. Also, to enhance the DNER model, the designed architecture utilizes the power of the RNN in drug prediction. The architecture consists of stacked Bi-LSTM layers [12] and a residual LSTM layer [13].

Initially, the sentence embedding model ELMo used in this architecture creates word vectors by functioning on an entire sentence to efficiently capture the context of the word. The specific context and the variations in the content is identified and helps the machine to understand better, unlike having the same word vector for every mention of the word in other word embedding models. Though a single Bi-LSTM layer itself could possibly recognize the entities, the power of the RNN shall be enhanced by arranging multiple Bi-LSTM layers on top of each other. We have experimented with layers of Bi-LSTM in this model. To avoid stacked Bi-LSTM suffering from the vanishing gradient problem, the residual LSTM is used. The residual connection is used between the two Bi-LSTM layers. It allows the gradients to pass through the network directly and also helps to preserve the long range dependencies [14], [15]. We have tested our proposed approach with a test data set and the performance is compared with other DNER models and found to improve the recognition rate over the preceding state-of-the-art models. We have also evaluated the performance in identifying the 2-, 3-, and 4- gram entities which is a major challenge in DNER. The content of this paper is categorized as follows: section 2 extends the research method. Results are discussed in section 3 and conclusion is given in section 4.

2. RESEARCH METHOD

In our model, we have used Bi-LSTM network where the sentence is read in both forward and reverse direction. The works represented in the following papers [16]-[19] have shown that the classification performance could be enhanced further by stacking many Bi-LSTM layers. Hence, in our model, we have used two Bi-LSTM layers stacked above each other as shown in Figure 1. When the sentences are long as given in the example in section 1, it is necessary to remember the long range dependency of the entity from the first instance and the next instance. When the depth of the neural network increases, accuracy of prediction also increases. Residual LSTM is used to avoid stacked Bi-LSTM suffering from the vanishing gradient problem and also it is suitable for handling such long range dependencies.

2.1. Data source and preprocessing

In this paper, we have adopted the tagged corpus namely, the DDI2013 drugbank dataset [20], the benchmark dataset to train the deep learning model. The data is categorized with the following labels: drug, brand, group and drug_n [20]. We have preprocessed the DDI2013 training dataset in such a way that the sentence is split into tokens and every token is labeled with the corresponding class labels. Since there are several n-gram words available as drug names in the biomedical articles, it is necessary to capture the beginning and end of the entity. The most common tagging scheme known as BIO tagging is useful to capture these details. Table 1 gives the count of the various BIO tags available in the training dataset. In BIO tagging, B, I and O correspond to the beginning, inside and outside or non-entity token respectively. For example, the label B-group, I-group represent the beginning and inside of the group respectively and O represents non-entity tokens.

Table 1. Tags and its counts

Tag	B-brand	I-brand	B-drug	I-drug	B-group	I-group	B-drug_n	I-drug_n
Count	1425	48	8333	549	3027	1937	96	29

2.2. Components of the model

2.2.1. Sentence embedding layer

Sentence embedding techniques addresses whole sentences and their semantic data as vectors. This aids the machine in understanding the specific context and different subtleties in the whole content. ELMo (embedding from language models) is an embedding model [11] which functions for an entire sentence. The word representation used in this model is deeply contextualized that can model characteristics such as syntax and semantics of the word and also finds how these characteristics can be used for different linguistic contexts [21]. Since this is also a character based representation, instead of simply looking into words and their vectors, it generates vectors that form representations of tokens that are not seen during training.

2.2.2. Stacked Bi-LSTM layers

Bi-LSTM, which took its idea from bidirectional RNN that proceeds in both directions – forward and reverse – having independent hidden layers for each direction. These hidden layers are linked to a common output layer. Bi-LSTM networks are found to be better in many research areas such as traffic prediction [12], speech recognition [22] and phoneme classification [23].

The previous studies [22], [23] have proved that deep LSTM models ie. stacked LSTM models with many hidden layers can develop a successively more significant level of description for the sequential data and hence could perform more effectively as illustrated in [12] using a two-layer Bi-LSTM model for traffic prediction. The effectiveness of stacked Bi-LSTM networks for better classification and regression tasks was also demonstrated in [17], [18], and [24].

In our work, based on the previous research works, we have also adopted the two layers of stacked Bi-LSTM. The lower layer of Bi-LSTM is more appropriate for extracting useful information from the input vectors. The unique vectors obtained for each word in the sentence using the sentence embedding model is given as input to the Bi-LSTM layer 1 which helps in capturing the features for predicting the drug categories. As we have used two stacked layers, the second layer or the top layer of the stack utilizes the features learned from the output of the lower layer. It also learns many complex features to enhance the achievement of the model.

2.2.3. Residual LSTM connection

To overcome the issue of vanishing gradients, a residual LSTM connection is used which provides a bypass link between the layers [13]. The shortcut path could be from any lower layers. In this paper, we have used residual LSTM as a shortcut between the stacked Bi-LSTM layers. Since we have used only two stacked Bi-LSTM layers, the shortcut is taken from the output of layer 1 and added with the output of layer 2 as shown in Figure 1.

2.3. Architecture of the proposed system

The system architecture of the proposed model for DNER system is shown in Figure 2. The main novelty of this architecture is the inclusion of sentence embedding layer which enables the system to capture the semantic information better than word or character embedding models. In addition, the architecture comprises of stacked Bi-LSTM (with two layers) and residual LSTM to capture the complex features of the sentences and to overcome the problem of vanishing gradients respectively.

Consider an input sequence w=(w_1, w_2,…w_n) where w_1,..w_n represents the words in the sequence padded with a fixed length for each sentence and series of output tags e=(e_1, e_2…e_n) where e_1, e_2,..e_n refers to entities. The sentence embedding layer ELMo creates the vector to every word for sentence w. The input for Bi-LSTM layer 1 is the sequence of word vectors found from the sentence embedding layer. The sequence of hidden states forms the output of the layer 1 and that in turn becomes the input to the Bi-LSTM layer 2. The Bi-LSTM layers have two passes in each layer, namely forward pass/forward layer and reverse pass/reverse layer.

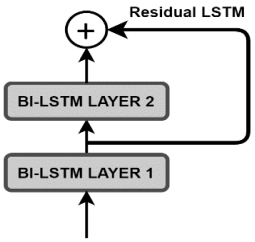

Figure 1. Stacked Bi-LSTM with residual LSTM connection

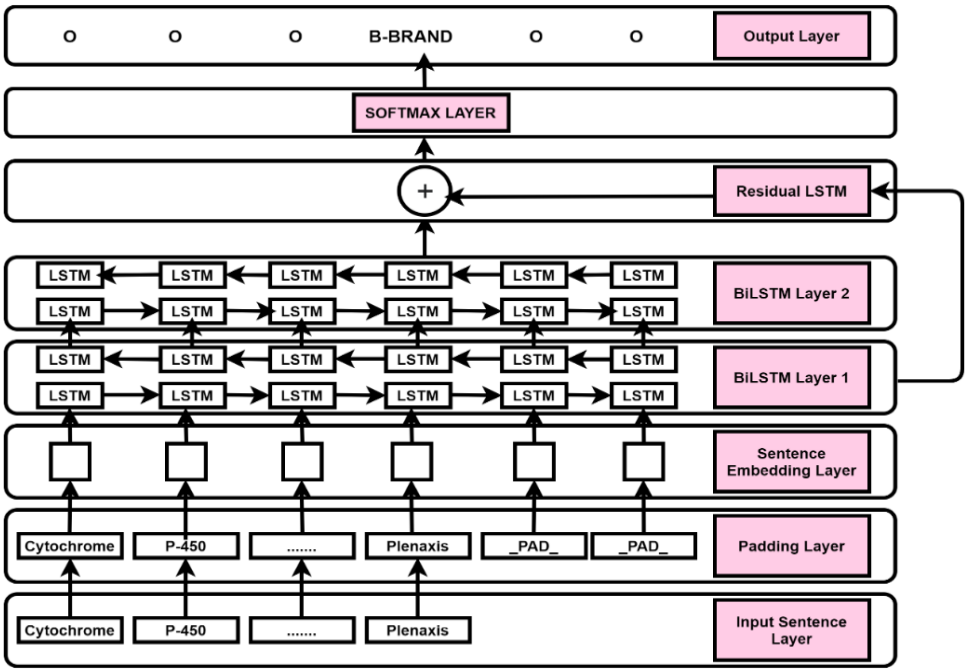

Figure 2. System architecture of the proposed model

In the forward layer, the input sequence w_t is fed from time t=1 to T_n and from t=T_n to 1 in the reverse layer. The hidden vector sequence and the output sequence are computed from the Bi-LSTM layer. The hidden vector sequence can be forward sequence and reverse sequence represented by $\vec{s}_t$ *and* $\overleftarrow{s}_t$ respectively. The forward layer is iterated from t=1 to T_n and the reverse layer is iterated from t=T_n to 1. The calculation of forward hidden vector, reverse hidden vector and output sequence respectively are shown as in (1), (2) and (3).

$$\vec{s}_t = H(W_{w\vec{s}}w_t + W_{\vec{s}\vec{s}}\vec{s}_{t-1} + b_{\vec{s}}) \quad (1)$$

$$\overleftarrow{s}_t = H(W_{w\overleftarrow{s}}w_t + W_{\overleftarrow{s}\overleftarrow{s}}\overleftarrow{s}_{t+1} + b_{\overleftarrow{s}}) \quad (2)$$

$$y_t = W_{\vec{s}y}\vec{s}_t + W_{\overleftarrow{s}y}\overleftarrow{s}_t + b_y \quad (3)$$

Where H symbolizes the hidden layer function, W symbolizes the various weight matrices, b symbolizes the bias vectors and y_t denotes the output layer variable. The weights $W_{w\vec{s}}$, $W_{\vec{s}\vec{s}}$, $W_{w\overleftarrow{s}}$, $W_{\overleftarrow{s}\overleftarrow{s}}$ and the biases $b_{\vec{s}}$, $b_{\overleftarrow{s}}$ represents the model parameters in (1) and (2). Then the bias parameter b_y is concatenated with forward hidden layer $\vec{s}_t$ and reverse hidden layer $\overleftarrow{s}_t$ to get the output layer y_t as in (3). In general, when more than one Bi-LSTM layer is used, the forward and reverse hidden sequence can be computed for n=1 to N and t=1 to T_n as given in (4) and (5).

$$\vec{s}_t^n = H(W_{\vec{s}^{n-1}\vec{s}_n} s_t^{n-1} + W_{\vec{s}^n\vec{s}_n} s_t^n + b_s^n) \quad (4)$$

$$\overleftarrow{s}_t^n = H(W_{\overleftarrow{s}^{n-1}\overleftarrow{s}_n} s_t^{n-1} + W_{\overleftarrow{s}^n\overleftarrow{s}^n} s_t^n + b_s^n) \quad (5)$$

The output sequence is calculated as given in (6).

$$y_t = W_{s^N} s_t^N + b_y \quad (6)$$

Now, the residual LSTM connection is applied by adding the output sequences of Bi-LSTM layer 2 with w. It is referred by H(w) and is shown in (7).

$$H(w) = y_t + w \quad (7)$$

The vanishing gradient problem could be resolved by the application of residual LSTM since the gradients could pass through the layers directly by using the addition operator. The residual LSTM [13] permits different layers of LSTM to adequately train complex networks with an optional temporal shortcut path from deeper levels. Finally, the scores given for each label by the Bi-LSTM layers are provided as an input into the softmax classifier output layer O_w, as given in (8).

$$O_w = softmax\big(H(w)\big) \quad (8)$$

This layer produces the predicted probabilities for all the labels to each word used for classification which includes B-drug, I-drug, B-brand, I-brand, B-group, I-group and O. The label which has got the highest prediction in the sequence would be considered as the label for the word.

2.4. Pseudocode of the proposed DNER Model

The general steps of the proposed DNER system based on stacked Bi-LSTM and residual LSTM is shown in Algorithm 1.

Algorithm 1:

1	Input	Sentences of various lengths from DDI2013 Drugbank training dataset.
2	Preprocessing	Tokenize the sentences from the input dataset.
3		For each token, include Part-of-Speech (POS) tags and BIO drug labels.
4		Each tokenized sentence is padded with _PAD_ tokens to bring it to a fixed length.
5	Model Construction	Construct Sentence embedding using ELMo for the pre-processed input dataset.
6		Implement Stacked Bi-LSTM layers (two layers) to obtain the previous and future contextual information for more accurate prediction ie. the sequence of hidden vectors obtained from Bi-LSTM layer 1 is given to Bi-LSTM layer 2 using (2),(3),(4),(5,(6), and (7).
7		Establish a Residual connection using a vector addition between the Bi-LSTM layer 1 output and Bi-LSTM layer 2 output to prevent Bi-LSTM suffering from the vanishing gradient problem as in (8)
8		Finally, apply softmax function in the output layer to classify drug names into multiple categories of drugs as in (9)

3. RESULTS AND DISCUSSION

3.1. Performance metrics

The DNER model needs to be evaluated by appropriate and unambiguous metrics to rightly judge the performance of the model. Precision, recall and F1-score are used as measurements to assess the model. As four different entities are available in DDI2013 corpus, it is necessary to compute the overall performance of all the entity classes. In this regard, we take the micro-average F1-score [25] metric for the comparison

with other systems. Micro-average F1-score, as in (9) is defined as the harmonic mean of micro-average precision (mP) and micro-average recall (mR) and it is given in (10) and (11) respectively.

$$\text{Micro-average F1-score} = 2 * (mP \times mR)/(mP + mR), \text{ where} \quad (9)$$

$$mP=tp_1 + tp_2 +\ldots+tp_n/(tp_1 + tp_2 +\ldots+tp_n + fp_1+ fp_2 +\ldots+fp_n) \quad (10)$$

$$mR=tp_1 + tp_2 +\ldots+tp_n/(tp_1 + tp_2 +\ldots+tp_n + fn_1+ fn_2 +\ldots+fn_n) \quad (11)$$

tp, fp and fn represents the true positive, false positive and false negative respectively.

3.2. Experimental setup

The datasource used is described in section 2.2. The training data given in DDI2013 drugbank corpus is preprocessed and we have used 4990 sentences with 8006 unique words and 30% of the dataset is considered as test dataset. We used 'adam' optimizer with loss as 'sparse_categorical_crossentropy' for compiling the model. The batch size is 32 and the number of epochs is made as 8. The recurrent dropout is taken as 0.2. The softmax function is fully used in the classifier's output layer as the activation layer where the probabilities of identifying the input class are effectively achieved.

3.3. Result analysis

Figure 3 shows the detailed results obtained in the form of precision, recall and f1-score for every class label of drug entity using the proposed model. The model has performed well with F1-score value of more than 85% in categorizing the drug entities except the drug_n class label. This may be due to the fact that less number of data is available in the training dataset to learn drug_n class label. However, this could be ignored as larger part of the corpus consisting of drug, group and brand labels have been classified efficiently.

Since it is necessary to compute the overall performance of all the entity classes, analysis has been carried out based on the performance metrics shown in (9)-(11) to contrast the proposed model with the existing DL based DNER model (LSTM-conditional random field (CRF)) [9] as well as other models from the DDI2013 challenge [26], [27]. The results are shown graphically in Figure 4.

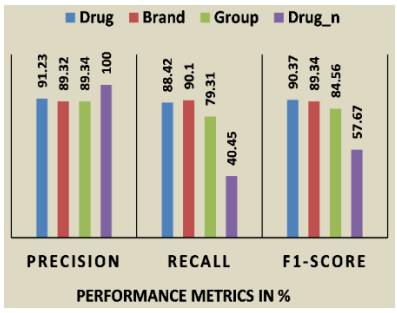

Figure 3. Performance of the proposed model

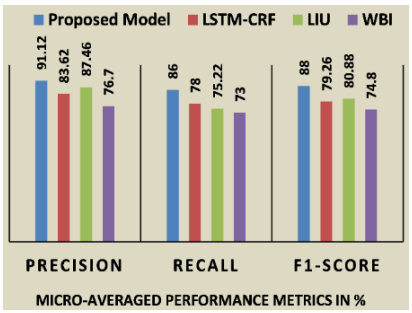

Figure 4. Micro-average performance of proposed model vs Other DNER models

Table 2 shows the results of the performance metrics obtained for each class label for the proposed model as well as other DNER systems. In LSTM-CRF, the features are based on both word and character level embedding. The percentage of improvement of the proposed model over LSTM-CRF with respect to micro-average precision, recall and F1-score are 9.22, 11.19, and 11.17 respectively.

Liu *et al.* [26], have experimented a CRF based model (LIU) with semantic features based on word embedding. On comparison with this system, our proposed model that uses sentence embedding in stacked Bi-LSTM and residual LSTM has improved the micro-average precision, recall and f1-score by 4.18%, 14.63%, and 8.80% respectively. Rocktäschel *et al.* [27], studied a model that ranked first in the DDI2013 challenge studies the impact of domain specific features using linear chain CRF (WBI) for identifying drug names. While comparing with this model, a significant improvement of 18.8%, 18.12%, and 17.64% respectively for micro-average precision, recall and F1-score is shown by the proposed model.

Based on the above results, it has been observed that the proposed model performs better as the sentence embedding included in the architecture considers the complete sentence for syntax and semantic features unlike word embedding which may ignore some of the character features. Even when character level

embedding is combined with word embedding as in [9], the sentence level embedding performs better in the proposed model in terms of all the micro-averaged performance metrics. In addition, the proposed model shows the power of using Bi-LSTM layers that reads the context back and forth in the sentence and captures the context well rather than using a single LSTM layer. Also, the automatic extraction of features using stacked Bi-LSTM layers in the proposed model is used to recognize entities better than using ML algorithms like CRF as in [26], [27].

In addition to the above results, the performance of the proposed approach is evaluated to address the major concern in the DNER field in identifying the n-gram drug entities where n>1 and the results are shown in Table 3. The results are promising in identifying the 2-gram and 3-gram drug entities. However, the results can be further improved.

Table 2. Comparison of performance metrics-proposed model vs other DNER systems

Class Label	Proposed Model			LSTM-CRF			LIU			WBI		
	Pr	Re	Fs	Pr	Re	Fs	Pr	Re	Fs	Pr	Re	Fs
Drug	91.23	88.42	90.37	85.78	80.86	82.59	92.34	85.67	89.54	74.3	85.59	79.32
Brand	89.32	90.10	89.34	88.22	77.83	82.14	100	95.32	97.21	81.27	86.71	84.77
Group	89.34	79.31	84.56	86.43	89.29	87.9	89.42	82.49	86.1	79.4	76.22	78.67
Drug_n	100	40.45	57.67	78.21	57.64	63.48	89.39	14.56	24.75	31.02	90.41	14.2
Micro-Average	91.12	86.00	88.00	83.62	78.00	79.26	87.46	75.22	80.88	76.7	73.00	74.8

Table 3. Percentage of n-gram entities recognized using the proposed model

Type of n-gram entity	% Recognized
2-gram	83.89%
3-gram	76.67%
4-gram	40%

4. CONCLUSION

In this paper, we have proposed a novel DNER architecture using the latest and advanced DL models. It includes stacked Bi-LSTM and a residual LSTM layers. The architecture takes the input in the form of vector from sentence level embedding model and outputs the desired drug label sequence with BIO tagging scheme. We conducted experiments using DDI2013 drugbank dataset. Our proposed model has achieved higher performance than the results obtained using the same dataset with previous state-of-the-art models. Besides, the proposed model has shown good results in recognizing 2- and 3- gram entities. The future research may be oriented towards further improving the performance using other latest embedding techniques and context aware DL architectures.

REFERENCES

[1] L. He, Z. Yang, H. Lin, and Y. Li, "Drug name recognition in biomedical texts: a machine-learning-based method," *Drug Discov. Today*, vol. 19, no. 5, pp. 610-617, May 2014, doi: 10.1016/j.drudis.2013.10.006.

[2] M. Cho, J. Ha, C. Park, and S. Park, "Combinatorial feature embedding based on CNN and LSTM for biomedical named entity recognition," *J. Biomed. Inform*, vol. 103, Mar. 2020, doi: 10.1016/j.jbi.2020.103381.

[3] M. Gridach, "Character-level neural network for biomedical named entity recognition," *J. Biomed. Inform*, vol. 70, pp. 85-91, Jun. 2017, doi: 10.1016/j.jbi.2017.05.002.

[4] B. Bhasuran, G. Murugesan, S. Abdulkadhar, and J. Natarajan, "Stacked ensemble combined with fuzzy matching for biomedical named entity recognition of diseases," *J. Biomed. Inform*, vol. 64, pp. 1-9, Dec. 2016, doi: 10.1016/j.jbi.2016.09.009.

[5] X. Li, H. Zhang, and X. H. Zhou, "Chinese clinical named entity recognition with variant neural structures based on BERT methods," *J. Biomed. Inform*, vol. 107, Jul. 2020, doi: 10.1016/j.jbi.2020.103422.

[6] A. Bashar, "Survey on Evolving Deep Learning Neural Network Architectures," *J. Artif. Intell. Capsul. Networks*, vol. 1, no. 2, pp. 73-82, Dec. 2019, doi: 10.36548/jaicn.2019.2.003.

[7] J. Li, A. Sun, J. Han, and C. Li, "A Survey on Deep Learning for Named Entity Recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, Mar. 2020, doi: 10.1109/TKDE.2020.2981314.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, May. 2015, doi: 10.1038/nature14539.

[9] D. Zeng, C. Sun, L. Lin, and B. Liu, "LSTM-CRF for Drug-Named Entity Recognition," *Entropy*, vol. 19, no. 6, pp. 283, Jun. 2017, doi: 10.3390/e19060283.

[10] A. G. Agirre, M. Marimon, A. Intxaurrondo, O. Rabal, M. Villegas, and M. Krallinger, "PharmaCoNER: Pharmacological Substances, Compounds and proteins Named Entity Recognition track," in *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, Nov. 2019, pp. 1–10, doi: 10.18653/v1/d19-5701.

[11] M. E. Peters *et al.*, "Deep contextualized word representations," *arXiv Prepr. arXiv1802.05365*, Feb. 2018.

[12] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," *arXiv preprint arXiv:1801.02143*, Jan. 2018.
[13] J. Kim, M. El-Khamy, and J. Lee, "Residual LSTM: Design of a deep recurrent architecture for distant speech recognition," *arXiv Prepr. arXiv1701.03360*, Jan. 2017.
[14] Y. Wang, X. Zhang, M. Lu, H. Wang, and Y. Choe, "Attention augmentation with multi-residual in bidirectional LSTM," *Neurocomputing*, vol. 385, pp. 340-347, Apr. 2020, doi: 10.1016/j.neucom.2019.10.068.
[15] Ş. Öztürk and U. Özkaya, "Residual LSTM layered CNN for classification of gastrointestinal tract diseases," *J. Biomed. Inform.*, vol. 113, pp. 103638, Jan. 2021, doi: 10.1016/j.jbi.2020.103638.
[16] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, Oct. 2014, doi: 10.3115/v1/d14-1162.
[17] Z. Liu *et al.*, "Entity recognition from clinical texts via recurrent neural network," *BMC Med. Inform. Decis. Mak.*, vol. 17, no. 2, pp. 53-61, Jul. 2017, doi: 10.1186/S12911-017-0468-7.
[18] C. Wang, H. Yang, and C. Meinel, "Image Captioning with Deep Bidirectional LSTMs and Multi-Task Learning," *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 14, no. 2s, pp. 1-20, Apr. 2018, doi: 10.1145/3115432.
[19] A. Sniegula, A. P. Mararida, and L. Chomatek, "Study of named entity recognition methods in biomedical field," in *Procedia Computer Science*, vol. 160, pp. 260-265, Jan. 2019, doi: 10.1016/j.procs.2019.09.466.
[20] M. H. Zazo, I. S. Bedmar, P. Martinez, and T. Declerck, "The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions," *J. Biomed. Inform.*, vol. 46, no. 5, pp. 914–920, Oct. 2013, doi: 10.1016/j.jbi.2013.07.011.
[21] A. Kutuzov and E. Kuzmenko, "To lemmatize or not to lemmatize: how word normalisation affects ELMo performance in word sense disambiguation," in *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, Sep. 2019, pp. 22-28.
[22] A. Graves, N. Jaitly, and A. R. Mohamed, "Hybrid speech recognition with Deep Bidirectional LSTM," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2013-Proceedings*, Dec. 2013, pp. 273–278, doi: 10.1109/ASRU.2013.6707742.
[23] A. Graves and J. Schmidhuber, "Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602-610, Jul. 2005, doi: 10.1016/j.neunet.2005.06.042.
[24] T. Liu, S. Yu, B. Xu, and H. Yin, "Recurrent networks with attention and convolutional networks for sentence representation and classification," *Appl. Intell.*, vol. 48, no. 10, pp. 3797–3806, Oct. 2018, doi: 10.1007/s10489-018-1176-4.
[25] V. Van Asch, "Macro-and micro-averaged evaluation measures," *Belgium: CLiPS*, pp. 1–27, Sep. 2013.
[26] S. Liu, B. Tang, Q. Chen, X. Wang, Y. Yu, and Y. Wang, "Effects of Semantic Features on Machine Learning-Based Drug Name Recognition Systems: Word Embeddings vs. Manually Constructed Dictionaries," *Information*, vol. 6, no. 4, pp. 848–865, Dec. 2015, doi: 10.3390/info6040848.
[27] T. Rocktäschel, T. Huber, M. Weidlich, and U. Leser, "WBI-NER: The impact of domain-specific features on the performance of identifying and classifying mentions of drugs," in *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, vol. 2, Jun. 2013, pp. 356-363.

BIOGRAPHIES OF AUTHORS

T. Mathu is an Assistant Professor in the Department of Computer Science and Engineering at Karunya Institute of Technology and Sciences, Coimbatore, India. She is also pursuing her Ph.D degree at the same university. Her research interests include data mining, text mining, natural language processing, machine learning and deep learning.

Kumudha Raimond is a Professor in the Department of Computer Science and Engineering at Karunya Institute of Technology and Sciences, Coimbatore, India. Her areas of expertise include machine learning, intelligent systems, biometrics, bioinformatics, biomedical applications, satellite image processing, watermarking, compression and wireless sensor networks.