*Hindi Named Entity Recognition*

Kunisetty Jaswanth  
*Dept. of Computer Science Engineering*  
*Amrita School of Computing*Bengaluru, India  
jaswanthkunisetty3301@gmail.com

S Sruthi   
*Dept. of Computer Science Engineering*  
*Amrita School of Computing*Bengaluru, India  
ssruthi1202@gmail.com

Pranav Ramachandrula   
*Dept. of Computer Science Engineering  
Amrita School of Computing*Bengaluru, India  
pranavramachandrula@gmail.com

Ms. Priyanka Vivek  
*Dept. of Computer Science Engineering*  
*Amrita School of Computing*

Bengaluru, India  
v\_priyanks@blr.amrita.edu

***Abstract : Named Entity Recognition is a crucial task in natural language processing that involves identifying and classifying named entities, such as person names, locations, organizations, and other significant entities, in text.The objective of this project is to develop an effective Named Entity Recognition (NER) system for the Hindi language.To achieve this goal, we explore and compare the performance of several state-of-the-art models on the Hindi NER task. Specifically, we employ BiLSTM (Bidirectional Long Short-Term Memory), BiLSTM-CRF (Conditional Random Fields), BiGRU (Bidirectional Gated Recurrent Unit), and IndicBERT models.***

***Keywords: Named Entity Recognition, Hindi NER, BiLSTM, BiLSTM-CRF, BiGRU, IndicBERT, Natural Language Processing***

1. INTRODUCTION

Named Entity Recognition (NER) is a fundamental task in natural language processing (NLP) that involves identifying and classifying named entities in text. Named entities can include person names, organization names, location names, date and time expressions, and more. NER plays a crucial role in various NLP applications, such as information retrieval, question answering, sentiment analysis, and knowledge extraction. While NER systems have achieved significant advancements for many languages, Hindi, one of the most widely spoken languages in the world, still poses unique challenges due to its complex linguistic characteristics.

In this project, we focus on developing an effective NER system specifically designed for the Hindi language. We employ four prominent models: BiLSTM, BiLSTM-CRF, BiGRU, and IndicBERT, each known for their effectiveness in capturing contextual information and sequence dependencies. The BiLSTM model leverages bidirectional recurrent neural networks to consider the context from both directions, enabling it to capture dependencies effectively. The BiLSTM-CRF model enhances the BiLSTM architecture by incorporating a conditional random field layer, which enforces global consistency in the predicted entity labels. The BiGRU model, similar to BiLSTM, utilizes bidirectional gated recurrent units to capture contextual information efficiently. Lastly, IndicBERT, a Hindi-specific variant of the BERT model, offers the ability to leverage pre-training on a large-scale Hindi corpus, providing a deep understanding of the language and its contextual intricacies.

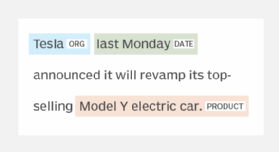
Through extensive experimentation and analysis, we aim to compare the performance of the models on the Hindi NER task. Metrics such as precision, recall, and F1 score will be employed to assess the models' ability to accurately identify and classify named entities in Hindi text. The results of this project will provide valuable insights into the effectiveness of different approaches for Hindi NER, contributing to the advancement of NLP research and facilitating practical applications in fields such as information extraction, document analysis, and information retrieval in Hindi language documents.

*a) Name Entity Recognition:*

Named Entity Recognition (NER) is a crucial task in natural language processing (NLP) that involves identifying and classifying named entities within a given text. Named entities refer to specific entities or objects that have names and play significant roles in the text. These entities can include person names, organization names, location names, date and time expressions, monetary values, and more.

The goal of NER is to accurately identify and categorize these named entities, extracting valuable information from unstructured text data. NER systems employ various techniques and models to analyze the linguistic patterns, context, and relationships within the text to identify and classify the named entities.

NER is essential for numerous NLP applications, including information retrieval, question answering systems, sentiment analysis, text summarization, and more. By accurately recognizing and categorizing named entities, NER systems enhance the understanding of text, enable efficient information extraction, and facilitate deeper analysis and processing of textual data.



1. RELATED WORK

Dadas and Protasiewicz [1] introduced a novel bidirectional iterative algorithm for the task of nested Named Entity Recognition (NER). This technique is focused on the identification and classification of named entities present within larger named entities, which is a less explored and challenging area of NER.

Yang and Xu [2] designed a Residual BiLSTM model for NER. This model uses bidirectional Long Short-Term Memory networks, along with residual connections to capture long-term dependencies in text. The approach improves the model's ability to learn from the semantic and syntactic relationships within the text.

Ngo, Kechadi, and Le-Khac [3] developed a semantic-based deep learning approach specifically for domain-specific NER. This approach leverages deep learning models to identify and classify named entities more accurately within specific fields or industries.

Zhang et al. [4] proposed a method for English drug name entity recognition that employs a BiLSTM-CRF model, enhanced with an attention mechanism. The attention mechanism helps the model focus on critical parts of the input sequence when predicting the output, which is particularly useful in the medical field for drug name extraction.

Chopra, Joshi, and Mathur [5] addressed NER in the Hindi language using Hidden Markov Models. This paper contributes to NER research for non-English languages, focusing on Hindi, where NER has received less attention compared to English.

Dalvi et al. [6] applied NER for a specialized task: drug-related page classification on the Dark Web. Their work showcases how NER can be used for targeted web content analysis and surveillance, demonstrating the versatility of NER applications.

Prasad and Fousiya [7] conducted a comparative study of NER approaches applied to the English and Hindi languages. Their work offers insights into the differences between NER tasks in these two languages and helps researchers understand the challenges associated with Hindi NER.

Jain, Yadav, and Tayal [8] implemented a system for Hindi NER using association rules. This approach demonstrates another method for NER in the Hindi language, enriching the set of techniques available for Hindi NER tasks.

Singh et al. [9] proposed a context-based deep learning approach for NER in Hindi. This technique enhances the ability of models to recognize named entities in Hindi by taking into account the wider context of the words. This is particularly crucial for languages like Hindi, where context can significantly change the meaning of words.

1. PREPROCESSING

*a) Using Direct Tokenization:*

*Loading the Data*: The script uses the load\_dataset function from the datasets package to load the 'HiNER-original' dataset. It then separates the data into training, validation, and test sets.

*Tokenization*: Each sentence in the dataset is already split into a list of tokens (words) and corresponding named entity recognition (NER) tags. This script consolidates all tokens and NER tags across the dataset into separate lists.

*Building Vocabulary*: The script then creates a vocabulary from the unique tokens and NER tags in the dataset, which is essentially a list of unique words and NER tags.

*Index Mapping*: An index mapping is created for both the tokens and the NER tags. This mapping is a dictionary where each unique token or tag is associated with a unique integer index. This is necessary for numerical operations during the training of the machine learning model.

*Padding*: Since sentences in the dataset can vary in length but machine learning models usually require fixed-size input, the script pads shorter sequences with a special padding value until they reach the length of the longest sequence in the dataset. This is done for both the tokenized sentences and the NER tag sequences.

*Converting to Categorical*: The last step in the preprocessing is to convert the NER tag sequences from arrays of integer values into binary class matrices, a process known as one-hot encoding. This is needed for multi-class classification where the model is expected to predict the probability distribution over the different NER classes for each token.

All these steps are applied consistently across the training, validation, and test data to ensure that they are in the same format. It's important to note that for validation and test sets, any token not found in the training set is considered as unknown ('<unk>') and mapped to the index associated with '<unk>' (22 in this case). For collapsed dataset the ‘<unk>’ is mapped to 6.

*b) Semi-Supervised Learning:*

The data preprocessing being done in this script is similar to the one in the previous script, but with an additional step of splitting the training data into labeled and unlabeled subsets. The purpose of this is typically for semi-supervised learning, where both labeled and unlabeled data are used during training. Here are the steps involved:

*Loading the Data*: The 'HiNER-original' dataset is loaded and divided into training, validation, and test sets.

*Tokenization*: Each sentence in the dataset is split into a list of tokens and corresponding NER tags, which are consolidated into separate lists.

*Building Vocabulary*: The script creates a vocabulary of unique tokens and NER tags in the dataset.

*Index Mapping*: A unique integer index is assigned to each token and tag in the vocabulary. This allows the model to work with numerical data.

*Splitting Labeled and Unlabeled Data*: The script sets aside a certain percentage of the training data (9% in this case) to be unlabeled. This is done by randomly selecting a subset of the training data. The rest of the training data remains labeled.

*Padding*: Sequences of tokens and tags are padded to match the length of the longest sequence in the dataset. This ensures that all inputs to the model have the same length.

*Converting to Categorical*: The NER tag sequences are converted to binary class matrices (a process known as one-hot encoding), which is required for the multi-class classification task the model is expected to perform.

*Preparing Unlabeled Data*: The script also preprocesses the unlabeled data by converting its tokens into integer indices and padding its sequences to the maximum length. However, since it's unlabeled data, there's no conversion to categorical as there are no associated tags to convert.

1. DATASETS DESCRIPTION

*1) HiNER-Original:*

This dataset was created for the task of Named Entity Recognition (NER) in the Hindi language by the CFILT Lab at IIT Bombay. The data was collected from various government information webpages and manually annotated. It contains sentences from ILCI and other sources. However, due to licensing requirements, the ILCI portion of the data is not distributed directly.

Languages: Hindi

Structure of Data:

id: The ID value of the data point.

tokens: Raw tokens in the dataset.

ner\_tags: the NER tags for this dataset.

Data Splits (original): Train - 76025, Valid - 10861, Test - 21722

*2) HiNER-Collapsed:*

The data structure and purpose are the same as the HiNER-Original dataset.

Languages: Hindi

Structure of Data:

id: The ID value of the data point.

tokens: Raw tokens in the dataset.

ner\_tags: the NER tags for this dataset.

Data Splits (collapsed): Train - 76025, Valid - 10861, Test - 21722

*3) Ai4bharat/naamapadam*:

Naamapadam is the largest publicly available Named Entity Annotated dataset for 11 Indic languages. This dataset was created by projecting named entities from English to the Indic language side of the English-Indic languages parallel corpus. It also includes a manually labelled test set for 8 Indic languages.

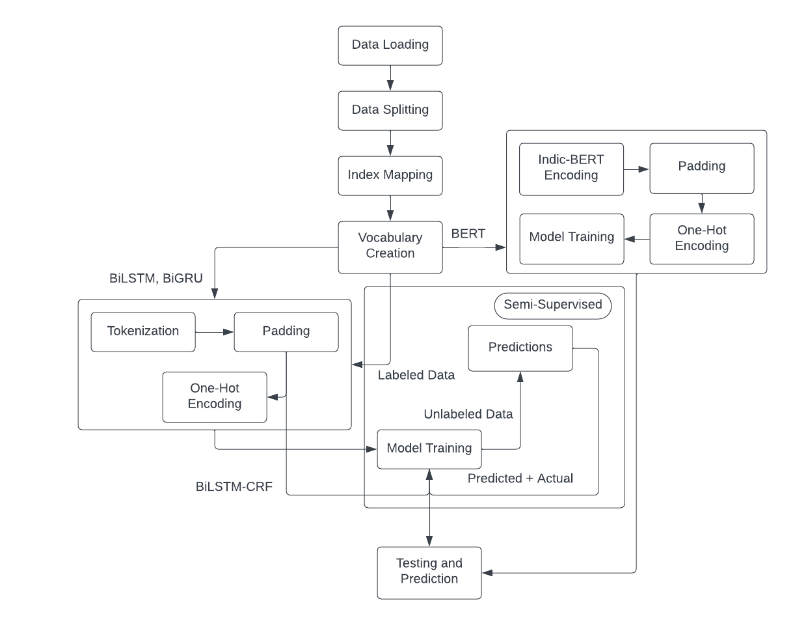
Languages: Assamese, Bengali, Gujarati, Kannada, Hindi, Malayalam, Marathi, Oriya, Punjabi, Tamil, Telugu.

Structure of Data:

words: Raw tokens in the dataset.

ner: the NER tags for this dataset.

1. METHODOLOGY



1. MODELS USED

*a) BiLSTM :*

BiLSTM (Bidirectional Long Short-Term Memory) is a variant of recurrent neural networks (RNNs) commonly used in natural language processing (NLP) tasks such as Named Entity Recognition (NER). Unlike traditional unidirectional RNNs, BiLSTM processes the input sequence in both forward and backward directions using two LSTM (Long Short-Term Memory) layers. This bidirectional approach allows the model to capture context and dependencies from both sides of each word, enhancing its ability to understand the sequence.

During the forward pass, the input sequence is processed sequentially, considering the previous context and the current input to update the hidden state of the forward LSTM layer. Simultaneously, the input sequence is processed in reverse during the backward pass, with the backward LSTM layer capturing dependencies and patterns in the opposite direction. The outputs of both LSTM layers are concatenated at each time step, resulting in a fused representation that combines the context from both directions. This concatenated representation encodes comprehensive information about the current word and its surrounding context, enabling the BiLSTM to capture long-range dependencies effectively.

*b) BiLSTM+ CRF*

The BiLSTM-CRF model is an extension of the BiLSTM (Bidirectional Long Short-Term Memory) architecture commonly used in tasks like Named Entity Recognition (NER). It enhances the BiLSTM model by incorporating a CRF (Conditional Random Field) layer on top of the BiLSTM layer. The BiLSTM captures contextual information and dependencies within the input sequence, while the CRF layer models the transitions between different labels in a global, sequence-level manner.

By incorporating the CRF layer, the BiLSTM-CRF model considers the dependencies between predicted labels and enforces global consistency in the predicted entity sequences. This enables the model to produce more coherent and accurate entity predictions by considering the overall label sequence rather than just individual word predictions. During training, the model is optimized using annotated data, adjusting the parameters of both the BiLSTM and CRF layers simultaneously to maximize the likelihood of the correct entity label sequence given the input sequence.

*c) BiGRU:*

BiGRU (Bidirectional Gated Recurrent Unit) is a variant of recurrent neural networks (RNNs) that is particularly relevant to your Hindi NER project. While similar to BiLSTM, BiGRU utilizes gated recurrent units (GRUs) instead of long short-term memory (LSTM) cells as its basic building blocks.

In the context of your project, BiGRU can effectively capture the sequential dependencies and linguistic patterns in Hindi text. By processing the input sequence bidirectionally, BiGRU models can extract valuable context from both ends of each word, enhancing their understanding of the Hindi language.

The key advantage of BiGRU lies in its ability to capture long-range dependencies efficiently and mitigate the vanishing gradient problem often encountered in traditional RNNs. The GRU units within BiGRU incorporate gating mechanisms that selectively retain and update information in the hidden states, allowing for the propagation of relevant context throughout the sequence.

*d) IndicBERT:*

IndicBERT, a specialized language model for Indian languages, offers significant advantages for your Hindi NER project. It goes beyond generic language models by capturing the intricacies and linguistic nuances specific to Indian languages. With its pretrained nature, IndicBERT has learned from a large amount of Indian language text, enabling it to understand the complexities of Hindi and other Indian languages effectively. By incorporating contextualized word embeddings and a self-attention mechanism, IndicBERT can grasp the contextual dependencies and hierarchical structures within Hindi text, facilitating accurate and robust named entity recognition. Leveraging IndicBERT in your project will empower your system to better handle the challenges of Hindi NER, resulting in improved performance and more precise identification of named entities in Hindi language text.

Tag representations for all 23 original tags are as follows:

B-FESTIVAL, B-GAME, B-LANGUAGE, B-LITERATURE, B-LOCATION, B-MISC, B-NUMEX, B-ORGANIZATION, B-PERSON, B-RELIGION, B-TIMEX, I-FESTIVAL, I-GAME, I-LANGUAGE, I-LITERATURE, I-LOCATION, I-MISC, I-NUMEX, I-ORGANIZATION, I-PERSON, I-RELIGION, I-TIMEX, O

Tag representations of all 7 collapsed tags are as follows:

B-LOCATION, B-ORGANIZATION, B-PERSON, I-LOCATION, I-ORGANIZATION, I-PERSON, O

1. RESULTS



1. CONCLUSION

Overall, in all the custom models built for Hindi-NER, the Bidirectional GRU showed better performance of all. Not only accuracy but all the metrics are comparatively higher than the other models and techniques. Here, mathematically the difference between all the models metrics are in decimals. But according to the pre-processing techniques, padding is involved in all the training methods and as while training, validating and testing it also checks for matching padding values and thus they cover most of the evaluation metrics. So, slightest difference in any metric in this scenario matters a lot with respect to models performance.

1. REFERENCES

[1] S. Dadas and J. Protasiewicz, "A Bidirectional Iterative Algorithm for Nested Named Entity Recognition," in IEEE Access, vol. 8, pp. 135091-135102, 2020, doi: 10.1109/ACCESS.2020.3011598.

[2] G. Yang and H. Xu, "A Residual BiLSTM Model for Named Entity Recognition," in IEEE Access, vol. 8, pp. 227710-227718, 2020, doi: 10.1109/ACCESS.2020.3046253.

[3] Q. H. Ngo, T. Kechadi and N. -A. Le-Khac, "Domain Specific Entity Recognition With Semantic-Based Deep Learning Approach," in IEEE Access, vol. 9, pp. 152892-152902, 2021, doi: 10.1109/ACCESS.2021.3128178.

[4] Y. Zhang, L. Guo, D. Huang, K. Huang, J. Li and Z. Pan, "English Drug Name Entity Recognition Method Based on Attention Mechanism BiLSTM-CRF," 2019 IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Dalian, China, 2019, pp. 831-836, doi: 10.1109/ISKE47853.2019.9170393.

[5] D. Chopra, N. Joshi and I. Mathur, "Named Entity Recognition in Hindi Using Hidden Markov Model," 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India, 2016, pp. 581-586, doi: 10.1109/CICT.2016.121.

[6] A. Dalvi, V. Shah, D. Gandhi, S. Shah and S. G. Bhirud, "Name Entity Recognition (NER) Based Drug Related Page Classification on Dark Web," 2022 International Conference on Trends in Quantum Computing and Emerging Business Technologies (TQCEBT), Pune, India, 2022, pp. 1-5, doi: 10.1109/TQCEBT54229.2022.10041261.

[7] G. Prasad and K. K. Fousiya, "Named entity recognition approaches: A study applied to English and Hindi language," 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015], Nagercoil, India, 2015, pp. 1-4, doi: 10.1109/ICCPCT.2015.7159443.

[8] A. Jain, D. Yadav and D. K. Tayal, "NER for Hindi language using association rules," 2014 International Conference on Data Mining and Intelligent Computing (ICDMIC), Delhi, India, 2014, pp. 1-5, doi: 10.1109/ICDMIC.2014.6954253.

[9] S. Singh, S. Patel, Y. Shah, R. Nargunde and J. Ramteke, "Context-Based Deep Learning Approach for Named Entity Recognition in Hindi," 2021 International Conference on Communication information and Computing Technology (ICCICT), Mumbai, India, 2021, pp. 1-5, doi: 10.1109/ICCICT50803.2021.9510090.