A Study on Named Entity Recognition with Different Word Embeddings on GMB Dataset using Deep Learning Pipelines

Surya Teja Chavali, Charan Tej Kandavalli, Sugash T M, Deepa Gupta
Department of Computer Science and Engineering, Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, India.
bl.en.u4aie19014@bl.students.amrita.edu, bl.en.u4aie19013@bl.students.amrita.edu, bl.en.u4aie19062@bl.students.amrita.edu,
g_deepa@blr.amrita.edu

Abstract— **Many natural language applications, such as QA tools, information retrieval, text summarization, and machine translation, are built on top of Named Entity Recognition (NER). The primary task of Named Entity Recognition is to classify all objects (Named Entities) in a given text sample into specified classes such as person, location, and organization names. This work aims to create a NER model for the popular Groningen Meaning Bank (GMB) dataset, comprising tens of thousands of unprocessed and tokenized texts. The dataset has been published by the University of Groningen and updated by developers at IBM in the year 2020. The proposed system is designed into 7 different models – passing the embedding vectors created by keras embedding (Python), BERT, Glove, Word2Vec, fastText, and other two involving Character Embedding along with the Attention layer into a BiLSTM-CRF network. These techniques were employed to compare and analyze the model performances against each other for a better understanding.**

Keywords — NER, GMB, BiLSTM-CRF, BERT, Glove, Word2Vec, fastText, Attention

I. Introduction

There has been an increasing demand for Natural Language Processing applications (NLP) in businesses and big technological firms [1], due to the ever-increasing customer database, and it is important for developers and retailers to know the satisfaction and loyalty levels of their customers. Analysing the result of this task can easily change the company's market position. NLP is a subfield of computer science that combines linguistics and machine intelligence to enable Human-Computer Interaction (HCI). Linguistic ability and knowledge are one of the most essential features of humans since it reveals information about how the brain works. In natural language processing, named entity recognition (NER) [2],[3] is an integral part of the development information extraction job. NER tries to recognize words like person, organization, business, names of places, temporal phrases, and economies in a written text using pre-established or published information.

Let us now discuss some of the ambiguities faced by a machine in NER. The category description for a person is conceptually pretty apparent, but there is considerable difficulty in classification for computers. For example, consider the sentences – "France (Organisation) has won the 2018 FIFA world cup (vs) The 2018 world cup took place in France (Location)" and "Washington (Location) is USA's capital (vs) Washington (Person) was USA's first president". We observe that two same words have been classified into different entities. One way of solving this kind of problem statement is to use different machine learning techniques to build the model for multi-class classification, but this takes a huge amount of labelling. To deal with the uncertainty of the statements, the approach requires a sophisticated awareness of the context in addition to labelling. This makes it a difficult task for a traditional machine learning model. Another option is to utilize a conditional random field (CRF), which both NLP Audio Tagger and the natural language toolkit (NLTK) [4] provide. It is a way of estimating for modelling sequential information, such as words and phrases. The CRF can obtain a thorough grasp of the sentence's context. Learning at a deeper level is required for a model to formulate, because it can build words. As a result, deep learning-based NER [5] frameworks are far more precise than the previous method. This is because they used a technique known as word embedding, as it can deduce the semantic and syntactic relationships between different words.

Generalized entities (e.g., person and place) and domain-specific entities are the two basic types of names entities (e.g., proteins, enzymes, and genes) in NER applications. This paper deals with generalized entities.

- The work proposed in this paper comprises a total of 7 NER models; a traditional word embedding along with BiLSTM and BiLSTM-CRF networks.
- Following that, we developed models of BERT, Glove, Word2Vec, fastText word embeddings using a BiLSTM-CRF layer to encode.
- Finally, a Character Embedding along with a BiLSTM-Attention layer has been employed to enhance the NER tag prediction.

The structuring of this paper is as follows: In section II we specify the existing works on NER applications based on various datasets. We present the GMB Dataset, and discuss the working and architectures of the Word Embedding models used. Following this, we detail several experimental results and analyse their outcomes. Finally, we conclude the work and describe the future scope of the work presented.

II. Related Works

Researchers in the recent times have developed applications of NER, which have shown great success in classical CNN-CRF or the LSTM-CRF architecture.

Bi-directional LSTMs with a combination of CRF layers have been first introduced in [6] by, Panchendrarajan, Rrubaa & Amaresan, Aravindh. They have experimented with the CRF to show that it can extract the dependencies easily from both directions of a given sentence. Similar work has been carried out in [7], where the authors (Yang, Gang & Xu) have presented a nice workflow of the architecture of Residual BiLSTM networks, which enable the effective learning of deep networks with many LSTM layers by adding an extra spatial detour path from lower levels.

To make this comparable to previous studies, we made use of a set of word vectors generated from the word embeddings –

BERT, Glove, Word2Vec, and fastText, separately trained them on a BiLSTM-CRF model specifically for the sentences on GMB, and this significance of sentences in the dataset in discussed more in Section III. The authors have discussed the idea and implementation of Semi-Supervised BiLSTM-CRF models in [8]. They have solved a problem where there is a lack of labelled data, which makes it harder for deep neural networks that use labelled data to function well to achieve excellent accuracy, using SCRNER, a semi-supervised deep learning model on very little labelled data, and a large chunk of unlabelled data. The authors in [9] have developed a novel deep NN-based BiLSTM to solve the NER problem on a Bio-medical data chunk. Their model outperforms the CRF and BiLSTM method with only word-level embeddings.

Many other researchers have experimented with different approaches to tackle the NER problem, such as in [10], where the authors have developed a discrete and a continuous model to extract specific features by fixing the model configuration of Skip Gram, with pre-defined hyper-parameters, based on its performance compared to others. CRF applications have been extended to other native Indian languages such as Telugu, Tamil, etc. In [11], the authors talk briefly about the development of their Hybrid model of rule-based and machine-learning approaches to train the model on word handlers and finally pass the data into a CRF layer for optimal tag prediction. Researchers in [12] proposed an NER model based on word position, which embeds directly the word position of each word, character by character into a word vector for further processing. The authors used a special kind of CNN model known as a Dilated CNN (IDCNN) [13] in order to build the deep learning model to carry out the NER task. The authors in [14] have proposed a similar approach of what has been discussed in this paper, where they have proved that BERT-BiLSTM-CRF model outperformed. The authors in [15] have used Tree LSTMs and TNNs [16] to encode information and later perform entity disambiguation, and a similar kind of methodology has been proposed in [17] and [18], but instead used a Tree-LSTM-CNN architecture to perform the encoding task.

Considering these developments, we have taken a prominent and a non-trivial dataset (GMB), to better understand the Word Embedding performances when random sentences with diverse POS and NER tag sets are used (described in Section III). Also, a majority of the recently developed works lack an analysis on several Generic NER tags such as Person, Time Indicators, Artifacts, Natural Phenomenon, etc. With this work, we aim to address this research gap.

III. Dataset Description

The Groningen Meaning Bank (GMB) dataset was developed by the University of Groningen in 2019 and was later put up by IBM on their developers' page[1]. The dataset contains sentences from headlines where every other word has been assigned to a specific entity – a geographical entity, or an organization. Syllables and words that are not significant at all are labeled with the background class; O and the entire data is divided into a total of 50,00 sentence.

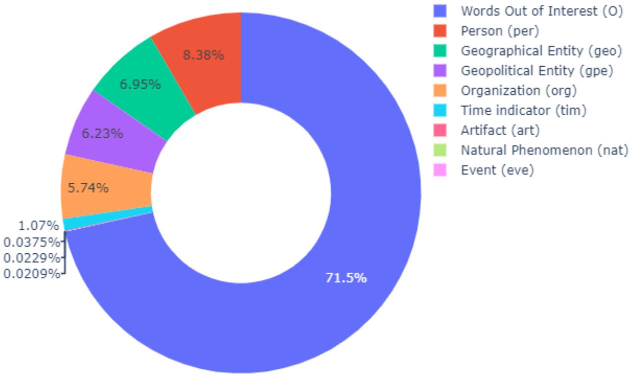
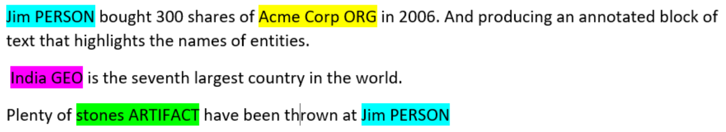


Fig. 1. NER Tag distribution across the entire dataset



Jim PERSON bought 300 shares of Acme Corp ORG in 2006. And producing an annotated block of text that highlights the names of entities.

India GEO is the seventh largest country in the world.

Plenty of stones ARTIFACT have been thrown at Jim PERSON

Fig. 2. A sample of NER Tag distribution of the data

[19] discussed the brief modifications done to the originally published data. The initial set of changes they made ensured that the dataset modifies entity labels such that they will no longer directly correspond to the identifiers (name class) in the dataset. The entities are classified according to the standard BIO/IOB (beginning, inside, outside) scheme [20], which prefixes each entity label with either a B or an I letter. The words that aren't relevant are marked with an O. Therefore, this is a multi-classification problem. Under this scheme, we have a total of 17 NER tags – a same set of 8 tags under B and I, and 1 tag is O. Below is the essential info about the main 8 entities in the GMB Dataset (each of them for both B and I) as visualized in Figure 1.

IV. System Architecture

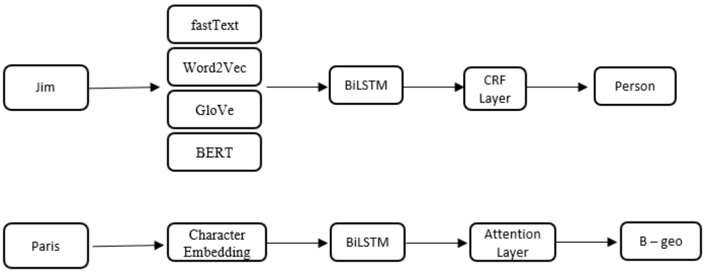


Fig. 3. Generalized System Architecture of the proposed models

The design of our proposed system is divided into three sections; one for building the ontology of pre-trained word embeddings, the second sub-section for detailing the architecture of BiLSTM encoder and the finally describing the CRF classifier and the Attention layer.

[1] https://developer.ibm.com/exchanges/data/all/groningen-meaning-bank/

A. Word Embeddings

- Keras Embedding: The Keras API has an Embedding class which requires the input data be given in an integer encoded format. It accepts, 3 inputs – input dimensions, output dimensions and the input of the data used for training.
- BERT Embedding: BERT [21] considers a word's extensive left- and right-hand contexts. BERT also manages semantic relationships using word-piece tokenization.
- GloVe Embedding: GloVe [22] is concerned with word co-occurrences throughout the entire corpus and the likelihood that two words will appear together are related by its embeddings. In this work, the glove – twitter pre-trained vectors have been used to in order to create the word vector matrix.
- Word2Vec Embedding: Word2Vec is a two-layer neural network called Word2vec interprets text by vectorizing words and the concept of interdependency across words or components, such as semantic relatedness, is extracted using this embedding. While employing Word2Vec from genism in this work, a window size of 5 has been considered, which conveys the optimal separation possible among the current word and its nearby words.
- fastText Embedding: FastText uses the skip-gram paradigm, in which each word is represented by a collection of character n-grams, through which it can embeddings for words that are misspelt, uncommon, or that are absent from the training sample. A total of 4 worker threads along with a window size of 5 has been taken in account in building the word vector matrices.
- Character Embedding: Character-level embedding utilizes a 1D-CNN network (which is regarded as a scanner) to find numeric word representations by examining at the character-level compositions dependency and composition of words. These scanners have the ability to focus on many characters at once. As they move along, these scanners extract data from the characters they are concentrating on. At the conclusion of the scanning process, data from various scanners are combined to generate a word representation.

B. BiLSTM Encoders

The Bi – directional LSTM layer takes in a recurrent layer as a parameter (for example, the first LSTM layer [23],[24]). The output from the preceding embedding layer is used in this layer. Since this is bidirectional LSTM, we will have backward and forward outputs. By adding up, taking the average, concatenating, or multiplying these outputs, one can consolidate them before sending towards the succeeding layer. These methods can be found in the BiLSTM layer's merging mode argument. In the standard mode, the results are concatenated, increasing the quantity of outputs to the following layer.
Since this layer is also dealing with a Many to Many RNN architecture, it indicates that the final output would be from each input sequence. For instance, in the sequence ($a_1b_1, a_2b_2, \dots\dots, a_nb_n$), a and b represent the sequence's inputs and outputs, respectively. Therefore, next layer comprises of the TimeDistributeDense layer, which allow for dense (fully-connected) operation across all outputs and time steps. If you don't use this layer, you'll only get one final output.

C. CRF and Attention Layers

The original text is tagged with the BiLSTM layer and the CRF layer [25], and then the predicted word segmentation results are obtained. Finally, the supervised learning method is utilized to iteratively learn the word segmentation results, allowing the model's performance to be improved and accurate results to be obtained.

The function of the following CRF linear layer is to map the hidden state vector from n-dimension to k-dimension, where k is the number of labels defined in the tagging scheme. As a result, the sentence features are extracted that are represented as a matrix P = (p_1, p_2, p_3, …, p_n). The parameters of the CRF layer are represented by a matrix A $\in$R.

In the case of a Character Embedding, we add an extra Attention layer at the end to overcome the limitations posed by the BiLSTM-CRF-CNN network in some cases. [26] descries about how the attention layer solves the contradictions raised by the XOR Limitation of BiLSTM-CRF. They considered four operations forming bit-wise XOR and discuss how a BiLSTM-CRF network shows ambiguity in weight updating for some cases. Therefore, the attention – a combination of SoftMax and Sigmoid layers can build a solution to this contradiction. Since character embedding is nothing but a 1D convolution operation happening on each word vector, we make use of this Attention layer, so that the model maintains disambiguity in the training phase.

V. Experimental Results and Analyses

A. Implementation Environment

The entire development of the models employed in this research have been executed on the Google Collaboratory Regular version, using Python 3 as the programming language, through which many existing tools and libraries such scikit_learn, tensorflow (keras), genism, etc., have been taken into account.

An overview of the configurable parameters employed while developing the models has been detailed in table II.

TABLE II
HYPER – PARAMETER VALUES

TensorFlow version	1.15.2
Embedding Matrix Dimension	100
BiLSTM Units	50
Recurrent dropouts in BiLSTM	0.1
Dense Layer Units	50
Activation Function	Relu
CRF Model	Viterbi
Optimizer	RMSprop
Test Size	20%
Window (Word2Vec)	5
Minimum Count (Word2Vec)	1
Minimum Count (fastText)	1
Skip-Gram(fastText)	1
BiLSTM units (Attention Layer)	16
Activation function (Attention Layer)	Relu, Tanh, SoftMax

B. Evaluation of Models

TABLE III
EVALUATION METRICS OF THE MODELS

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 Score (%)**
fastText + BiLSTM + CRF	96.99	23.0	21.0	20
Word2Vec + BiLSTM + CRF	98.86	41.0	38.0	38
BiLSTM + CRF	99.53	63.0	49.0	50
GloVe + BiLSTM + CRF	99.3	59.0	48.0	50
BiLSTM	98.8	68.0	52.0	55
BERT + BiLSTM + CRF	99.71	66.0	51.0	56
Character + BiLSTM + Attention	**99.25**	**62.0**	**57.0**	**58**

We train the BiLSTM CRF on 5 different word embeddings – Keras Embedding, BERT, GloVe, Word2Vec, fastText, and Character Embedding. A detailing about the comparison of the performances delivered by all 7 models including these models can been inferred from Table III.

The models performed extremely well and achieved an overall accuracy of over 95 %. We observe that GloVe, Word2Vec, and fastText were good competitors for BERT based BiLSTM model. While BERT's performance was the best among all Embeddings in terms of accuracy, the character–attention model showed a much more promising result in terms of F1 metrics. In general, the F1 score balances precision and recall on non-negative observations, whereas accuracy focuses on successfully identifying positive and negative inferences.

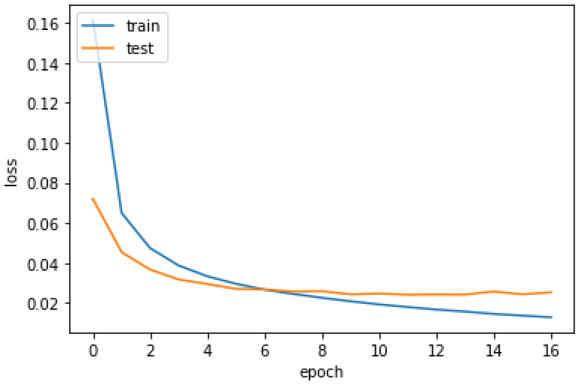


Fig. 4. Training (vs) Validation Loss of the Character + Bi - LSTM + Attention model

TABLE IV
CLASSIFICATION REPORT OF THE BEST MODEL

NER Tag	**Precision**	**Recall**	**F1 Score**
B-art	0.33	0.03	0.06
B-eve	0.53	0.34	0.41
B-geo	0.85	0.91	0.88
B-gpe	0.98	0.92	0.95
B-nat	0.33	0.13	0.19
B-org	0.77	0.69	0.73
B-per	0.81	0.82	0.81
B-tim	0.93	0.86	0.89
I-art	0.67	0.06	0.11
I-eve	0.48	0.28	0.35
I-geo	0.82	0.77	0.79
I-gpe	0.73	0.26	0.38
I-nat	0.33	0.06	0.11
I-org	0.78	0.75	0.77
I-per	0.82	0.90	0.86
I-tim	0.78	0.68	0.73
O	0.99	0.99	0.99
Accuracy			0.97
Macro Average	0.70	0.56	0.59
Weighted Average	0.97	0.97	0.97

Conventionally, BERT takes an entire sentence (word sequence) as input, and Word2Vec and fastText take only a single word at each time step while generating word vectors, which might be a reason due to a slight variation in the accuracies. Figure 3 shows the training (vs) validation loss across all epochs.

From Table IV, we infer that the F1 scores of certain entity tag sets such as Natural Phenomenon (nat), Artifacts (art), and Geopolitical Entity (I-gpe) are very low, whereas, that of Geographical Entity (B-gpe), Time Indicators (tim), Organization (org), Person (per) and Words out of Interest (O) were very good. This can be understood from figure 1, which emphasizes on the distribution of entity tags across the entire dataset. Since the tag 'O' is occurring the maximum number of times (71.5%), its F1 score has turned out to be the maximum. Furthermore, since ‘nat’ and ‘art’ make up <0.1% of the data, their scores have been poor.

VI. Conclusion

We have successfully established and implemented different word embedding techniques along with BiLSTM-CRF networks on the GMB Dataset. In conclusion, we hope that NER performed on this dataset is going to be of good help in developing large-scale applications, because of the diversity of the sentences. A total of 1000000 words have been taken into consideration, on which the development of the Character Embedding based model has been very robust.

The current work can still be extended, by implementing several different word embeddings such as Capitalization, TF–IDF [27], Paragraph word vectors, ELMo [28] and through some popular topic modelling techniques such as in [29] to better understand the effects of different tag sets posed on the Word Embedding models.

References

[1] Z. Wu, M. Galley, C. Brockett, Y. Zhang, and B. Dolan, “Automatic Document Sketching: Generating Drafts from Analogous Texts,” May 2021. [Online]. Available: https://www.microsoft.com/en-us/research/publication/automatic-document-sketching-generating-drafts-from-analogous-texts/

[2] A. D. Cohen et al., “LaMDA: Language Models for Dialog Applications,” in arXiv, 2022.

[3] B. Mohit, “Named Entity Recognition,” in Natural Language Processing of Semitic Languages, I. Zitouni, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 221–245. doi: 10.1007/978-3-642-45358-8_7.

[4] E. Loper and S. Bird, “NLTK: The Natural Language Toolkit,” in Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1, 2002, pp. 63–70. doi: 10.3115/1118108.1118117.

[5] 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. 34, no. 1, pp. 50–70, 2020.

[6] Panchendrarajan, Rrubaa & Amaresan, Aravindh. (2019). Bidirectional LSTM-CRF for Named Entity Recognition.

[7] Yang, Gang & Xu, Hongzhe. (2020). A Residual BiLSTM Model for Named Entity Recognition. IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.3046253.

[8] M. Zhang, G. Geng, and J. Chen, “Semi-Supervised Bidirectional Long Short-Term Memory and Conditional Random Fields Model for Named-Entity Recognition Using Embeddings from Language Models Representations,” Entropy, vol. 22, no. 2, p. 252, Feb. 2020, doi: 10.3390/e22020252.

[9] Saad, F., Aras, H., Hackl-Sommer, R. (2020). Improving Named Entity Recognition for Biomedical and Patent Data Using Bi-LSTM Deep Neural Network Models. In: Métais, E., Meziane, F., Horacek, H., Cimiano, P. (eds) Natural Language Processing and Information Systems. NLDB 2020. Lecture Notes in Computer Science(), vol 12089. Springer, Cham. https://doi.org/10.1007/978-3-030-51310-8_3M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[10] B. Ertopçu et al., "A new approach for named entity recognition," 2017 International Conference on Computer Science and Engineering (UBMK), 2017, pp. 474-479, doi: 10.1109/UBMK.2017.8093439.

[11] M. H. Khanam, M. A. Khudhus and M. S. P. Babu, "Named Entity Recognition using Machine learning techniques for Telugu language," 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2016, pp. 940-944, doi: 10.1109/ICSESS.2016.7883220.

[12] Y. Du and W. Zhao, "Named Entity Recognition Method with Word Position," 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI), 2020, pp. 154-159, doi: 10.1109/IWECAI50956.2020.00038.

[13] X. Lei, H. Pan and X. Huang, "A Dilated CNN Model for Image Classification," in IEEE Access, vol. 7, pp. 124087-124095, 2019, doi: 10.1109/ACCESS.2019.2927169.

[14] Z. Dai, X. Wang, P. Ni, Y. Li, G. Li and X. Bai, "Named Entity Recognition Using BERT BiLSTM CRF for Chinese Electronic Health Records," 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2019, pp. 1-5, doi: 10.1109/CISP-BMEI48845.2019.8965823.

[15] Q. Wang and M. Iwaihara, "Deep Neural Architectures for Joint Named Entity Recognition and Disambiguation," 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), 2019, pp. 1-4, doi: 10.1109/BIGCOMP.2019.8679233.

[16] K. S. Tai, R. Socher, and C. D. Manning, “Improved semantic representations from tree-structured long short-term memory networks,” in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, pp. 1556–1566.

[17] Eliyahu Kiperwasser, Yoav Goldberg; Easy-First Dependency Parsing with Hierarchical Tree LSTMs. Transactions of the Association for Computational Linguistics 2016; 4 445–461. doi: https://doi.org/10.1162/tacl_a_00110

[18] J. Wang, L. -C. Yu, K. R. Lai and X. Zhang, "Tree-Structured Regional CNN-LSTM Model for Dimensional Sentiment Analysis," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 581-591, 2020, doi: 10.1109/TASLP.2019.2959251.

[19] A. Abid and J. Zou, “Improving Training on Noisy Stuctured Labels,” arXiv [cs.LG], 2020.

[20] N. Alshammari and S. Alanazi, “The impact of using different annotation schemes on named entity recognition,” Egypt. Inform. J., vol. 22, no. 3, pp. 295–302, 2021.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional Transformers for language understanding,” arXiv [cs.CL], 2018.

[22] J. Pennington, R. Socher, and C. 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.

[23] A. Gopalakrishnan, K. P. Soman, and B. Premjith, “A deep learning-based named entity recognition in biomedical domain,” in Lecture Notes in Electrical Engineering, Singapore: Springer Singapore, 2019, pp. 517–526.

[24] V. Hariharan, M. Anand Kumar, and K. P. Soman, “Named entity recognition in Tamil language using recurrent based sequence model,” in Innovations in Computer Science and Engineering, Singapore: Springer Singapore, 2019, pp. 91–99.

[25] G. Veena, D. Gupta, S. Lakshmi, and J. T. Jacob, “Named entity recognition in text documents using a modified conditional random field,” in Advances in Intelligent Systems and Computing, Singapore: Springer Singapore, 2018, pp. 31–41.

[26] P.-H. Li, T.-J. Fu, and W.-Y. Ma, “Why Attention? Analyze BiLSTM Deficiency and Its Remedies in the Case of NER”, AAAI, vol. 34, no. 05, pp. 8236-8244, Apr. 2020.

[27] L. Sravani, A. S. Reddy and S. Thara, "A Comparison Study of Word Embedding for Detecting Named Entities of Code-Mixed Data in Indian Language," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 2375-2381, doi: 10.1109/ICACCI.2018.8554918.

[28] M. E. Peters et al., “Deep contextualized word representations,” in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 2227–2237.

[29] V. Gangadharan and D. Gupta, “Recognizing Named Entities in Agriculture Documents using LDA based Topic Modelling Techniques,” Procedia Comput. Sci., vol. 171, pp. 1337–1345, 2020.