Computer Speech and Language 74 (2022) 101356

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl

Named entity recognition using neural language model and CRF for Hindi language

Richa Sharma [a,*], Sudha Morwal [a], Basant Agarwal [b]

[a] Department of Computer Science and Engineering, Banasthali Vidyapith, India
[b] Department of Computer Science and Engineering, Indian Institute of Information Technology Kota, India

ARTICLE INFO

Keywords:
Neural network
Sequence labeling
MuRIL
Multilingual BERT
Transfer-learning
Language models

ABSTRACT

Named Entity Recognition (NER) plays an important role in various Natural Language Processing (NLP) applications to extract the key information from a huge amount of unstructured text data. NER is a task of identifying and classifying the named entities into predefined categories for a given text. Recently, language models are highly appreciable in several NLP tasks as these state-of-the-art models result better even in resource scarcity. In this paper, we perform NER task on the Hindi language by incorporating the recently released multilingual language model MuRIL which stands for Multilingual Representation for Indian Languages. MuRIL is specially trained for 16 Indian languages. We develop a Hindi NER system using MuRIL with a conditional random field (CRF) layer and fine-tune the model on the ICON 2013 Hindi NER dataset. Further, in the proposed approach, we compute the addition of the last 4 layers representations of the MuRIL model instead of just using the last layer's representation and fine-tune the whole model. Several variants of this model are presented by applying different computations on token representations provided by different layers of 12-layered MuRIL architecture. The proposed model achieves state-of-the-art results as 87.89% precision, 83.74% recall and 85.77% F1-score and outperforms all other existing Hindi NER systems developed on the ICON 2013 dataset. Additionally, we develop a similar Hindi NER system by replacing the MuRIL language model with another state-of-the-art language model, called multilingual Bidirectional Encoder Representations from Transformers (mBERT) to analyze the efficiency of both language models over the Hindi NER task.

1. Introduction

Named Entity Recognition (NER) is an imperative task in the field of Information Extraction. NER is a two-step process that comprises the identification and classification of named entities into predefined categories such as the name of person, place, organization, a numerical expression such as the amount of money, time expression and so on. A named entity (NE) can be a word/term or series of words, present in the piece of text as key information (Nadeau and Sekine, 2007). The below example shows the processing and outcome of the NER task.

Consider a couple of sentences: भारत के 11वें राष्ट्रपति डॉ. ए.पी.जे अब्दुल कलाम का जन्म 15 अक्टूबर 1931 को तमलिनाडु के रामेश्वरम में हुआ था । उन्होंने मुख्य रूप से रक्षा अनुसंधान और वकिास संगठन और भारतीय अंतरक्षि अनुसंधान संगठन में एक
वैज्ञानकि के रूप में चार दशक बतिाए।

* Corresponding author.
E-mail address: sharma.ric1@gmail.com (R. Sharma).

https://doi.org/10.1016/j.csl.2022.101356
Received 10 August 2021; Received in revised form 11 November 2021; Accepted 16 January 2022
Available online 18 January 2022

NER system first identifies the named entities such as

भारत, 11वें, राष्ट्रपति, डॉ. ए.पी.जे अब्दुल कलाम, 15 अक्टूबर 1931, तमलिनाडु, रामेश्वरम, रक्षा अनुसंधान और वकिास संगठन, भारतीय अंतरक्षि अनुसंधान संगठन, वैज्ञानकि, चार दशक.

In the next step, the NER system classifies these named entities in particular categories such as:

Location: भारत, तमलिनाडु, रामेश्वरम

Count: 11वें

Designation: राष्ट्रपति, वैज्ञानकि

Person: डॉ. ए.पी.जे अब्दुल कलाम

Date: 15 अक्टूबर 1931

Organization: रक्षा अनुसंधान और वकिास संगठन, भारतीय अंतरक्षि अनुसंधान संगठन

Period: चार दशक

The correct identification of such entities is important in information extraction, as it helps to generate structured information from unstructured data. These named entities are also valuable in search engines for indexing, organizing, and linking the documents efficiently. This further improves the accessibility of the documents to users based on probed named entities. For example, NER can be used to correctly identify the people characterize in the news article published over a period of time. Not only this, NER can be implemented as a sub-task in several Natural Language Processing (NLP) applications such as question-answering (Greenwood and Gaizauskas, 2003), text summarization (Toda and Kataoka, 2005), machine translation (Babych and Hartley, 2003), word-sense disambiguation (Moro et al., 2014), conference resolution (Dimitrov et al., 2005) and semantic search (Han and Zhao, 2010), etc., to escalate the performance of such applications.

In recent years, language models have gained significant attention in several NLP processes. Language modeling generates better representation and complete understanding of a language, so when we transfer this learning on a downstream task such as NER (Souza et al., 2019), sentiment analysis (Hoang et al., 2019), paraphrase detection (Arase and Tsujii, 2021), news category classification (Liu et al., 2020), headline prediction (Ma et al., 2020), speech recognition (Huang et al., 2021), machine translation (Zhu et al., 2020) and so on, it performs better even if, the labeled data is scarce.

Several language models are being developed for monolingual as well as multilingual languages in recent years. Language models such as BERT (Devlin et al., 2019), ALBERT (Soricut, 2020), T5 (Raffel et al., 2020), ELECTRA (Clark et al., 2020) and RoBERTa (Liu et al., 2019), etc. are performing well for the English language only while multilingual BERT (mBERT), XLM (Lample and Conneau, 2019), XLM-R (Conneau et al., 2019), MT5 (Xue et al., 2020) have been developed for multilingual languages.

Although Multilingual Language Models (MLLMs) are often trained on more than 100 languages concurrently, still the performance of these models is not such impressive for Indian languages, this may be due to the small representation of Indian languages in their vocabulary and training data. Besides that, the performance of these MLLMs is not satisfactory even for resource-scarce languages as limited resources are not sufficient to elaborate the various linguistic rules of the language. This gap can be fulfilled through the recently released Multilingual Representations for Indian Languages (MuRIL) (Khanuja et al., 2021), a language model which is trained especially on Indian languages.

MuRIL (Khanuja et al., 2021) has 12-layered encoder architecture similar to BERT-base language model. It can also handle Indian language text transliterated to Latin or code-mixed with English such as on social media platforms. MuRIL is trained with two learning objectives: a) Masked Language Modeling (MLM) (Taylor, 1953); and b) Translation Language Modeling (TLM) (Lample and Conneau, 2019);

In MLM, MuRIL is trained in an unsupervised way on monolingual text documents while for TLM it is trained on both translated and transliterated document pairs in a supervised manner. MuRIL significantly outperforms mBERT on all tasks in the challenging cross-lingual XTREME (Hu et al., 2020) benchmark. MuRIL is trained on 17 languages which include English language and 16 Indian languages. To the best of our knowledge, this is the first work that leverages the power of the MuRIL language model with conditional random field (CRF) (Lafferty et al., 2001) layer for the Hindi NER task.

In this work, we are aimed to develop a robust Hindi NER model that would be able to categorize a diverse set of named entities on raw text without any feature engineering and linguistic rules. More specifically, here, we design a hybrid deep neural network model which is benefited from the MuRIL neural language model and the CRF layer. The linear chain CRF layer is integrated as an output layer that implements sequential dependencies in the predictions.

In the next step, we manipulate the token representations produced by the MuRIL model on each layer. Finally, we calculate and apply the summation of token representations of the last four hidden layers of MuRIL architecture rather than applying only the last layer's representations. We named this proposed model as MuRIL-$CRF_{sum\_4\_layer}$ model and fine-tuned the model for the Hindi NER task. The proposed MuRIL-$CRF_{sum\_4\_layer}$ model achieves a state-of-the-art F1-score of 85.77% and outperforms other existing Hindi NER models evaluated on the ICON 2013 dataset. In addition, we develop another NER system using mBERT language model with CRF approach to analyze the efficiency of both language models over the Hindi NER task.

Major contributions of this research work are (i) developing MuRIL-based NER model as a new state-of-the-art for Hindi language (ii) developing another Hindi NER model based on mBERT language model (iii) extending these models by incorporating CRF as an output layer (iv) extensive experiments are carried out on MuRIL and mBERT architectures to capture the best token representations emerged by both models. In these experiments, different computations are performed on their pre-trained activations.

This paper is outlined as follows: Section 2 defines the research problem more specifically. Section 3 discusses the previous work accomplished in this field. Section 4 describes the proposed model architecture design. Section 5 elaborates the variants of the proposed model. Additionally, this section also describes the mBERT-based Hindi NER model and its variants. Section 6 presents experimental settings that include detailed information on the ICON 2013 dataset, hyperparameters tuning and evaluation metrics.

Section 7 discusses the results achieved by all experiments in this work as well as previous results achieved on the ICON 2013 dataset. This section also discusses the entity-wise classification results achieved through experimental models and analyzes the classifications produced by the proposed model on some random test samples. Finally, Section 8 concludes this work and discusses the future work.

2. Problem definition

Let S_1 is a sentence defined as S_1: $\{X_1,X_2,X_3,\ldots,X_t,\ldots X_N\}$ where N is number of words or N sets of features to represent the sentence S_1 and X_t represents a word at position t in the sequence S_1.

Considering a labeled sequence L_1 associated with the sentence S_1, represented as L_1: $\{Y_1,Y_2,Y_3,\ldots,Y_t,\ldots,Y_N\}$ where Y_t is label information associated with the word X_t in the sentence S_1. Thus, L_1 would also be a sequence of N labels mapped to each word of the sentence S_1.

Now assume there is a lean collection of such labeled training samples of Hindi sentences with their corresponding tag information.

So, the problem addressed in this paper is how to annotate or label each word of unseen, untagged Hindi sentence with their correct label. To solve the problem, we are aimed to develop a robust and efficient NER model that would be able to identify and annotate the named entities into predefined categories, for a given sentence, even if the set of predefined categories is highly diverse.

3. Related work

NER systems for the Hindi language can be studied in two phases as conventional NER systems and enhanced NER systems. Conventional NER systems include rule-based approaches and machine learning-based Hindi NER systems while enhanced NER

Table 1
Summarized report of applied approaches on Hindi NER task.

Model	Classifier	Attributes	Data Set
(Saha et al., 2008)	Maximum Entropy	orthographic features (decimal, digits), word affixes, context words, POS, gazetteer lists	243 K words collected from Hindi Newspaper "Dainik Jagaran"
(Ekbal and Bandyopadhyay, 2009)	CRF	Context words, orthographic word-level features, POS information, gazetteer lists such as first, middle, last names, weekdays, month names etc.	IJCNLP-2008 dataset
(Saha et al., 2010)	SVM	Current word, previous word information, prefix and suffix feature, NE tags of the previous words	243 K words collected from Hindi Newspaper "Dainik Jagaran"
(Ekbal and Bandyopadhyay, 2010)	SVM	contextual words, affixes of word, digit features, length of word feature, frequency of word feature and POS information etc.	IJCNLP-2008 dataset
(Devi et al., 2016)	SVM	stylometric features such as hyperlink, hash (#), apostrophe ('), numbers, punctuation, affixes	Hindi-English and Tamil-English code-mixed social media text
(Morwal et al., 2012)	HMM	Frequency of a tag, frequency of a tag sequence, frequency of a word	NLTK Indian Corpus
(Gayen and Sarkar, 2014)	HMM	POS information, chunk information, suffix of word feature	ICON-2013 dataset
(Chopra et al., 2016)	HMM	Frequency of a tag, frequency of a tag sequence, frequency of a word	IJCNLP-2008 dataset
(Gali et al., 2008)	CRF+Rule-based approach	Linguistic rules, orthographic features, affixes, digit features, context words, gazetteer lists, POS features	IJCNLP-2008 dataset
(Biswas et al., 2010)	Maximum entropy+ HMM	prefix and suffix feature, context word feature, digit information	List of gazetteers annotated in shakti standard format
(Devi et al., 2013)	Rule-based+CRF	Linguistic rules, POS information, chunk information, orthographic features	ICON-2013 dataset
(Srivastava et al., 2011)	CRF +maximum entropy +Rule-based	Linguistic rules, POS information, orthographic word-level features	IJCNLP-2008 dataset
(Chopra et al., 2012)	HMM + Rule-based approach	Several linguistic rules	General corpus created from Hindi newspaper and annotated manually.
(Kaur and Kaur, 2015)	Rule-based+ List lookup	Several rules with the new "no name entity rule"	Data collected from e-copies of Hindi newspapers such as "Danik Jagran", "Punjab Kesari" etc.
(Athavale et al., 2016)	Bidirectional LSTM	POS tag information of word	ICON-2013 dataset
(Murthy, 2017)	Bidirectional LSTM-CNN	–	IJCNLP-2008 dataset
(Singh et al., 2018)	Decision tree, CRF, LSTM	Char N-gram for suffix, patterns for punctuation, emoticons, numbers, numbers in strings, previous tag information	Hindi-English code-mixed 3638 tweets from last 8 years scrapped from Twitter
(Gupta et al., 2018)	GRU	–	Code-mixed Indian Social media text
(A P et al., 2019)	Bidirectional LSTM-CNN-CRF	Prefix and suffix of word feature	ARNEKT-IECSIL 2018 data
(Shah and Kopparapu, 2019)	De-noising auto-encoder LSTM	–	IJCNLP-2008 dataset
(Sharma et al., 2020)	Bidirectional LSTM-CNN-CRF	–	ICON-2013 dataset

systems describe Hindi NER on deep neural networks and transformer-based models. Table 1 shows the summarized history of applied approaches used to develop the Hindi NER system.

3.1. Conventional Hindi NER systems

A rule-based system for Hindi NER was designed with the combination of the list look-up approach (Kaur and Kaur, 2015). This system worked for three new named entities as money, direction and animal/bird. Since rule-based systems require a large set of linguistic rules to work and creating efficient rules is a very time-consuming process as well as it requires a language expert that's why machine learning approaches were explored by researchers for the NER task.

Several machine learning-based NER models have been developed for Hindi. In this series, some Hindi NER systems (Chopra et al., 2016; Gayen and Sarkar, 2014; Morwal et al., 2012) were developed by applying the hidden markov model (HMM) approach of machine learning. The authors (Morwal et al., 2012) used tourism corpus and NLTK Indian corpus to develop the NER model. Gayen and Sarkar (2014), applied the statistical HMM model on ICON 2013 NER dataset and evaluate the model on 7 Indian languages including Hindi. Another NER system for the Hindi language was developed by applying the maximum entropy approach (MaxEnt) of machine learning (Saha et al., 2008). Their system was trained to classify name entities belonging to the location, person, date and organization. Another approach of machine learning namely support vector machine (SVM) was applied for Bengali and Hindi (Ekbal and Bandyopadhyay, 2010), where several language-independent features were used such as contextual words feature, affixes of a word, several digit features, length and frequency of the word, part-of-speech and lexical context patterns. Besides that, Saha et al. (2010) used a novel kernel function for SVM and applied it on the biomedical domain. Another NER system was developed using SVM for Hindi-English and Tamil-English code-mixed social media text (Devi et al., 2016). A memory-based learning method using K-nearest neighbor (KNN) was applied with gazetteer lists, linguistic rules and features (Sarkar and Shaw, 2016). Moreover, the CRF algorithm was also implemented for the Hindi NER task (Das and Garain, 2014; Ekbal and Bandyopadhyay, 2009; Singh et al., 2018).

To increase the performance of the NER system several hybrid systems were also developed by combining rule-based and machine learning-based approaches. A hybrid system was developed by Gali et al. (2008) for Hindi NER which integrated CRF approach and rule-based heuristics. Similarly, the MaxEnt and HMM approaches were integrated to develop the NER model (Biswas et al., 2010). Their system also benefitted from a variety of features and contextual information. Moreover, rule-based heuristics and HMM were integrated to accomplish the Hindi NER task (Chopra et al., 2012). Another hybrid approach was implemented by integrating CRF, MaxEnt and rule-based approaches (Srivastava et al., 2011). In their work, to enhance the performance of the system, the authors applied the voting method. Similarly, Devi et al. (2013) applied rule-based heuristics with the CRF approach on the Hindi NER dataset. This extensive survey of past approaches shows that the machine learning-based models require high quality of handcrafted features and language-specific resources such as gazetteers and POS taggers to perform well. However, it is difficult to build such a model for resource-scarce language such as Hindi. Thus, to eliminate the need for linguistic resources and manually designed features, recently deep neural network-based models have shown successful results.

3.2. Enhanced Hindi NER systems

In this series, a Hindi NER system based on a deep learning approach developed by Athavale et al. (2016) in which GloVe (Pennington et al., 2014) and Skip-gram (Mikolov et al., 2013) embedding approach were used for creating word-vectors and designed the model with bidirectional long short term memory (Bi-LSTM) (Hochreiter and Schmidhuber, 1997) layer. However, they also used POS tag information for all sequences. Similarly, Gupta et al. (2018) developed a deep learning-based NER model, designed with character and word embedding layer with Gated Recurrent Unit (GRU) layer. The system was trained and evaluated on code-mixed Indian social media text corpus. A neural network-based model comprised of Bi-LSTM and convolutional neural network (CNN) layer was developed by Murthy (2017). In this model, words were represented via word embedding and character sequence was used to extract sub-word features using CNN. Both word embedding and sub-word features were concatenated and fed through Bi-LSTM to softmax layer to classify named entities. A neural model designed by A P et al. (2019), wherein three levels of representation of an input word named word-level, character-level and affix level were concatenated and fed as input into the Bi-LSTM-CRF layer. Shah and Kopparapu (2019) applied BiLSTM based techniques for the Hindi NER task with two enhancements namely de-noising auto-encoder (DAE) LSTM and conditioning LSTM which shows improvement on NER task compared to the BiLSTM approach. Recently, Sharma et al. (2020) used CNN to generate character embedding and integrated it with the corresponding word vectors of each token. In their work, the authors applied Bi-LSTM-CNN—CRF approach for NER classification.

Recently, language models are highly appreciated in the field of NLP. Language models can emerge the semantic and syntactic relationship between words and reduce feature engineering. However, only a few language models such as mBERT, XLM-R (Conneau et al., 2019), MT5 (Xue et al., 2020), MuRIL (Khanuja et al., 2021) and IndicBERT (Kakwani et al., 2020) are available for multilingualism. In this series, mBERT, XLM-R and MT5 are trained on 100+ languages including some Indian languages whereas IndicBERT and MuRIL are fully focused on Indian languages. Though mBERT has been applied for the NER task on several different languages such as Portuguese (Souza et al., 2019), Spanish (Hakala and Pyysalo, 2019), Chinese (Cui et al., 2019), Russian (Mukhin, 2020), German (Labusch et al., 2020) and Slavic (Arkhipov et al., 2019) languages, no above-mentioned language model has yet been studied for Hindi NER task. Thus, we are aimed to fine-tune MuRIL and mBERT language model for the Hindi NER task on the ICON 2013 dataset.

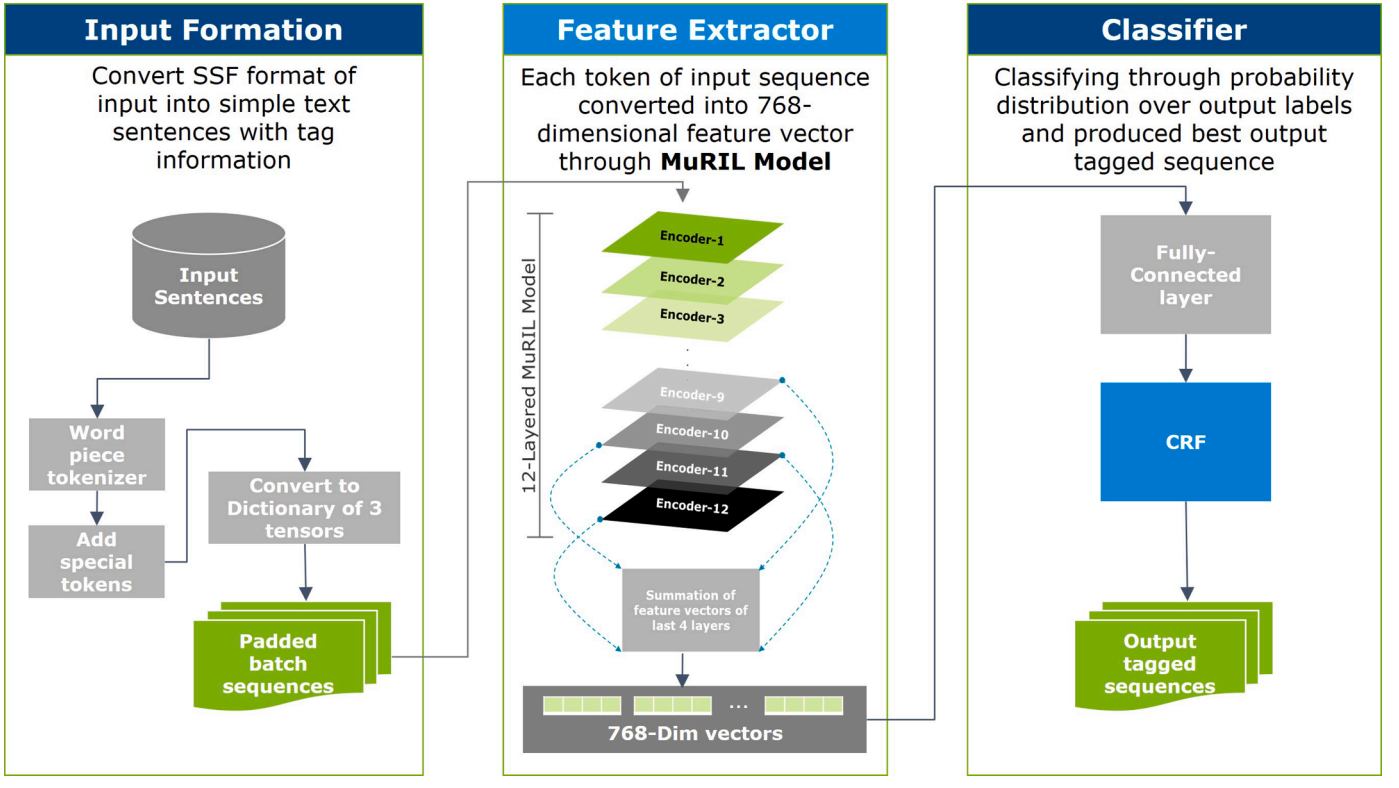

Fig. 1. Architecture of proposed model.

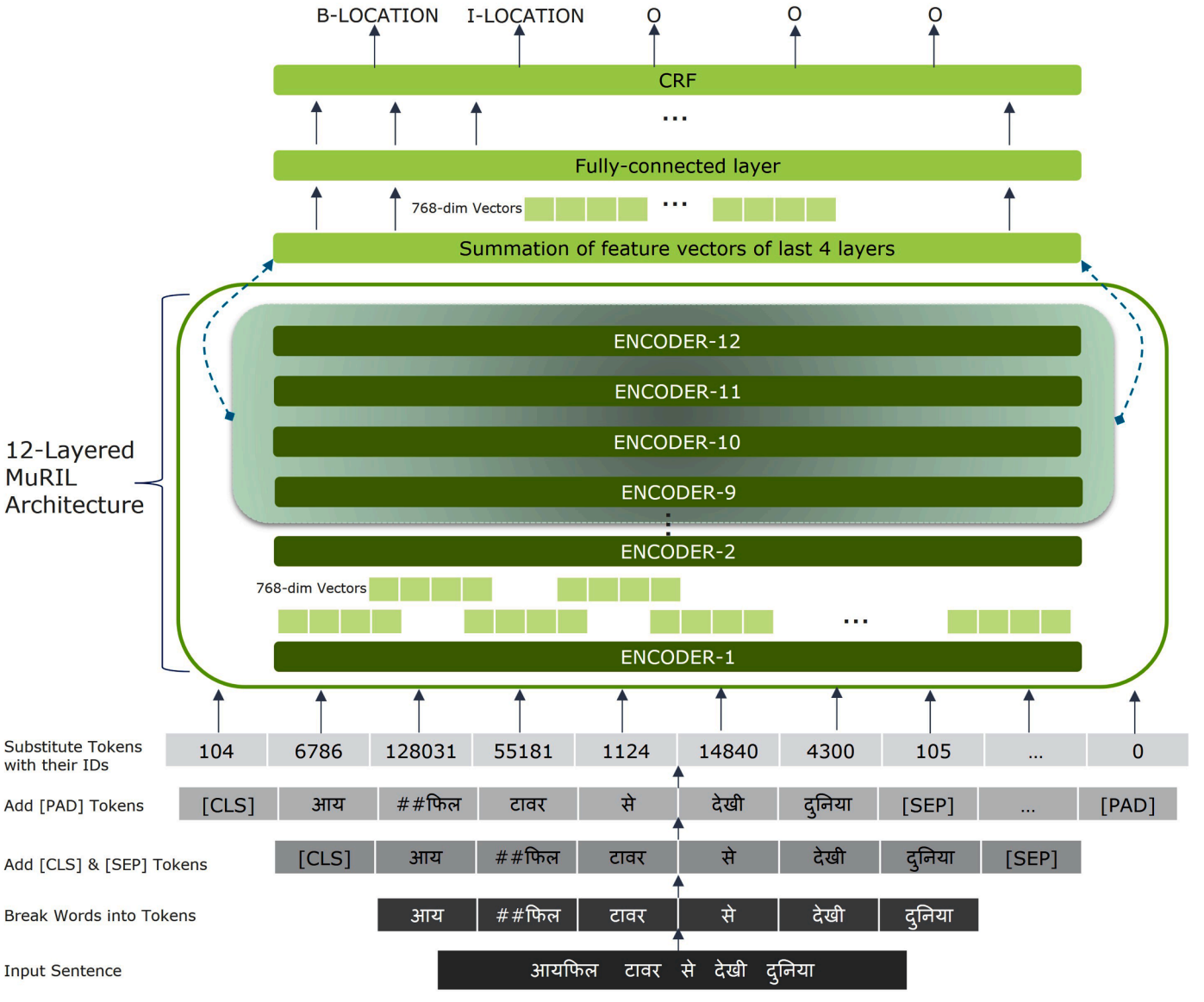

Fig. 2. Processing of a sentence through proposed model.

4. Model architecture

The proposed model is based on a hybrid deep neural network approach that incorporates the recently developed MuRIL (Khanuja et al., 2021) pre-trained model especially trained for Indian languages. The architecture of the MuRIL language model is comprised of 12 encoder layers with 12 self-attention heads and 768 hidden layer dimensions. Thus, every input token to the MuRIL layer is eventually represented as a 768-dimensional dense vector.

For the proposed NER model, first, we develop a simple $\text{MuRIL}_{\text{base\_classifier}}$ model wherein the MuRIL model is integrated as an input layer and tokens representations from the last layer of the MuRIL model are passed through an output layer. In this model, a fully connected layer with softmax activation is used as an output layer which produces the probability distribution over the output labels.

Further, we extend the $\text{MuRIL}_{\text{base\_classifier}}$ model by integrating CRF as an output layer at the top of the model instead of the softmax probability distribution, naming this baseline model as MuRIL-CRF model. Here, the aim of using CRF is to assign the best tag sequence for a given input sentence after considering all possible tag sequences and correlation between adjacent tags. CRF uses a single exponential model to determine the joint probability of the entire sequence of labels, given the observation sequence.

The MuRIL language model processes every input token through its 12-layered encoder architecture where each encoder block or layer is comprised of a self-attention layer and a feed-forward neural network. The self-attention layer helps the encoder to encode the specific word by considering all other words in the input sequence. Thus, every encoder layer generates its output feature vector corresponding to each input token as a 768-dimensional contextual feature vector. Since the input vector augments itself with some additional contextual feature values at every encoder layer, so the final hidden state of the last encoder i.e. last layer's representation enriches with a highly deep contextual feature vector.

This phenomenon is considered in the proposed strategy, and we create a model where the output emerged by some specified layers of the MuRIL language model are manipulated, rest architecture is the same as the MuRIL-CRF model. In the proposed model, we add the token representations produced by the last four encoder layers of 12-layered MuRIL architecture as later layers represent a rich set of features in terms of weights compared to initial layers. These added representations are passed into the subsequent fully connected layer and then the output of the fully connected layer is processed through the CRF layer for the classification task.

Eventually, all parameters of the proposed model are jointly fine-tuned on the downstream Hindi NER task. We named this proposed model as $\text{MuRIL-CRF}_{\text{sum\_4\_layer}}$ model. The architecture of the proposed $\text{MuRIL-CRF}_{\text{sum\_4\_layer}}$ is depicted in Fig. 1. As shown in Fig. 1, the proposed model is a three-step process as follows:

i In the first step, we perform the data pre-processing and form the input compatible with the MuRIL model.
ii The second step discusses the processing of the input through 12 layers of the MuRIL and
iii The last step describes the classifier used for the sequence labeling task.

The step-by-step processing of a Hindi sentence through the proposed model is illustrated in Fig. 2. First, we convert the dataset into simple text sentences where every sentence has its words with corresponding tag information. Since we are intended to evaluate the performance of the model on raw text only, thus for data pre-processing, we only apply three steps as data cleaning, sentence segmentation and tokenization on the ICON 2013 Hindi NER dataset which is available in shakti standard format (SSF) (Bharati et al., 2007).

In the data cleaning step, we just removed some 'NaN' values present in the dataset. Subsequently, sentences are segmented and tokenized on word level by word-piece tokenizer where split word pieces are denoted by '##' as shown in Fig. 2. In the next step, this tokenized sequence is appended with two special tokens as [CLS] and [SEP]. The output corresponding to [CLS] token represents the entire input sequence and is used for the sequence classification task. Since NER is the token-level classification task, the output corresponding to each input token is required rather than the output of [CLS] token only as shown in Fig. 2. The [SEP] token separates two input sentences in several sentence pair tasks. Though NER requires a single sentence as an input, this token is added to make the input format compatible with the MuRIL model requirement. Subsequently, the tokenized sequence is transformed into a dictionary of three int32 tensors to feed as an input to the MuRIL model.

The specifications of the three tensors are:

i input_word_ids: It lists the token_ids of the input sequence including [CLS], [SEP] and padding tokens.
ii input_mask: It lists with value 1 for all the input tokens and 0 for the padding tokens.
iii input_type_ids: Since the NER task requires only one sentence as an input, input_type_ids list with value 0 for all the input tokens including [CLS], [SEP] and padding tokens.

In this work, we also aimed to speed up the training of the model. For this purpose, instead of padding all input sequences up to a single fixed length, the length of padding tokens is varied for each batch. More specifically, all input sequences are sorted in ascending order of their length and put into batches. The input sequences within a batch are padded up to the max sequence length of that batch. Further, this dictionary of three input tensors, having the shape of (batch_size, 3, max_seq_length_batch) is passed to MuRIL pre-trained model.

At the first encoder layer of the MuRIL model, every input token is transformed into a 768-dimensional embedding vector by using the embedding algorithm. The input embeddings are the sum of token embeddings, segment embeddings and position embeddings. The position embeddings are the learnable embedding vectors that capture the dependency between tokens at different positions in the input sequence. The position embedding encodes the absolute position from 1 to maximum sequence length.

These 768-dimensional feature vectors corresponding to each token get enriched layer by layer in 12-layered MuRIL architecture as shown in Fig. 2. Eventually, for an input token, feature vectors produced from the last four encoder layers of the MuRIL model are summed up and marked as the final feature vector for that input token. This addition is performed for all input sequences and thus MuRIL generates output having the shape of (batch_size, max_seq_length_batch, 768). This output is passed to the classifier.

The subsequent fully connected layer in the MuRIL-CRF$_{sum_4_layer}$ model generates the output of the shape of (batch_size, max_seq_length_batch, num_classes) where num_classes denotes the number of entity classes for the classification task. Finally, the output of the fully connected layer is passed through the CRF layer which generates the best output tag sequence for a given input sentence as shown in Fig. 2.

The proposed MuRIL-CRF$_{sum_4_layer}$ model is trained for 75 epochs and a dropout layer is applied for regularization purposes with a 40% of dropout rate i.e. 40% connections to the subsequent layer are randomly dropped while training to prevent the model from overfitting. The proposed MuRIL-CRF$_{sum_4_layer}$ model achieves an F1-score of 85.77% and outperforms other existing Hindi NER models trained on the ICON 2013 dataset.

5. Variants of proposed model

We develop several variants of the proposed MuRIL-CRF$_{sum_4_layer}$ NER model to show the efficacy of the proposed model. These variants are based on the different ways of computing the activations, produced from some specified layers of 12-layered MuRIL architecture. Rather than passing only the last layer's activations, these computed activations are passed into the subsequent dense layer in MuRIL-CRF architecture. All experimental models are fine-tuned on the ICON 2013 dataset for the Hindi NER task. Thus, different experimental models based on different computing techniques applied on activations of specified layers of MuRIL are described below. The results achieved by these variations are reported in Table 4.

- Applying activations of second-to-last hidden layer: In this experiment, we apply token representations from the eleventh layer of the MuRIL pre-trained model into the fully connected layer and subsequent output is processed through the CRF layer. All the parameters of the model are then jointly fine-tuned for the Hindi NER task. We called this new model as MuRIL-CRF$_{eleventh_layer}$ model.
- Adding activations of all 12 layers: MuRIL is trained layer by layer, thus each layer has more information than the previous layer in terms of features. It could be a good idea to add all 12 layers' weights and get such weights that are leveraged by a rich set of features. We implement this idea as a MuRIL-CRF$_{sum_12_layer}$ model and fine-tuned the model on the Hindi NER task. However, it also includes weights of initial layers which are truly not representing high-quality features as we already discussed.
- Adding activations of the last two and three layers: In this strategy, we add the activations of the last two layers and last three layers of the MuRIL model and pass these activations to the subsequent fully connected layer and CRF layer for the classification task. We named these models MuRIL-CRF$_{sum_2_layer}$ and MuRIL-CRF$_{sum_3_layer}$ respectively. Both models are fine-tuned for the Hindi NER task.
- Concatenating activations of the last two, three and four layers: Another idea is to concatenate the activations of the last two layers, three layers and four layers instead of adding them. The representation of an input token at each layer of MuRIL is indicated by a 768-dimensional vector. If the last four layers' representations are concatenated, then it becomes a 3072-dimensional vector. This high dimensional feature vector is passed to the subsequent layer in MuRIL-CRF architecture. We named these models MuRIL-CRF$_{concate_2_layer}$, MuRIL-CRF$_{concate_3_layer}$ and MuRIL-CRF$_{concate_4_layer}$. However, this strategy scored less F1-score than the MuRIL-CRF$_{sum_4_layer}$ model.

Besides that, we develop similar Hindi NER systems using another breakthrough language model, called mBERT to analyze the efficiency of both language models over the Hindi NER task. The mBERT is a multilingual version of the BERT-base model comprising of 12 encoder layers with 12 self-attention heads and 768 hidden layer dimensions. First, we develop a model named mBERT$_{base_classifier}$, where the mBERT model is integrated as an input layer and classification is performed by a fully connected layer with softmax classification.

Similar to the MuRIL-CRF model, we extend the mBERT$_{base_classifier}$ model and add one more layer i.e. CRF layer as an output layer for the classification task and named it as the mBERT-CRF model. Both models are fine-tuned for the Hindi NER task and the performance of mBERT$_{base_classifier}$ and mBERT-CRF models are compared with MuRIL$_{base_classifier}$ and MuRIL-CRF models respectively.

Further, we modify the mBERT-CRF model in the same way as the MuRIL-CRF$_{sum_4_layer}$ model i.e. add the activations of the last four layers of the mBERT language model and named the model as the mBERT-CRF$_{sum_4_layer}$ model. These final feature vectors are passed through the subsequent fully connected layer and then the final output tag sequence is produced by the CRF layer.

Next, we create similar variants of the mBERT-CRF$_{sum_4_layer}$ model by applying similar computations as applied in the above MuRIL-based variants and named these variants as mBERT-CRF$_{eleventh_layer}$, mBERT-CRF$_{concate_2_layer}$, mBERT-CRF$_{concate_3_layer}$, mBERT-CRF$_{concate_4_layer}$, mBERT-CRF$_{sum_12_layer}$, mBERT-CRF$_{sum_2_layer}$ and mBERT-CRF$_{sum_3_layer}$ model. The experimental results of all these variants are also reported in Table 4. As per results, it can be observed that the proposed MuRIL-CRF$_{sum_4_layer}$ model and its variants outperformed all mBERT-based experimental models. Next, we would discuss the hyperparameter selection for the mBERT-CRF model and MuRIL-CRF model.

6. Experimental settings

In this section, we would discuss the dataset used for all models, performance measures used for the evaluation and selection of

hyperparameters.

6.1. Hyperparameter tuning

For this research work, we used TensorBoard HParam API to get the optimal values of various hyperparameters such as batch size, learning rate, number of epochs and dropout rate. In all training episodes, we perform the experiments with fixed random seed since different random seeds may lead to a substantially different result. The optimal values of hyperparameters for the mBERT-CRF model are shown in Table 2 and for the MuRIL-CRF model are shown in Table 3.

For both mBERT-CRF and MuRIL-CRF models, we chose AdamW optimizer as mBERT and MuRIL both pre-trained models are fine-tuned well on this optimizer (Devlin et al., 2019; Khanuja et al., 2021). AdamW optimizer implements Adam algorithm with weight decay. Here, we just tuned the learning rate of this optimizer and fixed all other hyperparameters such as $\beta 1$, $\beta 2$ and weight_decay with their default value at 0.9, 0.999 and 0.01 respectively.

The different learning rates on which the MuRIL-CRF model and mBERT-CRF model experimented are 1e-5, 2e-5, 3e-5, 4e-5, 5e-5, 6e-5. Besides the learning rate, we also search for the optimal value of the dropout rate for both models. The different dropout rate on which we run the experiments are 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6. The performance of both the MuRIL-CRF and mBERT-CRF models with different dropout rates and learning rates is shown in Fig. 3(a) and Fig. 3(b) respectively. We run the experiments on different epochs within the range of 5 to 60 with an interval of 5 for the mBERT-CRF model and within range of 5 to 100 with the interval of 5 for the MuRIL-CRF model.

Finally, the performance of the $\text{MuRIL}_{\text{base\_classifier}}$, $\text{MuRIL-CRF}_{\text{sum\_4\_layer}}$ model and all its variants are also evaluated on the same hyperparameters as used for the MuRIL-CRF model. Similarly, the performance of the $\text{mBERT}_{\text{base\_classifier}}$, $\text{mBERT-CRF}_{\text{sum\_4\_layer}}$ model and all its variants are evaluated on the same hyperparameters as used for the mBERT-CRF model.

6.2. Dataset detail

We evaluate our work on ICON 2013 NLP tools contest dataset. The ICON 2013 corpus for the Hindi language contains 4478 Hindi sentences where every token in a sentence also has POS tag information and chunk information, with NE tag. In this work, we are intended to develop a robust NER system without asking for any external linguistic information and for this purpose we didn't use the available POS tag and chunk information. The class-wise distribution of tags in the training and test dataset is shown in Fig. 4. The corpus is then randomly sampled into three subsets: training, development and the test set. The training set contains a total of 3776 sentences comprising of 74,596 tokens, out of that 10% samples are used for the development set. The test set contains 702 samples comprising 13,616 tokens for evaluation purposes.

The dataset is broadly categorized into 3 types of named entities namely ENAMEX, NUMEX and TIMEX, where ENAMEX is further classified into 11 types of entity classes such as person, entertainment, location, organization, material, livthings, artifact, locomotive, plants, disease and facilities. NUMEX category grouped the entity classes related to numerical values such as distance, count, money and quantity. Similarly, TIMEX as the name implies grouped the entity classes related to time and duration such as period, time, date, day, month, sday and year.

In this work, we encode NE tags with the IOB2 tagging scheme. In IOB2, 'I' stands for inside, 'O' stands for outside and 'B' stands for the beginning of tag. In this scheme, B-tag shows the beginning of the token if it is tagged with any of the entity classes and I-prefix before a tag indicates the inside value of the tagged token. Here, I-tag would always trail B-tag. A token that does not belong to any of the mentioned classes, tagged with O. For example, सेट केथेड्रल टॉवर would be tagged as सेट/B-LOCATION, केथेड्रल/I-LOCATION and टॉवर/I-LOCATION and the word साथ would be tagged as O, as the word is not fit to any of the mentioned classes. Fig. 4(a) and 4(b) depicts the statistics of entity classes in training and test dataset respectively.

6.3. Performance measures

To evaluate the developed NER systems, we used the standard performance measures that are precision, recall and F1-score as described below: -

- Precision is measured as the number of the correct named entities extracted out of the total named entities extracted by the NER system.

Table 2
Hyperparameters for mBERT-CRF model.

S. No.	Hyperparameter	Value
1	Batch Size	32
2	Optimizer	AdamW
3	Learning rate	5e-5
4	Epoch	50
5	Dropout	0.5

Table 3
Hyperparameters for MuRIL-CRF model.

S. No.	Hyperparameter	Value
1	Batch Size	32
2	Optimizer	AdamW
3	Learning rate	5e-5
4	Epoch	75
5	Dropout	0.4

- Recall is defined as a ratio between the number of the correct named entities extracted by the NER system and the total named entities present in the corpus.
- F1-score is the harmonic mean of precision and recall. Usually, the balanced F1-score is used for evaluation.

7. Results and discussion

In this section, we would discuss the results achieved by each experimental model on the test set of the ICON 2013 Hindi NER dataset. The results for all mBERT-based and MuRIL-based experimental models are presented in Table 4 as well as it also summarizes the results achieved through previous methodologies on the same Hindi NER dataset for comparative analysis of the proposed approach. The results presented in Table 4 also imply that the proposed model achieved state-of-the-art F1-score of 85.77% without any linguistic rules, handcrafted features, gazetteers, POS and chunk information.

In this work, first, we evaluate $\text{MuRIL}_{\text{base\_classifier}}$ and MuRIL-CRF baseline models which achieve F1-score of 82.02% and 85.14% respectively and observe that both models outperform all existing NER systems evaluated on the ICON 2013 dataset. As can be observed from the results that MuRIL-CRF achieves +3.12 points more F1-score than $\text{MuRIL}_{\text{base\_classifier}}$ which attests that integrating the CRF layer certainly increases the results.

Afterward, we evaluate the proposed $\text{MuRIL-CRF}_{\text{sum\_4\_layer}}$ model that achieves 87.89% precision, 83.74% recall and 85.77% F1-score and we observe that the proposed model outperforms all existing Hindi NER models developed and evaluated on the ICON 2013 dataset as per results shown in Table 4. As can be observed from the results that the proposed model also outperforms both $\text{MuRIL}_{\text{base\_classifier}}$ and MuRIL-CRF models by achieving +3.75 point and +0.63 point more F1-score than $\text{MuRIL}_{\text{base\_classifier}}$ and MuRIL-CRF models respectively.

Next, we evaluate the different variants of the proposed $\text{MuRIL-CRF}_{\text{sum\_4\_layer}}$ model. In this series, first, the $\text{MuRIL-CRF}_{\text{eleventh\_layer}}$ model is evaluated that uses activations from the eleventh layer of the MuRIL language model and achieves an F1-score of 84.13% as shown in Table 4. The MuRIL-CRF model achieves +1.01 point more F1-score than the $\text{MuRIL-CRF}_{\text{eleventh\_layer}}$ model which endorses that representation of a token gets enriched layer by layer. Next, we evaluate the $\text{MuRIL-CRF}_{\text{concate\_2\_layer}}$, $\text{MuRIL-CRF}_{\text{concate\_3\_layer}}$ and $\text{MuRIL-CRF}_{\text{concat\_4\_layer}}$ models. These models achieve a higher F1-score than the MuRIL-CRF model wherein $\text{MuRIL-CRF}_{\text{concat\_2\_layer}}$, $\text{MuRIL-CRF}_{\text{concat\_3\_layer}}$ and $\text{MuRIL-CRF}_{\text{concat\_4\_layer}}$ models achieve +0.45 point, +0.53 point and +0.56 point more F1-score than MuRIL-CRF model respectively. Among these models, the $\text{MuRIL-CRF}_{\text{concat\_4\_layer}}$ model achieves the highest F1-score of 85.70%, which implies that concatenation of the last four layers is better than a concatenation of the last two and three layers. However, the proposed $\text{MuRIL-CRF}_{\text{sum\_4\_layer}}$ model outperforms $\text{MuRIL-CRF}_{\text{concat\_2\_layer}}$, $\text{MuRIL-CRF}_{\text{concat\_3\_layer}}$ and $\text{MuRIL-CRF}_{\text{concat\_4\_layer}}$ models. Further, we evaluate $\text{MuRIL-CRF}_{\text{sum\_12\_layer,}}$ $\text{MuRIL-CRF}_{\text{sum\_2\_layer}}$ and $\text{MuRIL-CRF}_{\text{sum\_3\_layer}}$ models and observe that these models also achieve higher F1-score than the MuRIL-CRF model. However, the proposed $\text{MuRIL-CRF}_{\text{sum\_4\_layer}}$ model achieves +0.05 point, +0.03 point and +0.11 point more F1-score than $\text{MuRIL-CRF}_{\text{sum\_12\_layer,}}$ $\text{MuRIL-CRF}_{\text{sum\_2\_layer}}$ and $\text{MuRIL-CRF}_{\text{sum\_3\_layer}}$ model respectively. These results infer that the upper layers generate rich contextual feature vectors which produce better output. Consequently, the proposed model outperforms all these variants by achieving the highest F1-score of 85.77%.

Further, we evaluate $\text{mBERT}_{\text{base\_classifier}}$ and mBERT-CRF models for the Hindi NER task. The $\text{mBERT}_{\text{base\_classifier}}$ model achieves an F1-score of 79.02% i.e. 3.00 point less F1-score than $\text{MuRIL}_{\text{base\_classifier}}$ and the mBERT-CRF model achieves an F1-score of 79.61% i.e. 5.53 point F1-score less than the MuRIL-CRF model. As can be observed from the results presented in Table 4 that the mBERT-CRF model achieves a higher F1-score than existing NER models except Sarkar (2018) model that attained an F1-score of 81.98% i.e. +2.37 point more F1-score than the mBERT-CRF model. However, the mBERT-CRF model achieves higher precision than Sarkar (Sarkar, 2018) model.

Like the $\text{MuRIL-CRF}_{\text{sum\_4\_layer}}$ model, we develop and evaluate the $\text{mBERT-CRF}_{\text{sum\_4\_layer}}$ model and its variants namely $\text{mBERT-CRF}_{\text{eleventh\_layer}}$, $\text{mBERT-CRF}_{\text{concate\_2\_layer}}$, $\text{mBERT-CRF}_{\text{concate\_3\_layer}}$, $\text{mBERT-CRF}_{\text{conate\_4\_layer}}$, $\text{mBERT-CRF}_{\text{sum\_12\_layer}}$, $\text{mBERT-CRF}_{\text{sum\_2\_layer}}$ and $\text{mBERT-CRF}_{\text{sum\_3\_layer}}$ model. As per the results shown in Table 4, the $\text{mBERT-CRF}_{\text{sum\_4\_layer}}$ model achieves the highest F1-score of 80.35% among all its variants. However, the F1-score achieved by the $\text{mBERT-CRF}_{\text{sum\_4\_layer}}$ model is 5.42 points less than the $\text{MuRIL-CRF}_{\text{sum\_4\_layer}}$ model. The results shown in Table 4 conclude that MuRIL-based NER models outperform mBERT-based NER models for the Hindi NER task.

7.1. Entity-wise classification results

This section presents the entity-wise F1-score achieved by MuRIL-based baseline models, proposed $\text{MuRIL-CRF}_{\text{sum\_4\_layer}}$ model as well as all its variants in Table 5. Similarly, Table 6 reports the F1-score achieved by each entity in all mentioned mBERT-based models.

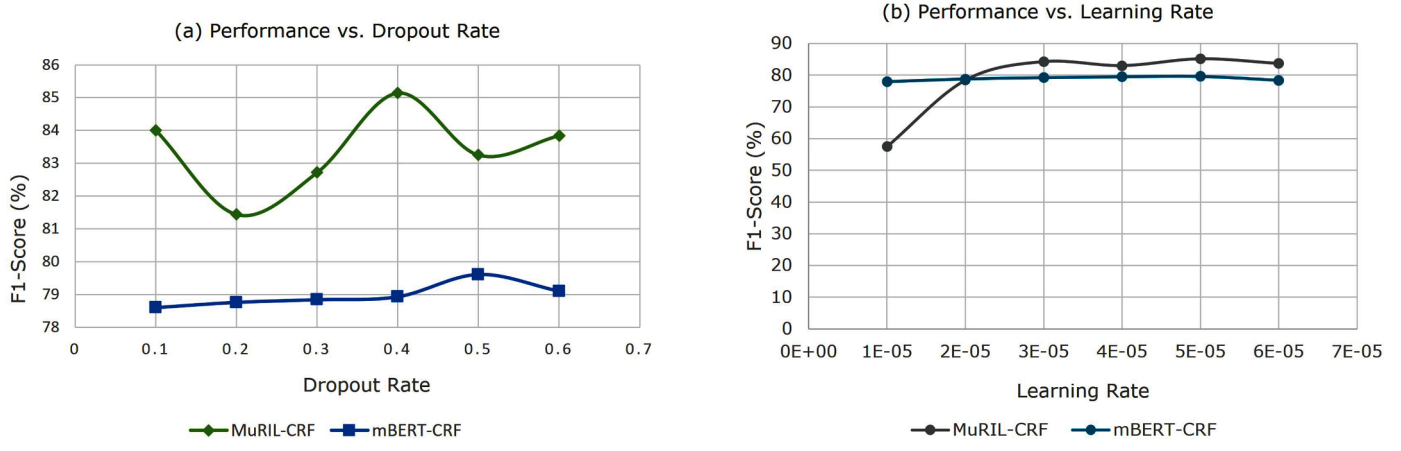

Fig. 3. Performance of MuRIL-CRF and mBERT-CRF model w.r.t different hyperparameters.

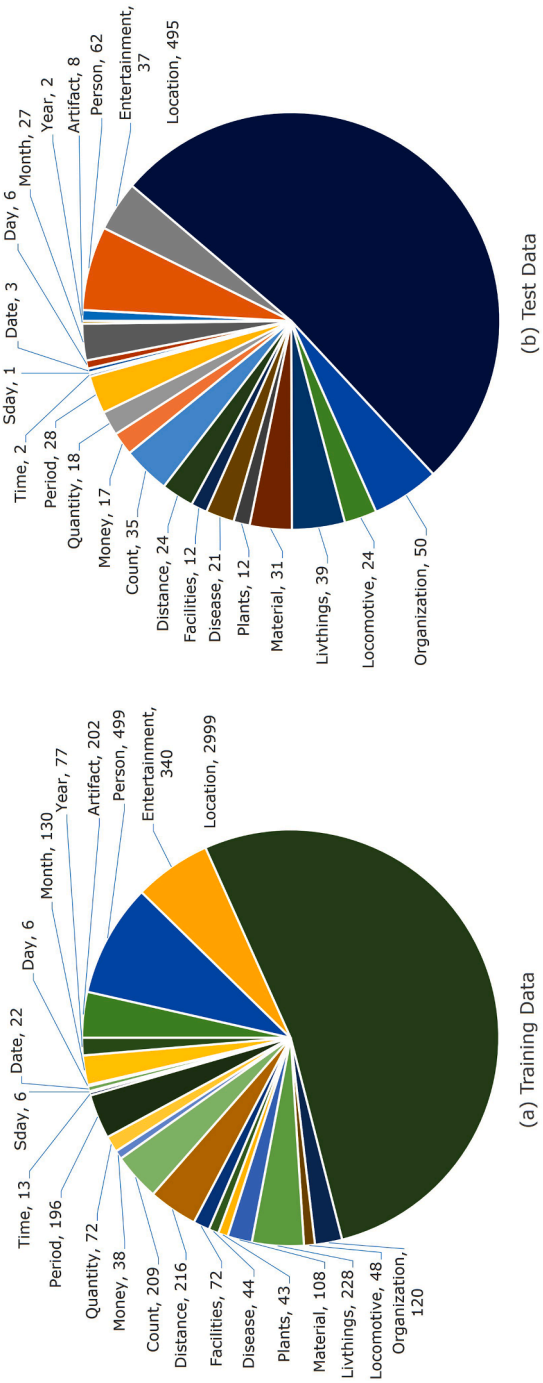

Fig. 4. Training and test data details.

Table 4
Results of our experimental models compared with existing Hindi NER systems on the ICON 2013 dataset.

Model	Classifier	Resources used	Precision (%)	Recall (%)	F1-Score (%)	NE-tags
(Athavale et al., 2016)	Bi-LSTM	POS tag information	75.86	79.17	77.48	NE-tags:11 (artifact, entertainment, person, location, livthings, facilities, locomotive, materials, diseases organization, plants)
(Sharma et al., 2020)	Bi-LSTM-CNN—CRF	No linguistic resource	69.00	71.00	70.00	NE-tags:9 (person, entertainment, count, location, livthings, organization, distance, period, month)
(Devi et al., 2013)	CRF	linguistic rules and features	77.52	77.36	77.44	NE-tags:22 (artifact, entertainment, date, location, livthings, facilities, locomotive, materials, person, organization, plants, count, disease, day, distance, money, year, day, sday, quantity, period, time)
(Das and Garain, 2014)	CRF	linguistic features and gazetteers	84.81	74.97	79.59	
(Sarkar and Shaw, 2016)	KNN	linguistic rules, features and gazetteers	80.13	76.67	78.37	
(Sarkar, 2018)	Ensemble approach with majority voting on 3 models (KNN, weighted KNN and HMM)	linguistic rules, features and gazetteers	82.30	81.67	81.98	
(Gayen and Sarkar, 2014)	HMM	POS tag, chunk information	75.40	75.00	75.20	
This work*	$mBERT_{base\_classifier}$	No linguistic resource	82.96 (83.11, ±0.68)	75.44 (74.64, ±0.64)	79.02 (78.64, ±0.31)	NE-tags:22 (artifact, entertainment, date, location, livthings, facilities, locomotive, materials, person, organization, plants, count, disease, day, distance, money, year, day, sday, quantity, period, time)
	mBERT-CRF		83.49 (83.25, ±0.55)	76.06 (75.27, ±0.49)	79.61 (79.06, ±0.39)	
	$mBERT\text{-}CRF_{eleventh\_layer}$		85.39 (84.15, ±0.95)	74.46 (74.36, ±0.83)	79.55 (78.95, ±0.51)	
	$mBERT\text{-}CRF_{concat\_2\_layer}$		84.20 (84.37, ±0.13)	76.06 (75.17, ±0.61)	79.93 (79.51, ±0.34)	
	$mBERT\text{-}CRF_{concat\_3\_layer}$		84.40 (83.54, ±0.66)	75.72 (75.60, ±0.72	79.82 (79.36, ±0.36)	
	$mBERT\text{-}CRF_{concat\_4\_layer}$		84.79 (84.58, ±0.43)	75.85 (75.07, ±0.46)	80.07 (79.54, ±0.28)	
	$mBERT\text{-}CRF_{sum\_12\_layer}$		82.57 (81.74, ±0.50)	73.73 (73.98, ±0.22)	77.90 (77.66, ±0.18)	
	$mBERT\text{-}CRF_{sum\_2\_layer}$		84.08 (84.19, ±0.20)	76.45 (75.38, ±0.65)	80.08 (79.53, ±0.31)	
	$mBERT\text{-}CRF_{sum\_3\_layer}$		84.36 (84.30, ±0.50)	76.41 (75.67, ±0.99)	80.19 (79.74, ±0.38)	
	$mBERT\text{-}CRF_{sum\_4\_layer}$		**85.18 (84.63, ±0.97)**	**76.03 (75.42, ±0.55)**	**80.35 (79.75, ±0.42)**	
	$MuRIL_{base\_classifier}$		82.34 (82.39, ±0.20)	81.70 (81.05, ±0.51)	82.02 (81.72, ±0.31)	
	MuRIL-CRF		87.11 (86.31, ±0.69)	83.25 (82.92, ±0.51)	85.14 (84.58, ±0.28)	
	$MuRIL\text{-}CRF_{eleventh\_layer}$		85.86 (85.62, ±0.52)	82.48 (82.24, ±0.32)	84.13 (83.89, ±0.26)	
	$MuRIL\text{-}CRF_{concat\_2\_layer}$		87.44 (86.70, ±0.51)	83.81 (83.61, ±0.37)	85.59 (85.13, ±0.32)	
	$MuRIL\text{-}CRF_{concat\_3\_layer}$		87.46 (86.73, ±0.48)	83.95 (83.79, ±0.60)	85.67 (85.23, ±0.28)	

(*continued on next page*)

Table 4 (*continued*)

Model	Classifier	Resources used	Precision (%)	Recall (%)	F1-Score (%)	NE-tags
	MuRIL-CRF$_{concat_4_layer}$		86.85 (86.28, ±0.60)	84.59 (84.22, ±0.45)	85.70 (85.24, ±0.48)	
	MuRIL-CRF$_{sum_12_layer}$		86.97 (86.62, ±0.53)	84.52 (83.76, ±0.69)	85.72 (85.16, ±0.45)	
	MuRIL-CRF$_{sum_2_layer}$		87.08 (86.98, ±0.48)	84.45 (83.56, ±0.87)	85.74 (85.24, ±0.57)	
	MuRIL-CRF$_{sum_3_layer}$		86.18 (86.39, ±0.74)	85.15 (84.06, ±0.67)	85.66 (85.21, ±0.40)	
	MuRIL-CRF$_{sum_4_layer}$ **(proposed)**		**87.89 (86.75, ±0.70)**	**83.74 (84.20, ±0.30)**	**85.77 (85.46, ±0.23)**	

[*] The best performance of each model is reported as well as the average score and standard deviation of 10 successful runs are depicted in brackets.

In addition, a comparative analysis of entity-wise classification results achieved by the mBERT-CRF$_{sum_4_layer}$ and MuRIL-CRF$_{sum_4_layer}$ model is shown in Fig. 5. The presented results report the efficacy of the MuRIL language model over the mBERT language model for Indian languages. Among 22 entity classes, MuRIL-based NER models achieved a higher F1-score for most entities. As per the analysis of results, we found that the proposed MuRIL-CRF$_{sum_4_layer}$ achieved a higher F1-score than the mBERT-CRF$_{sum_4_layer}$ model for several entities such as count (+14.98 point), date (+6.67 point), day (+9.52 point), disease (+22.35 point), entertainment (+5.81 point), facilities (+29.14 point), livthings (+17.98 point), location (+3.63 point), locomotive (+4.44 point), materials (+22.01 point), month (+1.41 point) organization (+8.05 point), period (+4.46 point), person (+6.46 point), plants (+23.38 point), quantity (+5.95 point) and sday (+66.67 point). It can be noticed that count, disease, facilities, livthings, materials, plants and sday entities achieved excellent improvement in their F1-score compared to mBERT-CRF$_{sum_4_layer}$ model.

Besides that, other entities also achieved impressive F1-score as shown in Fig. 5 except artifact, money and year which achieved respectively 7.59 points, 9.34 points and 6.15 points less F1-score than the mBERT-CRF$_{sum_4_layer}$ model. Although, both mBERT and MuRIL are based on 12-layered encoder architecture, MuRIL-based NER systems outperformed mBERT-based NER models for the Hindi language, as per results shown in Table 5 and Table 6.

The reasons behind this performance gap are:

Table 5
Entity-wise F1-score (%) achieved on MuRIL-based baseline models, proposed model and its variants.[d]

Entity	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
Artifact	10.39	8.89	15.38	15.79	19.35	15.79	15.79	16.67	13.33	14.63
Count	79.01	84.62	93.51	93.33	89.47	85.33	82.05	94.74	90.67	88.31
Date	28.57	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00
Day	0.00	92.31	100	100	100	100	100	100	100	100
Disease	21.82	20.41	28.57	16.33	48.39	53.12	50.00	40.00	48.39	40.74
Distance	100	96.77	100	98.41	96.88	100	98.41	100	100	98.41
Entertainment	75.00	79.07	77.08	80.43	77.65	74.16	77.65	78.57	76.60	77.65
Facilities	54.55	55.81	42.86	47.83	51.43	43.90	58.54	47.37	63.83	57.14
Livthings	88.29	92.31	90.76	96.55	96.55	94.74	97.39	94.74	95.65	97.39
Location	90.57	92.00	91.09	92.27	91.28	91.64	92.26	91.46	91.58	91.88
Locomotive	81.36	78.69	84.21	84.75	79.37	83.33	86.21	84.21	84.75	81.97
Materials	65.12	72.94	60.00	68.29	54.76	70.00	60.53	56.47	62.22	65.85
Money	56.00	60.00	52.00	58.82	53.06	53.06	44.44	50.98	53.06	54.55
Month	94.74	94.74	93.10	93.10	96.43	94.74	94.74	93.10	94.74	94.74
Organization	76.82	77.85	72.73	72.48	82.67	80.79	79.17	74.51	80.75	80.00
Period	93.55	93.55	96.77	95.08	95.24	92.06	90.91	93.55	95.24	96.77
Person	75.58	76.57	75.14	77.46	80.72	77.46	74.12	87.06	83.04	78.86
Plants	85.71	96.30	96.30	96.15	85.11	96.30	100	100	85.71	88.00
Quantity	87.80	90.00	90.00	89.47	89.47	94.74	89.47	92.31	87.80	92.31
Sday	0.00	100	50.00	66.67	66.67	50.00	66.67	50.00	50.00	66.67
Time	50.00	100	100	100	100	100	100	100	100	100
Year	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00
Avg/total	**82.02**	**85.14**	**84.13**	**85.59**	**85.67**	**85.70**	**85.72**	**85.74**	**85.66**	**85.77**

[d] M1: MuRIL$_{base_classifer}$, M2: MuRIL-CRF, M3: MuRIL-CRF$_{eleventh_layer}$, M4: MuRIL-CRF$_{concat_2_layer}$, M5: MuRIL-CRF$_{concat_3_layer}$, M6: MuRIL-CRF$_{concat_4_layer}$, M7: MuRIL-CRF$_{sum_12_layer}$, M8: MuRIL-CRF$_{sum_2_layer}$, M9: MuRIL-CRF$_{sum_3_layer}$, M10: MuRIL-CRF$_{sum_4_layer}$

Table 6
Entity-wise F1-score (%) achieved on mBERT-based baseline models, proposed model and its variants.[a]

Entity	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10
Artifact	11.76	8.33	12.50	14.16	11.43	9.09	18.39	13.56	9.52	22.22
Count	75.29	77.27	71.91	75.29	80.46	76.40	80.90	76.92	78.65	73.33
Date	40.00	40.00	80.00	40.00	40.00	40.00	80.00	80.00	40.00	33.33
Day	78.95	90.48	85.00	87.80	90.48	90.20	60.61	90.48	78.95	90.48
Disease	13.33	7.41	23.40	16.67	13.64	23.16	7.32	20.22	21.98	18.39
Distance	99.07	100	98.15	99.07	95.15	100	100	99.07	100	98.15
Entertainment	81.08	74.17	81.45	76.36	75.21	74.34	76.39	86.11	67.22	71.84
Facilities	36.07	46.91	40.74	39.29	39.34	36.62	35.96	46.38	50.00	28.00
Livthings	76.33	83.17	81.77	83.25	79.19	76.92	79.59	77.23	77.32	79.41
Location	86.41	87.44	86.55	87.88	88.34	88.08	85.17	86.59	88.13	88.25
Locomotive	75.14	79.55	74.74	80.87	73.37	80.90	77.42	82.05	77.95	77.53
Materials	44.97	36.24	36.36	33.73	42.53	41.67	35.42	39.52	44.16	43.84
Money	57.89	64.79	70.13	59.15	61.54	64.00	78.38	65.71	66.67	63.89
Month	96.55	96.55	96.55	95.45	96.55	96.55	97.67	96.55	96.47	93.33
Organization	77.74	68.53	76.49	78.15	68.59	76.98	76.29	75.24	80.97	71.95
Period	92.31	91.14	87.80	91.14	91.14	88.89	93.51	90.00	90.00	92.31
Person	68.72	75.06	69.33	73.25	73.20	71.36	70.87	69.70	69.90	72.40
Plants	50.85	70.59	58.06	58.06	54.55	50.00	62.50	63.64	49.18	64.62
Quantity	93.62	88.37	85.11	91.67	95.65	97.78	89.80	88.89	93.62	86.36
Sday	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100	0.00	0.00
Time	100	100	100	100	100	100	100	100	100	100
Year	46.15	46.15	46.15	46.15	46.15	46.15	46.15	46.15	46.15	46.15
Avg/total	**79.02**	**79.61**	**79.55**	**79.93**	**79.82**	**80.07**	**77.90**	**80.08**	**80.19**	**80.35**

[a] B1: $\text{mBERT}_{\text{base\_classifier}}$, B2: mBERT-CRF, B3: $\text{mBERT-CRF}_{\text{eleventh\_layer}}$, B4: $\text{mBERT-CRF}_{\text{concat\_2\_layer}}$, B5: $\text{mBERT-CRF}_{\text{concat\_3\_layer}}$, B6: $\text{mBERT-CRF}_{\text{concat\_4\_layer}}$, B7: $\text{mBERT-CRF}_{\text{sum\_12\_layer}}$, B8: $\text{mBERT-CRF}_{\text{sum\_2\_layer}}$, B9: $\text{mBERT-CRF}_{\text{sum\_3\_layer}}$, B10: $\text{mBERT-CRF}_{\text{sum\_4\_layer}}$

- Since mBERT is trained on 100+ languages, this leads to a small representation of Indian languages in their vocabulary and training data. While MuRIL is specially built for Indian languages and trained on significantly massive text data of 16 Indian languages. So, it can capture all linguistics of Indian languages which could be difficult for mBERT.
- As MuRIL is trained on transliterated data as well, it significantly handles transliterated text present in the Indian language as compared to mBERT.
- mBERT and MuRIL both use MLM as their pre-training objective, however, in mBERT, MLM uses WordPiece masking whereas, in MuRIL, it uses the WholeWord masking approach which reflects their results.
- In WordPiece masking, WordPiece tokens are randomly selected for masking. For example, wordpiece tokens associated with a word 'डहिाइड्रेट' are डि##हा ##इड ##्रेट. In WordPeice masking random wordpiece tokens are masked such as डि##हा ##इड ##्रेट → [MASK] ##हा ##इड [MASK] whereas in WholeWord masking all wordpiece tokens associated with a word are masked altogether like डि##हा ##इड ##्रेट → [MASK] [MASK] [MASK] [MASK]. Thus, the WholeWord masking makes it easy to recover the whole word in the MLM pre-training task.

We also analyzed the performance of both $\text{MuRIL-CRF}_{\text{sum\_4\_layer}}$ and $\text{mBERT-CRF}_{\text{sum\_4\_layer}}$ models with varying amount of training samples which is demonstrated as a learning curve in Fig. 6.

As can be observed, the learning curve of the $\text{MuRIL-CRF}_{\text{sum\_4\_layer}}$ model has plateaued out at around 50% of training samples with an F1-score of 83%, which indicates that we have sufficient training samples for the proposed model to perform well and the performance is not further improving with the increased number of training samples.

7.2. Error analysis

Further, to analyze the performance of the proposed $\text{MuRIL-CRF}_{\text{sum\_4\_layer}}$ model, we discuss the outcome yielded by the proposed model on some random samples taken from the test set and compare these annotations with gold-standard annotations. We also present a comparative analysis of these resultant annotations with the annotations produced by the $\text{mBERT-CRF}_{\text{sum\_4\_layer}}$ model for the same samples. Table 7 presents the comparative analysis of the resultant annotations yielded by both models for these samples.

In the first sentence, the proposed system could identify both named entity 'बुधवार' and '५८ डॉलर' and could classify them correctly. It can be observed from Table 5 that the proposed model achieved 100% F1-score for entity class 'day' even when the train and test set have only 6 'day' entities each, as per statistics shown in Fig. 4. It shows the efficiency of the MuRIL pre-trained language model that transferred its knowledge and enhanced the training of the proposed system. While the $\text{mBERT-CRF}_{\text{sum\_4\_layer}}$ model couldn't classify 'बुधवार' as 'day' however, it could classify '५८ डॉलर' correctly. Next, the second sentence has a named entity 'सारस', the proposed model could classify it correctly as 'livthings' while the $\text{mBERT-CRF}_{\text{sum\_4\_layer}}$ model classified it as 'O'. Similarly, in the third sentence, the entity 'कगिफशिर रेड एयरलाइन' is identified and classified correctly as 'locomotive' by both models.

Next consider sentences 4, 5, 6, 7 and 8, the results are remarkable as it can be noticed that the proposed model could classify even ambiguous named entities.

In sentences 4 and 5, the ambiguous entity is 'टहिरी' which is correctly classified as a 'person' in sentence 4 while classified as a

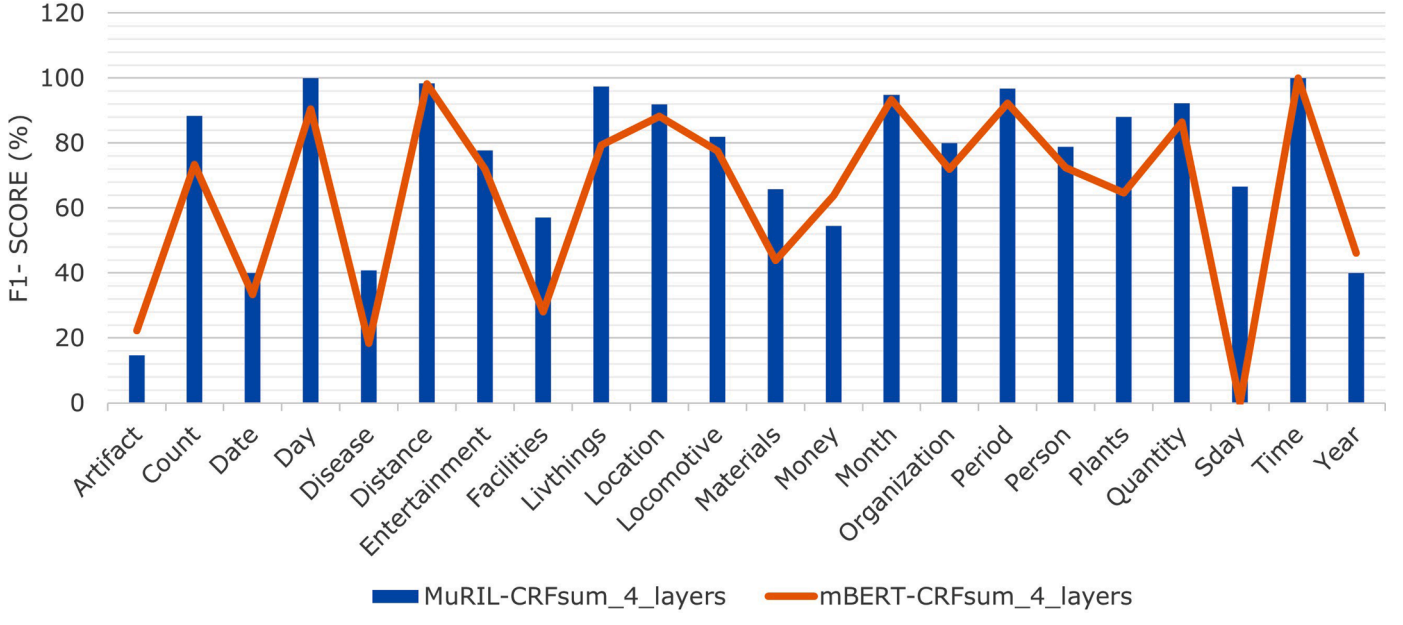

Fig. 5. Entity-wise classification performance of models.

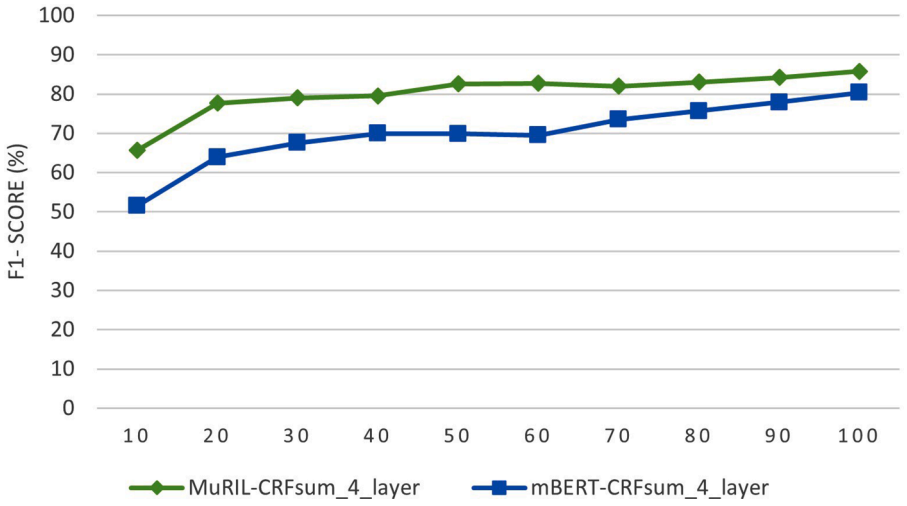

Fig. 6. Learning curve.

Table 7
Comparative analysis of some samples from test corpus.

Sr. No	Sentence	Gold annotation	mBERT-CRF$_{sum_4_layer}$ model	MuRIL-CRF$_{sum_4_layer}$ model (proposed)	Remark
1.	मसलन, बुधवार को कच्चे तेल की कीमतें ५८ डॉलर तक गिर गईं.	['O', 'B-DAY', 'O', 'O', 'O', 'O', 'O', 'B-MONEY', 'I-MONEY', 'O', 'O', 'O', 'O']	['O', 'O', 'O', 'O', 'O', 'O', 'O', 'B-MONEY', 'I-MONEY', 'O', 'O', 'O', 'O']	['O', 'B-DAY', 'O', 'O', 'O', 'O', 'O', 'B-MONEY', 'I-MONEY', 'O', 'O', 'O', 'O']	'बुधवार' is not classified by the mBERT-CRF$_{sum_4_layer}$ model.
2.	ये सारस इसी क्षेत्र में डेरा डालते हैं और इस दौरान इन्हें देखना वाकई एक अलग अनुभव है.	[O, B-LIVTHINGS, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O]	[O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O]	[O, B-LIVTHINGS, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O]	'सारस' is not classified by the mBERT-CRF$_{sum_4_layer}$ model.
3.	कगिफशिर रेड एयरलाइन के लिए किसी तरह की मंदी नहीं है.	[B-LOCOMOTIVE, I-LOCOMOTIVE, I-LOCOMOTIVE, O, O, O, O, O, O, O, O, O]	[B-LOCOMOTIVE, I-LOCOMOTIVE, I-LOCOMOTIVE, O, O, O, O, O, O, O, O, O]	[B-LOCOMOTIVE, I-LOCOMOTIVE, I-LOCOMOTIVE, O, O, O, O, O, O, O, O, O]	All entities are correctly classified.
4.	यहां टहिरी नरेश की इस्पेक्शन बल्डिगि होती थी.	[O, B-PERSON, I-PERSON, O, O, O, O, O, O]	[O, B-PERSON, O, O, O, O, O, O, O]	[O, B-PERSON, I-PERSON, O, O, O, O, O, O]	'नरेश' is not classified as I-Person by the mBERT-CRF$_{sum_4_layer}$ model.
5.	मसूरी से लगभग 30 कि.मी. दूर टहिरी मार्ग पर स्थति है शांत एवं सुरम्य पर्वत स्थल "धनौल्टी".	[B-LOCATION, O, O, B-DISTANCE, I-DISTANCE, O, B-LOCATION, I-LOCATION, O, O, O, O, O, O, O, O, O, B-LOCATION, O, O]	[B-LOCATION, O, O, B-DISTANCE, I-DISTANCE, O, B-LOCATION, I-LOCATION, O, O, O, O, O, O, O, O, O, B-LOCATION, O, O]	[B-LOCATION, O, O, B-DISTANCE, I-DISTANCE, O, B-LOCATION, I-LOCATION, O, O, O, O, O, O, O, O, O, B-LOCATION, O, O]	All entities are correctly classified and both models disambiguate 'टहिरी' as 'location'.
6.	गंगा नदी 243 मीटर ऊंचाई पर है.	['B-LOCATION', 'I-LOCATION', 'B-DISTANCE', 'I-DISTANCE', 'O', 'O', 'O', 'O']	['B-LOCATION', 'I-LOCATION', 'B-DISTANCE', 'I-DISTANCE', 'O', 'O', 'O', 'O']	['B-LOCATION', 'I-LOCATION', 'B-DISTANCE', 'I-DISTANCE', 'O', 'O', 'O', 'O']	All entities are correctly classified.
7.	गंगा ने कॉफी बनाई.	['B-PERSON', 'O', 'B-MATERIALS', 'O', 'O']	['B-LOCATION', 'O', 'O', 'O', 'O']	['B-PERSON', 'O', 'B-MATERIALS', 'O', 'O']	'कॉफी' is not classified and 'गंगा' couldn't be disambiguated as 'person' by the mBERT-CRF$_{sum_4_layer}$ model.
8.	यहां भूटान का महत्वाकांक्षी भूटान हाइड्रोइलैक्ट्रकि प्रोजेक्ट चल रहा है.	[O, B-LOCATION, O, O, B-FACILITIES, I-FACILITIES, I-FACILITIES, O, O, O, O]	[O, B-LOCATION, O, O, B-LOCATION, O, O, O, O, O, O]	[O, B-LOCATION, O, O, B-FACILITIES, I-FACILITIES, I-FACILITIES, O, O, O, O]	mBERT-CRF$_{sum_4_layer}$ couldn't resolve ambiguity present in 'भूटान हाइड्रोइलैक्ट्रकि प्रोजेक्ट' entity.
9.	हवाई यात्रा के समय डकिॉनजस्टिेड ले.	[O, O, O, O, B-ARTIFACT, O, O]	[O, O, O, O, B-ARTIFACT, O, O]	[O, O, O, O, O, O, O]	'डकिॉनजस्टिेड' is not classified by the proposed model.
10.	इससे डहिाइड्रेशन नहीं होगा.	[O, B-DISEASE, O, O, O]	[O, O, O, O, O]	[O, O, O, O, O]	'डहिाइड्रेशन' is not classified by both models.

'location' in sentence 5 by both models. This implies that this ambiguity is resolved by both models efficiently. Similarly, in sentences 6 and 7, the ambiguous entity is 'गंगा' which is correctly classified by the proposed model as a 'location' in sentence 6 and as a 'person' in sentence 7. While the mBERT-CRF$_{sum_4_layer}$ model couldn't disambiguate the entity 'गंगा' and classified it as 'location' in both sentences. Besides that, these sentences also have two other entities like '243 मीटर' and 'कॉफी' which are correctly labeled by the proposed system as 'distance' and 'material' respectively whereas the mBERT-CRF$_{sum_4_layer}$ couldn't classify 'कॉफी' as 'material'.

Further, consider sentence 8 which have two named entities such as 'भूटान', and 'भूटान हाइड्रोइलैक्ट्रकि प्रोजेक्ट' with a common word 'भूटान'. As per gold-standard annotation, in the first entity 'भूटान' should be classified as a 'location' and in the second one it should be classified as a 'facility' due to part of nested named entity 'भूटान हाइड्रोइलैक्ट्रकि प्रोजेक्ट'. The proposed model could resolve this ambiguity and classify them correctly while the mBERT-CRF$_{sum_4_layer}$ model couldn't resolve this ambiguity. Thus, the results of sentences 4, 5, 6, 7 and 8 conclude that the MuRIL language model is much efficient compared to the mBERT language model to disambiguate the entities.

As can be observed from Table 5, the proposed model achieved an impressive F1-score for almost all named entities except artifact, date, disease and year. In the low-scored named entities, artifact scored only 14.53% F1-score that is the lowest among all, and disease scored 40.74% F1-score as shown in Table 5. It is also reflected by the ninth and tenth sentences in Table 7 where the proposed system misclassified the entities such as 'डकिॉनजस्टि्रेंड' and 'डहिाइड्रेशन' as 'O'.

After this deep analysis of results, we conclude that the performance of fine-tuned MuRIL-CRF$_{sum_4_layer}$ model is excellent compared to the mBERT-CRF$_{sum_4_layer}$ model, not just because it identified and classified most named entities correctly even it also resolved the ambiguity present in the named entities.

8. Conclusion

In this paper, we proposed a robust Hindi NER model which is developed by incorporating multilingual language model MuRIL and CRF. It is robust in the sense that, unlike other existing Hindi NER models, the presented system performed superior even without the use of any linguistic feature, rules, any POS information, and chunk information. The proposed model outperformed all experimental and existing Hindi NER models on the ICON 2013 dataset. In this work, we also investigated the mBERT language model with CRF algorithm on the same Hindi NER dataset. As per obtained results, the performance of the MuRIL language model is far better than the mBERT language model for the Hindi NER task on the ICON 2013 dataset.

The performance of the proposed model also verifies that incorporation of the MuRIL language model escalates the accuracy of the Hindi NER task even when training data is scarce. The proposed model is also able to classify ambiguous named entities in their correct mentions. Hence, this work is also able to resolve the named entity disambiguation problem substantially. We hope that this research work would benefit further NLP research on Indian languages. However, these multilingual models have their limitations towards the sequence length due to their full attention mechanism, as future research work, we would use Big Bird, a sparse attention mechanism that handles the very long sequences efficiently and drastically improves the performance of various NLP task.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

A P, A., K, M., Mary Idicula, S., 2019.. An improved word representation for deep learning based NER in Indian languages. Information 10. https://doi.org/10.3390/info10060186.

Arase, Y., Tsujii, J., 2021. Transfer fine-tuning of BERT with phrasal paraphrases. Comput. Speech Lang. 66, 101164 https://doi.org/10.1016/j.csl.2020.101164.

Arkhipov, M., Trofimova, M., Kuratov, Y., Sorokin, A., 2019. Tuning multilingual transformers for language-specific named entity recognition. In: Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing. Association for Computational Linguistics, Florence, Italy, pp. 89–93. https://doi.org/10.18653/v1/W19-3712.

Athavale, V., Bharadwaj, S., Pamecha, M., Prabhu, A., Shrivastava, M., 2016. Towards deep learning in Hindi {NER:} an approach to tackle the labelled data sparsity, in: Proceedings of the 13th International Conference on Natural Language Processing, {ICON} 2016, Varanasi, India, December 17-20, 2016. pp. 154–160.

Babych, B., Hartley, A., 2003. Improving machine translation quality with automatic named entity recognition. In: Proceedings of the 7th International EAMT Workshop on MT and Other Language Technology Tools, Improving MT through Other Language Technology Tools: Resources and Tools for Building MT, pp. 1–8.

Bharati, A., Sangal, R., Sharma, D., 2007. Ssf: shakti standard format guide.

Biswas, S., Mishra, M.K., Sitanath_biswas, S.A., Mohanty, S., 2010. A two stage language independent named entity recognition for Indian languages. IJCSIT). Int. J. Comput. Sci. Inf. Technol. 1, 285–289.

Chopra, D., Jahan, N., Morwal, S., 2012. Hindi named entity recognition by aggregating rule based heuristics and hidden Markov model. Int. J. Inf. 2.

Chopra, D., Joshi, N., Mathur, I., 2016. Named entity recognition in Hindi using hidden Markov model, in: 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT). pp. 581–586.

Clark, K., Luong, M.-.T., Le, Q.V., Manning, C.D., 2020. ELECTRA: pre-training text encoders as discriminators rather than generators.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V., 2019. Unsupervised Cross-lingual Representation Learning at Scale. arXiv Prepr. arXiv1911.02116.

Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z., Wang, S., Hu, G., 2019. Pre-training with whole word masking for Chinese {BERT}. CoRR abs/1906.0.

Das, A., Garain, U., 2014. CRF-based named entity recognition @ICON 2013. CoRR abs/1409.8.

Devi, G.R., Veena, P.V., Kumar, M.A., Soman, K.P., 2016. Entity extraction of Hindi-English and Tamil-English code-mixed social media text, in: Forum for Information Retrieval Evaluation. pp. 206–218.

Devi, S.L., Malarkodi, C.S., Marimuthu, K., Chrompet, C., 2013. Named entity recognizer for Indian languages (ICON NLP tool contest 2013), in: 10th International Conference on Natural Language Processing.

Devlin, J., Chang, M.-.W., Lee, K., Toutanova, K., 2019. {BERT}: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. 10.18653/v1/N19-1423.
Dimitrov, M., Bontcheva, K., Cunningham, H., Maynard, D., 2005. A light-weight approach to coreference resolution for named entities in text. Anaphora Process. Linguist. Comput. Model. 97–112. https://doi.org/10.1075/cilt.263.07dim.
Ekbal, A., Bandyopadhyay, S., 2010. Named entity recognition using support vector machine: a language independent approach. Int. J. Electr. Comput. Syst. Eng. 4, 155–170.
Ekbal, A., Bandyopadhyay, S., 2009. A conditional random field approach for named entity recognition in Bengali and Hindi. Linguist. Issues Lang. Technol. 2, 1–44.
Gali, K., Surana, H., Vaidya, A., Shishtla, P., Sharma, D.M., 2008. Aggregating machine learning and rule based heuristics for named entity recognition. In: Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages.
Gayen, V., Sarkar, K., 2014. An HMM based named entity recognition system for Indian languages: the JU system at ICON 2013. CoRR abs/1405.7.
Greenwood, M.A., Gaizauskas, R., 2003. Using a named entity tagger to generalise surface matching text patterns for question answering. In: Proceedings of the Workshop on Natural Language Processing for Question Answering (EACL03), pp. 29–34.
Gupta, D., Ekbal, A., Bhattacharyya, P., 2018. A deep neural network based approach for entity extraction in code-mixed Indian social media text. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018).
Hakala, K., Pyysalo, S., 2019. Biomedical named entity recognition with multilingual {BERT}. In: Proceedings of The 5th Workshop on BioNLP Open Shared Tasks. Association for Computational Linguistics, Hong Kong, China, pp. 56–61. https://doi.org/10.18653/v1/D19-5709.
Han, X., Zhao, J., 2010. Structural semantic relatedness: a knowledge-based method to named entity disambiguation. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Uppsala, Sweden, pp. 50–59.
Hoang, M., Bihorac, O.A., Rouces, J., 2019. Aspect-based sentiment analysis using {BERT}. In: Proceedings of the 22nd Nordic Conference on Computational Linguistics. Linköping University Electronic Press, Turku, Finland, pp. 187–196.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput 9, 1735–1780.
Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., Johnson, M., 2020. XTREME: a massively multilingual multi-task benchmark for evaluating cross-lingual generalization.
Huang, W.-.C., Wu, C.-.H., Luo, S.-.B., Chen, K.-.Y., Wang, H.-.M., Toda, T., 2021. Speech recognition by simply fine-tuning bert, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7343–7347. 10.1109/ICASSP39728.2021.9413668.
Kakwani, D., Kunchukuttan, A., Golla, S., N.C., G., Bhattacharyya, A., Khapra, M.M., Kumar, P., 2020. IndicNLPSuite: monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages, in: Findings of EMNLP.
Kaur, Y., Kaur, E.R., 2015. Named entity recognition (NER) system for Hindi language using combination of rule based approach and list look up approach. Int. J. Sci. Res. Manag. 3.
Khanuja, S., Bansal, D., Mehtani, S., Khosla, S., Dey, A., Gopalan, B., Margam, D.K., Aggarwal, P., Nagipogu, R.T., Dave, S., Gupta, S., Gali, S.C.B., Subramanian, V., Talukdar, P., 2021. MuRIL: multilingual representations for Indian languages.
Labusch, K., Neudecker, C., Zellhöfer, D., 2020. BERT for named entity recognition in contemporary and historical German, in: Proceedings of the 15th Conference on Natural Language Processing, KONVENS 2019. pp. 1–9.
Lafferty, J., McCallum, A., Pereira, F.C.N., 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data.
Lample, G., Conneau, A., 2019. Cross-lingual language model pretraining. Adv. Neural Inf. Process. Syst.
Liu, J., Xia, C., Li, X., Yan, H., Liu, T., 2020. A BERT-based ensemble model for Chinese news topic prediction. In: Proceedings of the 2020 2nd International Conference on Big Data Engineering, 2020. Association for Computing Machinery, New York, NY, USA, pp. 18–23. https://doi.org/10.1145/3404512.3404524. BDE.
Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. RoBERTa: {A} robustly optimized {BERT} pretraining approach. CoRR abs/1907.1.
Ma, J., Xie, S., Jin, M., Lianxin, J., Yang, M., Shen, J., 2020. {XSYSIGMA} at {S}em{E}val-2020 task 7: method for predicting headlines{'} humor based on auxiliary sentences with {EI}-{BERT}, in: Proceedings of the Fourteenth Workshop on Semantic Evaluation. International Committee for Computational Linguistics, Barcelona (online), pp. 1077–1084.
Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space, in: 1st International Conference on Learning Representations, {ICLR} 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings.
Moro, A., Raganato, A., Navigli, R., 2014. Entity linking meets word sense disambiguation: a unified approach. Trans. Assoc. Comput. Linguist. 2, 231–244. https://doi.org/10.1162/tacl_a_00179.
Morwal, S., Jahan, N., Chopra, D., 2012. Named entity recognition using hidden Markov model (HMM). Int. J. Nat. Lang. Comput. 1, 15–23.
Mukhin, E., 2020. Using Pre-trained Deeply Contextual Model BERT For Russian Named Entity Recognition, in: Analysis of Images, Social Networks and Texts. Springer International Publishing, Cham, pp. 167–173.
Murthy, V.R., 2017. Named entity recognition using deep learning. In: 14th International Conference on Natural Language Processing.
Nadeau, D., Sekine, S., 2007. A survey of named entity recognition and classification. Lingvisticae Investig 30, 3–26.
Pennington, J., Socher, R., Manning, C., 2014. {G}love: global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing ({EMNLP}). Association for Computational Linguistics, Doha, Qatar, pp. 1532–1543. 10.3115/v1/D14-1162.
Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J., 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. 21, 1–67.
Saha, S.K., Narayan, S., Sarkar, S., Mitra, P., 2010. A composite kernel for named entity recognition. Pattern Recognit. Lett. 31, 1591–1597.
Saha, S.K., Sarkar, S., Mitra, P., 2008. A hybrid feature set based maximum entropy Hindi named entity recognition, in: Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I.
Sarkar, K., 2018. Hindi named entity recognition using system combination. Int. J. Appl. Pattern Recognit. 5, 11–39.
Sarkar, K., Shaw, S., 2016. A memory-based learning approach for named entity recognition in Hindi. J. Intell. Syst. 26 https://doi.org/10.1515/jisys-2015-0010.
Shah, B., Kopparapu, S.K., 2019. A deep learning approach for Hindi named entity recognition. arXiv Prepr. arXiv1911.01421.
Sharma, R., Morwal, S., Agarwal, B., Chandra, R., Khan, M.S., 2020. A deep neural network-based model for named entity recognition for Hindi language. Neural Comput. Appl. 32 https://doi.org/10.1007/s00521-020-04881-z.
Singh, V., Vijay, D., Akhtar, S.S., Shrivastava, M., 2018. Named entity recognition for Hindi-english code-mixed social media text, in: Proceedings of the Seventh Named Entities Workshop. pp. 27–35.
Soricut, Z.L. and M.C. and S.G. and K.G. and P.S. and R., 2020. ALBERT: a lite BERT for self-supervised learning of language representations. arXiv Prepr. arXiv1909.11942.
Souza, F., Nogueira, R., Lotufo, R., 2019. Portuguese named entity recognition using BERT-CRF. arXiv Prepr. arXiv1909.10649.
Srivastava, S., Sanglikar, M., Kothari, D.C., 2011. Named entity recognition system for Hindi language: a hybrid approach. Int. J. Comput. Linguist. 2, 10–23.
Taylor, W.L., 1953. Cloze procedure": a new tool for measuring readability. J. Q. 30, 415–433.
Toda, H., Kataoka, R., 2005. A search result clustering method using informatively named entities. In: Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management, pp. 81–86.
Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C., 2020. mT5: {A} massively multilingual pre-trained text-to-text transformer. CoRR abs/2010 1.
Zhu, J., Xia, Y., Wu, L., He, D., Qin, T., Zhou, W., Li, H., Liu, T., 2020. Incorporating BERT into neural machine translation. In: International Conference on Learning Representations.

Richa Sharma is a Research Scholar in the field of Artificial Intelligence at Banasthali Vidyapith, India. Her research area includes natural language processing (NLP), machine learning and deep learning. She received her Master's in computer science with a specialization in machine learning from Banasthali Vidyapith. She has worked on exploring the methods for knowledge extraction from text using deep learning algorithms. She has worked as an Assistant Professor in the department of computer science & engineering at Global Institute of Technology, India. She has also experience in software development for more than 7 years in JAVA and other technologies.

Dr. Sudha Morwal is an active researcher in the field of artificial intelligence, data mining and natural language processing. She is currently working as an Associate Professor in the department of computer science and engineering at Banasthali Vidyapith, India. She has done her Ph.D. in Computer Science from Banasthali Vidyapith and published several research papers in reputed national and international journals.

Dr. Basant Agarwal is an Assistant Professor at Indian Institute of Information Technology Kota, India. He holds a Ph.D. in computer science and engineering from Malaviya National Institute of Technology, Jaipur, India. His research interests include natural language processing, machine learning, deep learning, sentiment analysis and Big data analytics. He has worked as a PostDoctoral Fellow at the department of computer science, Norwegian University of Science & Technology (NTNU), Norway. He has also worked as a Research Assistant at Temasek Laboratories, National University of Singapore (NUS), Singapore. He has published several research papers in international conferences and journals of repute.