Named Entity Recognition using Bi-LSTM and Tenserflow Model

N V Sahana[1], Prof. Bhanushree K J[2]

[1]Associate Software Engineer, Bosch Global Software Technologies, Electronic City, Bengaluru, Karnataka, India
[2]Professor, Dept. of Computer Science & Engineering, Bangalore Institute of Technology, Bengaluru, India

---***---

Abstract - *Named entity recognition (NER) is a difficult task that has conventionally needed enormous amount of knowledge in the form of lexicons and feature engineering to achieve good performance. In the past, Named Entity Recognition systems were able to achieve great success in performing well with the cost of humankind engineering in designing domain-specific features and rules. In this paper, we propose a recurrent neural network architecture based on the variant of RNN called LSTM with four layers to perform the task of named entity recognition. The proposed model has five steps; dataset collection, data preprocessing, sequential data extraction, building the model, fitting the model and analyzing results. This model will classify the texts by understanding the meaning of them or context of sentences using the bidirectional LSTM without the need to remove stop words. With the proposed model an accuracy of 96.89% was achieved.*

Key Words**: *NER, LSTM, Keras, RNN, Tenserflow***

1. INTRODUCTION

Named Entity Recognition is considered as one of the first and key steps in the Natural Language Processing pipeline, it is also a task in information extraction that looks to identify and categories named entities mentioned in input textual content into some predefined categories such as organizations, locations, time expressions, quantities, monetary values, medical codes, percentages, person names etc. In recent years, the widespread and proliferated use of social media and other media catering to all kinds of people has resulted in increased unstructured cyber data and Named Entity Recognition is considered to be the instigating step to convert this unstructured format of data into structured format which is of use to lot of different applications. To understand NER formally, assume a sequence of tokens, *sequence=<w1, w2, w3........wn>*. The task of performing Named Entity Recognition is to output a list of tuples [1], *<Is, Ie, ti,>*, where each one of them is a named entity that is mentioned in s. Here, *Is* and *Ie* are in the range represented by [1, N], which represents the start index and end index of the mentioned named entity. *ti* is an entity type from the predefined set of categories.

This concept is discussed in detail in the following section.

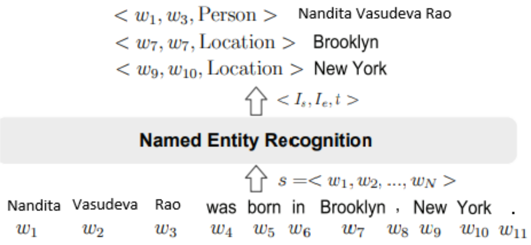

$< w_1, w_3, \text{Person} >$ Nandita Vasudeva Rao
$< w_7, w_7, \text{Location} >$ Brooklyn
$< w_9, w_{10}, \text{Location} >$ New York
$< I_s, I_e, t >$
Named Entity Recognition
$s = < w_1, w_2, ..., w_N >$
Nandita Vasudeva Rao was born in Brooklyn , New York .
w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8 w_9 w_{10} w_{11}

Fig-1: The concept of Named Entity Recognition

Figure 1 shown above presents an example of the task of named entity recognition. Here the NER system is fed with a sequential arrangement of words that forms a sentence. The first token is identified as a named entity and that entity is a person. Hence, the tag associated with the first token is 'person'. Similarly the second and third tokens when fed to the NER system will be recognized as person ('person' is the entity). The next token in the sequence is 'was', which is not a named entity. The next named entity recognized by the NER system would be 'Brooklyn' which is of the entity type 'Geo-Location'.

In the conventional domain of information extraction, two of the innovations resulted in notable enhancements. First, word embeddings are considered and used to present every token by a vector of low dimensions, which provides the frequencies of the adjacent tokens that are co-occurring. When it is juxtaposed with the bag-of-words approach i.e. the basis of the general methods, word embeddings do seize semantic similarities with the tokens that cannot be clearly figured out from their surface. The idea behind illustrating words is that the company those words keep is an ancient definition in linguistics. Its sudden popularity is due to the fact that the embeddings of the words are automatically tuned such that tools of information extraction leverage the best out of it. Second, it has been shown that the utilization of neural networks, which has the tendency of automatically learning features of non-linear combinations, results in enhanced identification than the utilization of CRFs, which have the tendency to only learn features of linear combinations[5]. Deep neural networks, specifically, long short-term memory networks (LSTMs), have the ability to do this task very effectively and efficiently [5]. The proposed method uses the LSTM to find internal features in the sequences of words eliminating the necessity of using stop words for the purpose.

The rest of the paper is organized as follows: Section II Literature review, Section III presents the proposed architecture, and IV and V explain the experimental settings, results, and discussion. Finally, Section VI discusses the conclusion.

2. LITRATURE REVIEW

Jing L et al. [1] proposed a survey on usage of deep learning in NER, in which they have provided a review on existing techniques of deep learning for the task of NER including discussion about NER resources with off-the-shelf NER tools and NER corpora. They also provide surveys on techniques of deep learning that are most recently used in new NER problem settings and applications.

Yanyao S et al. [2] proposed the CNN-CNN-LSTM model comprising the convolutional character, a long short term memory (LSTM) tag decoder and word encoders. The authors also carried out incremental active learning [2], during the training process, and were able to achieve good performance results.

Y Wu et al. [3] contrasted three varying NER methods; the traditional CRF-based NER approach, a DNN-based NER method that utilizes a arbitrarily initialized matrix of embedded words; and an DNN-based NER method that makes use of ta matrix of embedded words derived from the unlabelled corpus. All three approaches were trained with the training set and their functioning on the test set was documented.

Asim G et al. [4] suggested a successful implementation of Deep Bidirectional Long Short Term Memory (DB-LSTM) which achieved a 93.69% F1 score, considered to be a good result for the task of Named Entity Recognition in Turkish.

Maryam H et al. [5] proposed a model which shows that a purely generic model corresponding to statistical word embeddings [called long short-term memory network-conditional random field (LSTM-CRF)] and deep learning and their proposed model outperformed conventional NER tools that are entity-specific, and by a great margin.

M. Ali Fauzi et al. [6] proposed an approach to perform the task of named entity recognition using a recurrent neural network, known as Long Short-Term Memory. The network proposed is trained to perform two passes on each sequence input, outputting its decisions on the second pass. For performing disambiguation in the second pass, the necessary information is acquired in the first step. For the second pass, the network is trained to output a vector representation of the relevant output tags.

P. V. Q. de Castro et al. [7] proposed a Deep Learning architecture based on Bi-LSTM with CRF and this architecture was evaluated by performing the altering of hyperparameters for Portuguese corpora. The output achieved high performance using the optimal values, enhancing the outputs achieved for Portuguese language to up to 5 points in the F1 score.

Chuanhai d et al. [8] proposed a bidirectional LSTM-CRF neural network that utilizes both radical-level and character-level representations. By comparing the outputs of different variations of LSTM blocks, the authors found the perfectly suitable LSTM block for CNER. They evaluated their system on the third SIGHAN Bakeoff MSRA dataset for a simplified CNER task and achieved a F1 score of 90.95%.

Arya R et al. [9] proposed a review on the learning approaches that have been used for NER in the recent past and also focusing on how they evolved from the linear learning approaches of the past. They also present the progress of related tasks eg. Entity linking, sequence tagging etc. wherever the processes have also enhanced NER outputs.

3. PROPOSED SYSTEM

Figure 2 shows the architecture of our proposed system which performs the task of Named Entity Recognition. The proposed architecture comprises five key steps i.e dataset collection, data preprocessing, sequential data extraction, building model, fitting and analyzing model. These steps are discussed in detail in the following sections.

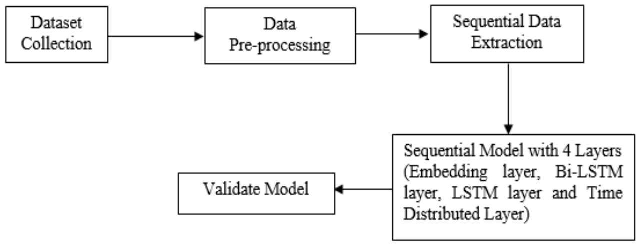

Dataset Collection
Data Pre-processing
Sequential Data Extraction
Sequential Model with 4 Layers (Embedding layer, Bi-LSTM layer, LSTM layer and Time Distributed Layer)
Validate Model

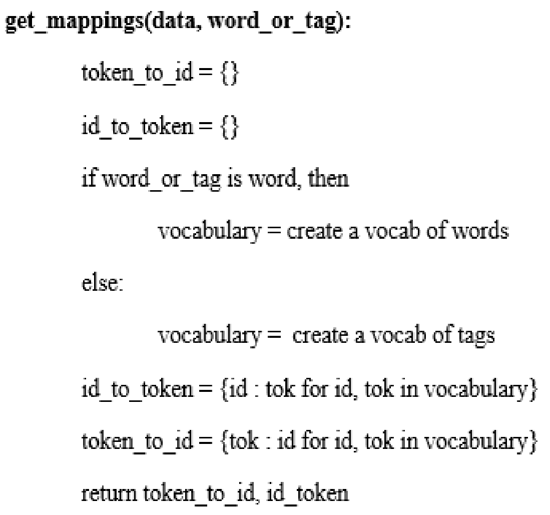
Fig-2: System Architecture of proposed model

3.1 DATASET COLLECTION

The implementation of the proposed model begins by loading the dataset. The dataset used for this model is available publicly on kaggle. The dataset contains sentences, and these sentences are tokenized and saved into a field called words, finally, the tag corresponding to each token is present in the field called tag.

3.2 DATA PREPROCESSING

The proposed model requires us to train a neural network for the task of NER. So we need to perform some modifications in the collected data and prepare it in such a way that it can easily fit into a neutral network. The modifications to data are done in this step which includes extraction of mappings that are required to train the neural network. We will obtain the word and tag mappings and we create two new fields in the dataset which will contain all the mappings.



```
get_mappings(data, word_or_tag):
        token_to_id = {}
        id_to_token = {}
        if word_or_tag is word, then
                vocabulary = create a vocab of words
        else:
                vocabulary =  create a vocab of tags
        id_to_token = {id : tok for id, tok in vocabulary}
        token_to_id = {tok : id for id, tok in vocabulary}
        return token_to_id, id_token
```

Fig-3: Algorithm for extracting the mappings

The algorithm to extract mappings is represented in the figure 3 shown above.

3.3 SEQUENTIAL DATA EXTRACTION

In this step, we will transform the columns in the dataset into sequential arrays. We have to also convert the new tag column created in the previous step into one hot encoding which is leveraged by the model to recognize the named entities.

3.4 BUILDING THE MODEL

Neural network models have the tendency to work only with graphical structure. Hence,, we first outline the structure of the network and set the dimensions of input and output with respect to each layer. RNNs are capable of working well with varying combinations of input and output. We have used the RNN many-to-many architecture to perform this task. The goal is to output a tag for any given word consumed at all time steps. In the proposed neural network, we are functioning with primarily three layers; embedding, bi-LSTM and LSTM layers and the last timeDistributed layer is used for output production.

The architecture as mentioned before has four layers and these layers have specific input and output dimensions which depending on the dataset vacillates. In accordance with the dataset collected in this research, we have set the input and output dimensions.

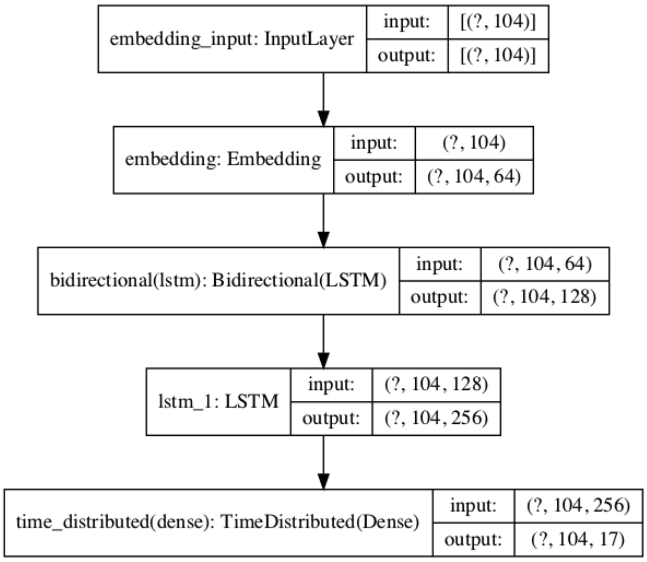


Fig-4: The proposed architecture

The architecture/model with the four layers and its associated input and output dimensions as presented in the Figure 4 shown above.

These layers are explained in detail in the subsequent sections.

- **The embedding layer**: We define the input to be the maximum length of the longest sequence. Once the proposed network is trained, this particular layer will transform each and every token into a n-dimensional vector. For this proposed architecture, we have defined the n-dimension as 64.
- **Bidirectional LSTM**: Bidirectional LSTM layer takes the output from the previous embedding layer (104, 64).This layer has two outputs, one is the forward output and another backward output. These outputs are integrated before passing it to the next layer by concatenation of the two outputs. Due to this integration, the number of outputs at this layer are doubled. In our model, it becomes 128.
- **LSTM Layer**: An LSTM network is a RNN that consists of LSTM cell blocks instead of the typical RNN layers. This layer has an output dimension of 256 (Twice the dimensions of the Bi-LSTM layer).
- **TimeDistributed** Layer: The TimeDistribute Dense layers provide completely-connected functioning across each output over each time-step. If this layer is not used, it would result in one-sole output. This layer takes the output dimension from the previous LSTM layer and outputs the maximum sentence length and maximum tags.

3.5 VALIDATING MODEL

The model was validated by analyzing accuracy obtained when testing data is provided to the ensemble model. 20% of the data provided for training the model was reserved for testing the model.

4. RESULTS AND ANALYSIS

The proposed model was tested by reserving a part of the training data for the purpose and the number of epochs used was 25 and the batch size being 1000. The graph in the Figure 5 shows the model accuracy and loss across the epochs:

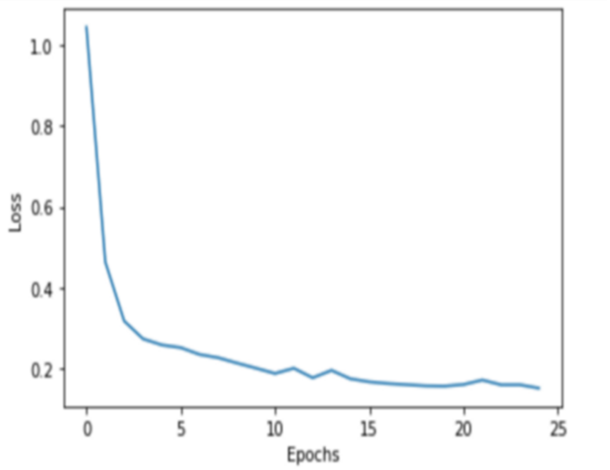


Fig-5: Results achieved with proposed model.

It was observed that the proposed neural network achieved an accuracy of 91% with the loss being at 1.2164 for the first epoch and by the last epoch, the network achieved an accuracy of 96.97% with the loss being reduced to 0.0916 which shows that the proposed model proved a high performance. The table 1 below presents the outputs achieved in the first, twelfth and the last epoch:

Table 1: Outputs for the proposed model w.r.t to intermediate epochs.

Epoch	Loss	Accuracy%
1	1.2164	91.00
12	0.1817	96.79
25	0.0916	96.97

This table shows that our model has gradually enhanced in recognizing the named entities over time.

5. CONCLUSION

As described in this paper, we have proposed a Bi-LSTM based Neural Network for the task of Named Entity Recognition. The results show that the model provides a high performance with an overall accuracy of 96.89% by the end of the implementation. The task of NER is difficult because, though the tokenization of a sequence of tokens will reveal its components, understanding the underlying context can be difficult. Hence, we have proposed a model based on LSTM, which can effectively understand the internal features and associations between the tokens and provide a leverage for recognizing named entities. To summarize, our results show that an Bi-Lstm based RNN enhances substantially upon current Named Entity Recognition approaches.

6. FUTURE WORK

The scope for enhancement over the proposed model could be to change the model hyperparameters like the number of epochs, embedding dimensions etc.

REFERENCES

[1] J. Li, A. Sun, J. Han and C. Li, "A Survey on Deep Learning for Named Entity Recognition," in IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 1, pp. 50-70, 1 Jan. 2022, doi: 10.1109/TKDE.2020.2981314.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[2] Shen, Yanyao & Yun, Hyokun & Lipton, Zachary & Kronrod, Yakov & Anandkumar, Animashree. (2018). Deep Active Learning for Named Entity Recognition. K. Elissa, "Title of paper if known," unpublished.

[3] Wu Y, Jiang M, Lei J, Xu H. Named Entity Recognition in Chinese Clinical Text Using Deep Neural Network. Stud Health Technol Inform. 2015;216:624-8. PMID: 26262126; PMCID: PMC4624324.

[4] A. Güneş and A. C. TantuĞ, "Turkish named entity recognition with deep learning," 2018 26th Signal Processing and Communications Applications Conference (SIU), 2018, pp. 1-4, doi: 10.1109/SIU.2018.8404500.

[5] Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, Ulf Leser, Deep learning with word embeddings improves biomedical named entity recognition, Bioinformatics, Volume 33, Issue 14, 15 July 2017, Pages i37–i48, https://doi.org/10.1093/bioinformatics/btx228.

[6] Hammerton, James. (2003). Named Entity Recognition with Long Short-Term Memory. Proceedings of CoNLL-2003. 4. 10.3115/1119176.1119202.

[7] Quinta de Castro, P.V., Félix Felipe da Silva, N., da Silva Soares, A. (2018). Portuguese Named Entity Recognition Using LSTM-CRF. In: , et al. Computational Processing of the Portuguese Language. PROPOR 2018. Lecture Notes in Computer Science(), vol 11122. Springer, Cham. https://doi.org/10.1007/978-3-319-99722-3_9.

[8] Dong, C., Zhang, J., Zong, C., Hattori, M., Di, H. (2016). Character-Based LSTM-CRF with Radical-Level Features for Chinese Named Entity Recognition. In: Lin, CY., Xue, N., Zhao, D., Huang, X., Feng, Y. (eds) Natural Language Understanding and Intelligent Applications. ICCPOL NLPCC 2016 2016. Lecture Notes in Computer Science(), vol 10102. Springer, Cham. https://doi.org/10.1007/978-3-319-50496-4_20.

[9] Roy, Aryan. "Recent Trends in Named Entity Recognition (NER)." ArXiv abs/2101.11420 (2021): n. pag.

BIOGRAPHIES

N V Sahana is currently working as an associate software engineer in Bosch Global Software Technologies, Electronic City, Bengaluru, Karnataka, India. She is a graduate from Bangalore Institute of Technology, Bengaluru, Karnataka, India.

Dr. Bhanushree K. J. Professor at the Department of Computer Science & Engineering, Bangalore Institute of Technology, Visveswaraya Technological University, Karnataka, India. Completed PhD in Computer Engineering from VTU University in 2022. Published several papers in international journals and conferences. Research areas include Image Processing, Face Recognition and Machine Learning.