ABSTRACT Linking event triggers with their respective arguments is an essential component for building an event extraction system. It is challenging to link event triggers with their corresponding argument triggers when the sentence contains multiple event and argument triggers. The task becomes even more challenging in a low-resource setup due to the unavailability of natural language processing resources and tools. In this paper, we study the event-argument linking task based on disaster event ontology in a low resource setup. We use BERT and non-BERT-based deep learning models in both monolingual and cross-lingual event-argument linking tasks. We also perform an ablation study of various features like position embeddings (PE), position indicator (PI), and segment ID (SI) to understand their contribution to performance improvement in non-BERT-based models. Using three different languages viz. Hindi, Bengali, and Marathi, we compare the results with multilingual BERT-based deep neural models in both monolingual and cross-lingual scenarios. We observe that the multilingual BERT-based model outperforms the best performing non-BERT-based model in cross-lingual settings. But in monolingual settings, the performance is similar in Hindi and Bengali datasets and slightly better in Marathi dataset. We choose the disaster domain due to its social implications. Our current experiments can be helpful in mining important information related to disaster events from news articles and building event knowledge graphs in low-resource languages.

INDEX TERMS Argument trigger, event trigger, event-argument linking, event extraction.
network models and proposes an effective masked language model-based deep learning approach for event-argument linking. In general, if any sentence or two consecutive sentences consist of a single event trigger and multiple argument triggers, then we can assume that all the arguments are linked to that particular event. However, if a particular sentence or two consecutive sentences consist of multiple event and argument triggers, it is difficult to determine which arguments are linked to which events. Suppose a sentence, for example, consists of two event and four argument triggers. In that case, instances are generated out of that sentence considering each of the eight event-argument pairs at a time. Depending on the semantic meaning of the sentence, instances will be labeled positive if the mentioned pair of event and argument trigger possesses any relationship between them. Otherwise, instances will be labeled negative. Thus, event-argument linking is a binary classification problem i.e. there are two possibilities for output: ‘Yes’ and ‘No’. We formulate the event-argument linking task as a relation classification task. The goal of the task is to find out whether there is any relationship between any pair of event and argument triggers or not. The relation classification task has been studied extensively in the literature using features like position embeddings and position indicator. Our current study uses those features in various deep learning models to understand their effect on performance improvement. Also, we observe that some of the argument triggers of an event trigger are not present in the same sentence rather at different sentences. We consider two consecutive sentences while generating instances for classification. We also use segment IDs using 0s and 1s to distinguish between two sentences.

Our current task becomes more challenging as we are under the situation where our target languages are less-resourced, i.e. annotated data and tools are not readily available in the required measure. Although Indian languages have a vibrant cultural and literary history, technological developments in web technologies have started very late. However, of late, the scenario is changing drastically as an increasing number of users are using the Internet in their own languages. However, most of the Indian languages, including Hindi, Bengali, or Marathi, still fall under “low resource”. In recent times, there has been considerable interest in investigating effective deep learning models that can operate under the low-resource scenario. Research and development in Indian languages, especially in knowledge graph research, are still in their infancy. In our current study, we try to address this problem. Also, cross-lingual and multilingual embeddings have been offering effective ways to train deep learning models in cross-lingual and multilingual embedding scenarios by considering the representation in a shared space. More recently, multilingual-BERT (m-BERT) is being used to produce better results in various NLP tasks in low-resource scenarios. Being a highly populous but multilingual country, no single language in India has a dominant user base, though together, it is one of the most extensive user bases. This is the perfect situation where knowledge transfer from one language to another is more effective. In our current proposed work, we also investigate the effectiveness of cross-lingual embeddings in transferring knowledge from one language to another language and compare the results with m-BERT based models. Since the target languages in our experiments have grammatical and linguistic typological (Subject-Object-Verb) similarities, it is assumed that transfer of knowledge between them will be effective.

For experiments, we choose disaster as a domain for its impact on society. To alert both the public and the government, extracting relevant information at an appropriate time is crucial. Equipped with such information, disaster management can be performed. However, it is impossible to mine information manually from the web due to its enormous size. Moreover, if all the information can be stored in a knowledge graph, then that information can be used further for various applications. Our current research is to build a multilingual event knowledge graph under a low resource setup.

II. RELATED WORK

Our current task is to determine if there is any relation between the event and the argument triggers. Thus the task is related to relation extraction, where the relations between a pair of entities are extracted. Currently, deep neural networks are being used for relation extraction. Convolutional Neural Network (CNN) [1] is a very useful feature extractor and has been used widely for various text classification tasks in the past. Zeng et al. [2] proposed to use CNN in relation extraction for the first time, where CNN was used for lexical and sentence level feature extraction. In this paper, the authors also proposed a novel position embeddings (PE) feature, which was very helpful to achieve high accuracy in classification by using two-fold benefits. Firstly, it specifies the pair of words or phrases to which the predicted relation label will be assigned. Secondly, it encodes the relative distance feature to the target words or phrases. Benefiting from these two-fold help, the proposed CNN-based approach improves the classification accuracy. Similarly, to minimize the dependence of external toolkit and resources, the authors in [3] also proposed to use CNN for relation extraction. They used multiple window-sized filters in their CNN architecture to capture wider ranges of n-grams. Santos et al. [4] proposed a CNN-based relation classifier that performs classification using a Ranking CNN (CR-CNN). The proposed model learns a vector representation for each class. The CNN representation of input text is compared to the class representation to generate a score for each class. The authors introduced a pairwise ranking loss which helps in reducing the impact of artificial class. Xu et al. [5] proposed to extract features from the shortest dependency path (SDP) using CNN to avoid irrelevant words in long-distance relationships. Shen and Huang [6] proposed to use POS embeddings along with word and position embeddings to improve the performance. A multi-level attention CNN was proposed in [7] to detect more subtle cues for relation classification in heterogeneous context. The multi-label attention captures...
both entity-specific and relation-specific pooling attentions to help to increase the accuracy of the architecture. They also propose a pairwise margin-based loss function and claim that the proposed loss function is superior to the standard loss function. Though CNN is a suitable feature extractor, it fails to extract syntax and hierarchical information of sentences. Based on this observation, Li et al. [8] introduced hierarchical layers and dependency embedding to CNN architecture. The authors claim that their proposed dependency tag and direction embeddings help CNN in capturing the dependency structure with the window size. More recently, in [9], the authors proposed a CNN-based method with Adversarial training to enhance the robustness of the classifier. They also used intra and cross-sentence attention to select the most crucial information. Alongside the techniques mentioned above in architecture and training labels, the authors also used the description of the marked entities to enhance the performance of the proposed model. Though CNN-based models are helpful in automatically learning features, they fail to capture the long-distance relationship in text. To overcome these difficulties, the Recurrent Neural Network (RNN) based approach was proposed in [11] to capture the long-distance relationship. The authors also used position indicators (PI) to indicate the target nominal in the input text. Further, they also showed that position indicators are better than previously proposed position embeddings. An SDP-based Long Short Term Memory network (SDP-LSTM) was proposed in [12] to pick helpful information along with SDP for different information channels. They also used other information viz. POS tags, grammatical relations, and wordnet hypernyms along with SDP in their proposed model. Liu et al. [13] proposed an Augmented Dependency Path (ADP), which consists of an augmented subtree to the SDP in relation classification. Peng et al. [14] proposed to use a document graph to capture in and across sentences. Then, they used graph-LSTM to encode the input text. Zhang et al. [15] used the attention layer to capture word-level context information and tensor layer to capture the complex connection between two entities, respectively, on the top of Bi-LSTM. Miwa and Bansal [16] presented a novel end-to-end neural model to extract entities and relations between them. Their proposed model allows joint modeling of both entities and relations using both bidirectional sequential and bidirectional tree-structured LSTM-RNNs.

Some researchers used both RNN and CNN to capture local features as well as long-term relationships [17], [18], [19]. Vu et al., for example, explored both CNN and RNN and their combined representation for relation classification. They presented an extended middle context for the CNN, which uses all parts of the input sentence, including the relation arguments, left/right, and between them, focusing on the middle part. They used a connectionist bidirectional RNN model and ranking loss for the RNN model. Finally, they used a simple voting mechanism to integrate CNN with RNN. Cai et al. [18] proposed a bidirectional neural network (BRCNN) which consists of two RCNNs to learn features along SDP in both directions simultaneously. A two-channel LSTM unit was used to extract information of words and dependency relations. A convolution layer extracts the features of dependence units in an SDP. Apart from CNN and RNN, some other approaches were reported in the literature. He et al. [20] proposed syntax-aware entity embeddings based on tree-GRU for neural relation classification. Reinforcement learning was used in [21] in relation classification from noisy data. Inspired by Generative Adversarial Networks (GANs), Zeng et al. [22] proposed a GAN-based method for Distant Supervised Relation Extraction. Reinforcement learning was also used by Zeng et al. [23] to learn sentence relation extractors with the distantly supervised dataset. The research community has also explored relation extraction using deep learning techniques in the cross-lingual setup. A pipeline in a relation extraction system was developed in [24] for any source language. Lin et al. [25] proposed a multilingual attention-based neural relation extraction (MNRE) model. This model employs mono-lingual attention to select the informative sentences within each language. The proposed model employs cross-lingual attention to take advantage of pattern consistency and complementarity among languages. After the recent success of Transformer, architecture [26], pre-trained models like BERT [27] are being used by the research community for relation extraction. In [28], the authors used pre-trained BERT models for relation classification. They also leveraged the entity information in their proposed model to improve performance. In a recent study, Liu et al. further extend this architecture in [29]. To capture the latent information around the target entities, the authors utilize piecewise convolution [30]. They also employ the focal loss function [31] to solve the problem of class imbalance. In another work, [32], a task agnostic representation from entity linked text was proposed. The authors’ main goal is to learn a mapping from relation statement to relation representation. While most of the previous work is based on English, very little work is done in Indian languages. Recently a benchmark corpus is proposed in [33] which consists of news data from disaster domains.

### III. TASK DESCRIPTION AND CONTRIBUTIONS

Event-Argument linking can be defined as a task to find out a link between event and argument triggers which are marked in a given sentence. We formulate the task as a classification problem where we classify a sentence, marked with event and argument triggers, into two labels namely ‘1’ and ‘0’. If event and argument triggers are linked, then predicted label will be ‘1’, or ‘0’ otherwise. In the given example, there are two event triggers, namely “**भूकंप** (earthquake) and ‘**तsunami**” (tsunami) and three argument triggers which are ‘**7.2**’, ‘**इंडोनेशिया**’ (Indonesia) and ‘**71 लोगो**’ (71 people).

Here the place argument ‘**इंडोनेशिया**’ (Indonesia) is linked to both ‘**भूकंप** (earthquake) and ‘**tsunami**’ (tsunami) but ‘7.2’ is linked to ‘**भूकंप** (earthquake) only and ‘**71 लोगो**’ (71 people) is linked to ‘**tsunami**’ (tsunami). So, for all
the above case the classification label will be ‘1’. On the other hand ‘7.2’ is not linked to ‘तुनामी’ (tsunami) and ‘71 लोगों’ (71 people) is not linked to ‘भूकंप’ (earthquake).
In this two case, the classification label will be ‘0’. Thus, for the given example, total six training instances will be created out of which four have label ‘1’ and two have label ‘0’.

Example:
- Input Sentence: 7.2 की तीव्रता के भूकंप के बाद हुई तुनामी में इंडोनेशिया में 71 लोगों की मौत।
- Transliteration: 7.2 key tivrata ke bhukamp ke baad hui tsunami mein Indonesia mein 71 logon ki maut.
- Translation: 71 people died in Indonesia in a tsunami after a magnitude 7.2 earthquake.

In this research paper, we have the following contribution.
- We present a comparative study of the performance of various deep learning models for event-argument linking for three Indian languages, viz. Hindi, Bengali and Marathi.
- We also perform cross-lingual transfer of knowledge from Hindi to Bengali, Bengali to Hindi, Hindi to Marathi and Marathi to Hindi data for our current task using top performing feature set, namely position embeddings (PE), position indicator (PI) and segment ID (SI).
- We propose multilingual-BERT based models for event-argument linking task for both monolingual as well as cross-lingual settings and compare the result with previous deep learning based models.

We describe features and deep learning architectures in next section.

IV. METHODOLOGIES
This section will discuss various deep learning architectures we have used for building our models. In details, we will also discuss all the three features that we have used in our current experiment, namely position indicator (PI), position embeddings (PE), and segment ID (SI). We consider six non-BERT-based deep learning architectures and two m-BERT based architecture for the experiments. Six non-BERT-based architectures are CNN, PCNN, RCNN, Bi-LSTM-CNN, Bi-LSTM-Attention, Bi-LSTM-Self-Attention.

A. BASELINE MODELS

1) CNN
CNN (Convolutional Neural Network) is a type of artificial neural network which can automatically capture features from vector representation of input sentences. It performs convolution on input representation using a set of filters of different sizes. Finally, a max-pooled operation [34] extracts maximum values from each of the filters. Thus, CNN can extract the most informative n-gram features from the input representations. Owing to this capability, CNN is successfully applied on various classification tasks such as sentence classification [1], relation classification [2], [3], [35] etc. In our current experiment, we use three filters of kernel size 3, 4 and 5, respectively (Fig. 1). We also apply a max-pooling layer on each filter.

2) PCNN
The main drawback of using CNN is that it uses a single max-pool operation on each filter to capture the essential features. This strategy works well for the sentence classification task. However, a single max-pool operation is not sufficient for the relation classification task, where it is equally important to model both entities’ structural information. To address this problem, Zeng et al. [30] proposed a novel Piecewise Convolutional Neural Network (PCNN). To capture the structural as well as other latent information between and around two entities, the authors divided the convolution results from each filter into three segments based on the given entity positions and performed a piece-wise max-pooling operation instead of the single max-pooling operation. The proposed max-pooling operation successfully captures each part’s maximum value, resulting in superior performance than normal CNN. Similar to our CNN model, we use three filters of kernel size 3,4 and 5 respectively for the PCNN model and then perform a piece-wise max-pooling operation on each filter (Fig. 1(b)).

3) RCNN
Recurrent Neural Network (RNN) is good at capturing the contextual information in long texts. However, the RNN is a biased model, where later words have more influence than earlier words. CNN is introduced to cope with this bias problem, which can identify the discriminative phrases in an input sentence with its max-pooling layer. But, CNN has fixed window size problem. It is tough to determine the optimal window size to choose a trade-off between loss of crucial information and large parameter space. To address the issues of both the models, [36] proposed a Recurrent Convolutional Neural Network (RCNN) for text classification (Fig. 1(c)). The proposed model comprises of a bi-directional recurrent structure followed by a max-pooling layer. The recurrent structure captures crucial information from a larger context window than traditional CNN. Then the max-pooling layer automatically judges the key features for the classification task. Thus, the proposed architecture utilizes the combined advantages of both RNN and CNN. We use Bi-directional Long Short-Term Memory (Bi-LSTM) as the recurrent unit in our current implementation, followed by a max-pooling layer. LSTM is a variant of RNN with a ‘memory cell’ that stores information in memory for a longer period. It also has three gates that control the update, deletion, and output of information. Better control over the gradient flow help to solve the vanishing and exploding gradient problem.

4) BI-LSTM-CNN
We also experiment with Bi-LSTM-CNN architecture which was proposed by [37] for text classification (Fig. 1(d)). Unlike RCNN, which has a Bi-LSTM layer followed by a max-pooling layer, Bi-LSTM-CNN has a Bi-LSTM layer followed
by a CNN layer. The CNN layer performs convolution on the hidden representation of Bi-LSTM using a set of filters of sizes 3, 4, and 5, respectively. Finally, a max-pooled operation extracts maximum values from each of the filters. Thus, the hybrid model can capture long distance information and local information.
5) BI-LSTM-ATTENTION
The attention mechanism was introduced on the top of Recurrent Neural Network in seq2seq model to solve the Neural Machine Translation (NMT) problem by [38] to automatically find the important parts of the source sentence which are relevant to predict a target word. Though the decoder part of the encoder-decoder architecture is not required in a classification problem, the attention mechanism is useful to find the influential words in the input sentence to predict the correct class. In our current experiments, we use the attention mechanism on top of Bidirectional LSTM based sentence encoder (Fig. 1(e)).

6) BI-LSTM-SELF-ATTENTION
After the successful use of self-attention in Transformer [26], which has achieved state-of-the-art performance on multiple machine translation datasets without having recurrence or convolution components, it has been widely used in solving various NLP tasks. In our problem, information related to how each word relates with all the other words is necessary. This is because repetition of an identical event and/or argument triggers in a sentence can occur. It is not necessary to always have similar events linked to identical arguments. In such scenarios, information about how each word is related to others is of utmost significance. We apply a self-attention mechanism on top of the Bi-LSTM layer (Fig. 1(f)).

7) MULTILINGUAL-R-BERT
These days, masked language model (MLM) based BERT (Bidirectional Encoder Representations from Transformers) [27] achieves very successful results in various NLP tasks ranging from classification as well as sequence labeling problems. However, unlike other classification problems, the relation extraction task relies on the information of both the sentence and two marked entities. As in our case, we formulate the event-argument linking task as a relation extraction problem; we use the BERT-based relation extraction architecture proposed by [28] with slight modification (Fig. 2). We are working in multiple languages. So we use multilingual BERT instead of vanilla BERT for our experiments. To use information about both the marked event and argument triggers, we first perform an average operation on both the event and argument triggers marked in our input representation to obtain the vector representations of both of them. Then, we apply fully connected layers to each of the two vectors followed by the $\tanh$ activation function. Then we concatenate both the hidden representations with the pooled output (CLS token) to obtain a final representation. Then, we use a linear layer followed by a $\text{softmax}$ function for further classification as we perform in all the previous models.

8) MULTILINGUAL-R-BERT + PCNN
We also use another BERT-based architecture proposed in [29] with minor changes. This architecture is an extension of the previous BERT-based architecture proposed by [28]. On top of the BERT representation, an additional Piecewise Convolutional Neural Network (PCNN) [30] was employed to capture the latent information between and around the event and argument trigger in an input sentence. In addition, instead of the vanilla BERT proposed in the original paper, we employ multilingual BERT. We keep the focal loss function [31] proposed by the authors of the main research article in [29], even if the class imbalance problem is not applicable in our instance.

B. INPUT REPRESENTATION
Fig. 3 depicts the representation of input instance "इंडोनेशिया में 7.2 की तीव्रता का भूकंप आया इसमें"
To incorporate the position embedding feature for each word, we first calculate each word’s relative distance with respect to event and argument triggers, respectively. This relative distance can be both positive and negative based on the position of each word with respect to the trigger words. A random vector of dimension 50 is chosen to convert each distance to a vector. Fig. 3 shows how the relative distance is calculated for each word with respect to event and argument triggers of the input instance. Table 6, Table 7 and Table 8 show the effect of PE.

3) POSITION INDICATORS
We also use position indicator (PI) [11] as a feature that indicates the respective triggers. Position indicators are a simple yet effective way to indicate the target triggers in an input instance. Position indicators specify the start and end of the event and argument triggers. These four indicators are regarded as words in input instances.

4) SEGMENT ID
Segment ID (SI) help the model to identify whether the event triggers and the argument triggers belong to the same sentence or not. For the first sentence, the segment ID value of each word is 0s, and for the second sentence, the same is 1s. This feature is important as most of the cases, event, and argument triggers that belong to different sentences are not linked to each other (‘Yes’ class). In contrast, event and argument triggers that belong to the same sentence are linked (‘No’ class).

V. DATASET AND EXPERIMENTAL SETUP
This section describes the datasets, hyper-parameter settings, comparison to the baseline architectures, and ablation study to decide the contribution of the different aspects of our system.
A. DATASET
For Hindi event-argument linking dataset, we use the dataset proposed in [33] which is an event extraction dataset consisting of disaster news articles in Hindi language. Since there was no existing dataset for event-argument linking in Bengali and Marathi, similar datasets are annotated internally following the same approach introduced by [33] with slight modification. Hindi dataset contains 28 event types as well as 11 argument types. Bengali dataset consists of 33 event types as well as 14 argument types. Additional event types than Hindi dataset are Drought, Epidemic, Pandemic, Heavy_Rainfall, Seismic_Risk, Suicide_Attack whereas event
type Rock_Fall is not present in Bengali dataset. Additional three new argument types in the Bengali dataset are Depth, Temperature, and Type. Marathi dataset consists of 30 event types having three additional event classes, namely Drought, Seismic_Risk, and Suicide_Attack and one absent event class, namely Hurricane as compared to Hindi dataset. Similar to the Bengali dataset, the Marathi dataset consists of three additional argument types, namely Depth, Temperature, and Type as compared to the Hindi dataset. The news data related to disaster events are crawled from different online news portals. The downloaded raw files are pre-processed and converted into XML files. Similar to the Hindi dataset in [33], the annotation includes highlighting the event and argument triggers, mentioning the type of such triggers, and incorporating the linking information present in the text. Each marked trigger and link are assigned an unique ID for a particular document. These unique IDs help to identify the triggers and links while preparing the input-output data for the experiments. For annotation, three annotators for each language are employed. All the annotators have proficiency in their respective languages. The dataset statistics is described in Table 1.

**B. EXPERIMENT DETAILS**

In this section, we describe the experimental details. We perform both monolingual and cross-lingual experiments using non-BERT-based deep learning models and compare their performance with the multilingual BERT-based models in all three languages. We also perform the ablation study of various feature combinations of the non-BERT-based deep learning models for monolingual experiments. We consider four non-BERT-based deep learning models for cross-lingual experiments, namely RCNN, Bi-LSTM-CNN, Bi-LSTM-Self-Attention and PCNN with the full feature set. We use both monolingual and cross-lingual word embeddings to assess the usefulness of cross-lingual embeddings over monolingual embeddings. In monolingual embedding setup, we train the non-BERT-based models using the monolingual word embeddings of source language and test using the monolingual word embeddings of target language. For example, in case of Hindi-Bengali cross-lingual experiment, we train the Hindi data using monolingual word embeddings of Hindi fastText embeddings. While testing, we use monolingual word embeddings of Bengali fastText embeddings to represent Bengali test data. In a cross-lingual embedding setup, we align all three fastText embeddings of three languages, namely Hindi, Marathi, and Bengali, into English word embeddings vector space. Finally, we compare the performance of non-BERT based models with BERT based models.

**C. HYPER-PARAMETER SETTINGS**

We perform the experiments using the hyper-parameter settings described in Table 3. We optimize parameters using the Adam optimization algorithm with default settings. We also use gradient clipping to deal with exploding gradient. For Non-BERT based model, we clip the derivatives of the loss function using the clip value 0.1. For BERT based models, we use gradient value with norm value 1.0. We use cross-entropy loss as our loss function.

**VI. RESULT AND ANALYSIS**

In this section, we describe the obtained results from our experiments. We first perform the quantitative and qualitative analysis of the monolingual experiments followed by cross-lingual experiments.

1) MONO-LINGUAL EVENT-ARGUMENT LINKING

The primary observation from the Table 6, Table 7 and Table 8 is that, for Hindi and Bengali monolingual experiments, RCNN and Bi-LSTM-CNN, with all features, namely position embedding (PE), position indicator (PI), and segment ID (SI), produce similar or better performance compared to BERT based model (refer Table 6 and Table 8). However, the BERT-based Marathi model performs better than all non-BERT-based deep learning models in the monolingual experiments (refer Table 7). We observe that the combined set of all features helps RCNN and Bi-LSTM-CNN, which take into account long-distance contextual information from both directions, perform at per with the BERT-based model. The possible reason behind such performance could be that, apart from semantic information embedded in the vector representation, both the above models are using contextualized information similar to BERT. Moreover, the above models use similar features like segment ID and position information which are also present in the proposed multilingual BERT models. We have also carried out an ablation study of various features in non-BERT-based deep learning models to understand their usefulness.

We also observe that the performance of the Marathi language is best among all languages, and the performance of the Hindi language is worst. This phenomenon can be understood from Fig. 5, where it shows that the average length of sentence and the average length between event and argument triggers are highest in the case of Hindi and lowest in the case of Marathi. Also, the average event and argument trigger lengths

| Hyperparameter | Final value |
|----------------|-------------|
| Batch          | 32          |
| Learning Rate  | 2e-5        |
| Dropout        | 0.5         |
| \(d_w\)        | 300         |
| \(d_p\)        | 50          |
| \(d_s\)        | 50          |
| Epoch          | 20          |

**TABLE 3. Hyperparameter settings. \(d_w\) stands for dimension of word embeddings, \(d_p\) stands for dimension of position embeddings and \(d_s\) stands for dimension of segment embeddings.**
**TABLE 4.** Average monolingual model performance w.r.t length between event and argument triggers.

| Language | Experiments | Prediction | Average Length between Event-Argument - ‘Yes’ Cases | Average Length between Event-Argument - ‘No’ Cases |
|----------|-------------|------------|---------------------------------------------------|--------------------------------------------------|
| Hindi    | Average of All Models | Correct | 8.20 | 18.89 |
|          |             | Incorrect | 16.41 | 12.07 |
| Marathi  | Average of All Models | Correct | 5.33 | 12.75 |
|          |             | Incorrect | 10.94 | 8.26 |
| Bengali  | Average of All Models | Correct | 5.55 | 13.89 |
|          |             | Incorrect | 12.47 | 9.18 |

**TABLE 5.** Average monolingual model performance w.r.t intra and inter sentence event argument triggers. When both event and argument triggers belong to same sentence, then it is called intra-sentence event-argument triggers. When event and argument triggers belong to different sentences, then it is called inter-sentence event-argument triggers.

| Language | Experiments | Prediction | Intra Sentence ‘Yes’ cases (%) | Inter Sentence ‘Yes’ cases (%) | Intra Sentence ‘No’ cases (%) | Inter Sentence ‘No’ cases (%) |
|----------|-------------|------------|-------------------------------|-----------------------------|------------------------------|-------------------------------|
| Hindi    | Average of All Models | Correct | 84.25 | 15.75 | 5.20 | 94.80 |
|          |             | Incorrect | 19.01 | 80.99 | 37.32 | 42.68 |
| Marathi  | Average of All Models | Correct | 91.78 | 8.72 | 8.23 | 91.78 |
|          |             | Incorrect | 27.73 | 72.27 | 61.16 | 38.84 |
| Bengali  | Average of All Models | Correct | 84.29 | 15.71 | 15.57 | 84.43 |
|          |             | Incorrect | 21.16 | 78.84 | 54.13 | 45.87 |

**TABLE 6.** Results for Hindi monolingual event-argument linking. Only F-score is reported.

| Features | Class Labels | RCNN + Bi-LSTM | Bi-LSTM + CNN | Bi-LSTM + Self-Attn | Bi-LSTM + Attn | PCNN | CNN | M-BERT | M-BERT + PCNN |
|----------|--------------|----------------|--------------|---------------------|----------------|------|-----|--------|---------------|
| All Feature | 0 | 0.86 | 0.85 | 0.84 | 0.84 | 0.82 | 0.78 | 0.86 | 0.86 |
| 1 | 0.89 | 0.89 | 0.88 | 0.88 | 0.86 | 0.85 | 0.89 | 0.89 |
| PE+SI | 0 | 0.84 | 0.85 | 0.81 | 0.85 | 0.83 | 0.80 | 0.89 | 0.89 |
| 1 | 0.88 | 0.88 | 0.88 | 0.88 | 0.87 | 0.85 | 0.85 | 0.85 |
| PE+PI | 0 | 0.85 | 0.84 | 0.83 | 0.84 | 0.80 | 0.77 | 0.84 | 0.84 |
| 1 | 0.88 | 0.88 | 0.88 | 0.87 | 0.86 | 0.84 | 0.85 | 0.85 |
| PE | 0 | 0.84 | 0.82 | 0.83 | 0.85 | 0.82 | 0.80 | 0.89 | 0.89 |
| 1 | 0.88 | 0.88 | 0.88 | 0.87 | 0.86 | 0.85 | 0.85 | 0.85 |
| PI | 0 | 0.84 | 0.83 | 0.83 | 0.83 | 0.79 | 0.70 | 0.86 | 0.86 |
| 1 | 0.88 | 0.88 | 0.87 | 0.87 | 0.84 | 0.79 | 0.79 | 0.79 |
| No Feature | 0 | 0.42 | 0.42 | 0.37 | 0.49 | 0.78 | 0.34 | 0.72 | 0.72 |
| 1 | 0.72 | 0.72 | 0.71 | 0.72 | 0.84 | 0.72 | 0.72 | 0.72 |

are highest for Hindi and lowest for Marathi (refer Fig. 4). Longer sentence and trigger length and the longer length between event-argument cause more difficulty in learning long-distance contextual information in the Hindi dataset compared to the other two languages, namely Marathi and Bengali.

Detailed analysis of dataset reveals the usual fact that, in most of ‘Yes’ cases, event and argument trigger are in same sentences (intra-sentence) (refer Table 2) and the distance between them is lesser (refer Fig. 5), whereas for ‘No’ cases, event and argument trigger are in different sentences (inter-sentence) and the distance between them is also higher. Thus the position of triggers and distance between them are crucial. Table 4 shows that, across all languages and all models, the average length between event-argument triggers in correct ‘Yes’ cases are much smaller than the average length between event-argument triggers in correct ‘No’ cases. In contrast, for incorrect cases, the opposite situation is observed. Thus, inter-sentence ‘Yes’ cases and intra-sentence ‘No’ cases are more challenging than usual. Table 5 evidences this observation. It shows that most correct cases belong to intra-sentence ‘Yes’ cases and inter-sentence ‘No’ cases. For example, in the Hindi language, 84.25% correct predictions are intra-sentence ‘Yes’ cases, whereas 15.75% predictions are inter-sentence ‘Yes’ cases. This proportion is different than their respective proportion of 77.76% and 22.24% in Hindi test set (refer Table 2). Table 9 shows the performance of various models in inter-sentence ‘Yes’ and intra-sentence
TABLE 7. Results for Marathi monolingual event-argument linking. Only F-score is reported.

| Features | Class Labels | Models | 
|----------|--------------|--------|
|          | RCNN | Bi-LSTM + CNN | Bi-LSTM + Attn | Bi-LSTM + Self-Attn | PCNN | CNN | M-BERT | M-BERT + PCNN |
| All Feature | 0 | 0.92 | 0.92 | 0.92 | 0.92 | 0.89 | 0.84 | 0.94 | 0.94 |
|           | 1 | 0.93 | 0.93 | 0.93 | 0.93 | 0.90 | 0.87 | 0.95 | 0.95 |
| PE+SI     | 0 | 0.92 | 0.92 | 0.90 | 0.89 | 0.89 | 0.87 |      |      |
|           | 1 | 0.93 | 0.93 | 0.91 | 0.91 | 0.90 | 0.89 | 0.90 | 0.89 |
| PE+PI     | 0 | 0.90 | 0.91 | 0.90 | 0.90 | 0.88 | 0.84 |      |      |
|           | 1 | 0.92 | 0.91 | 0.91 | 0.91 | 0.91 | 0.88 | 0.91 | 0.88 |
| PE        | 0 | 0.91 | 0.91 | 0.89 | 0.90 | 0.89 | 0.88 |      |      |
|           | 1 | 0.92 | 0.92 | 0.92 | 0.92 | 0.90 | 0.87 | 0.94 | 0.87 |
| PI        | 0 | 0.91 | 0.91 | 0.90 | 0.90 | 0.88 | 0.85 |      |      |
|           | 1 | 0.92 | 0.92 | 0.91 | 0.91 | 0.88 | 0.84 | 0.91 | 0.84 |
| No Feature| 0 | 0.51 | 0.48 | 0.38 | 0.47 | 0.86 | 0.34 |      |      |
|           | 1 | 0.68 | 0.69 | 0.67 | 0.70 | 0.88 | 0.69 |      |      |

TABLE 8. Results for Bengali monolingual event-argument linking. Only F-score is reported.

| Features | Class Labels | Models | 
|----------|--------------|--------|
|          | RCNN | Bi-LSTM + CNN | Bi-LSTM + Attn | Bi-LSTM + Self-Attn | PCNN | CNN | M-BERT | M-BERT + PCNN |
| All Feature | 0 | 0.92 | 0.92 | 0.92 | 0.92 | 0.90 | 0.89 | 0.93 | 0.93 |
|           | 1 | 0.93 | 0.93 | 0.93 | 0.93 | 0.92 | 0.89 | 0.93 | 0.93 |
| PE+SI     | 0 | 0.91 | 0.91 | 0.90 | 0.91 | 0.89 | 0.87 |      |      |
|           | 1 | 0.92 | 0.92 | 0.92 | 0.92 | 0.90 | 0.89 | 0.93 | 0.93 |
| PE+PI     | 0 | 0.90 | 0.90 | 0.89 | 0.89 | 0.89 | 0.87 |      |      |
|           | 1 | 0.91 | 0.91 | 0.91 | 0.91 | 0.91 | 0.89 | 0.93 | 0.89 |
| PE        | 0 | 0.89 | 0.89 | 0.88 | 0.89 | 0.89 | 0.88 |      |      |
|           | 1 | 0.91 | 0.91 | 0.91 | 0.91 | 0.91 | 0.90 | 0.94 | 0.90 |
| PI        | 0 | 0.90 | 0.90 | 0.88 | 0.90 | 0.88 | 0.88 |      |      |
|           | 1 | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 | 0.88 | 0.94 | 0.88 |
| No Feature| 0 | 0.43 | 0.50 | 0.40 | 0.41 | 0.84 | 0.42 |      |      |
|           | 1 | 0.71 | 0.69 | 0.71 | 0.72 | 0.87 | 0.70 |      |      |

TABLE 9. Accuracy of various monolingual models w.r.t intra and inter sentence event argument triggers. When both event and argument triggers belong to same sentence, then it is called intra-sentence event-argument triggers. When event and argument triggers belong to different sentences, then it is called inter-sentence event-argument triggers.

| Trigger Position | Models | RCNN | Bi-LSTM + CNN | Bi-LSTM + Attn | Bi-LSTM + Self-Attn | PCNN | CNN | M-BERT | M-BERT + PCNN |
|------------------|--------|------|---------------|---------------|--------------------|------|-----|--------|----------------|
| Hindi | | | | | | | | | |
| Intra-sentence ‘Yes’ case (%) | 98 | 99 | 98 | 97 | 99 | 97 | 97 | 97 | 96 |
| Inter-sentence ‘Yes’ case (%) | 64 | 71 | 60 | 66 | 56 | 66 | 68 | 67 | 67 |
| Intra-sentence ‘No’ case (%) | 35 | 28 | 27 | 33 | 13 | 14 | 41 | 43 | 43 |
| Inter-sentence ‘No’ case (%) | 93 | 89 | 92 | 90 | 91 | 83 | 92 | 93 | 93 |
| Marathi | | | | | | | | | |
| Intra-sentence ‘Yes’ case (%) | 99 | 99 | 98 | 98 | 98 | 98 | 98 | 98 | 98 |
| Inter-sentence ‘Yes’ case (%) | 68 | 75 | 64 | 67 | 45 | 56 | 77 | 74 | 74 |
| Intra-sentence ‘No’ case (%) | 54 | 55 | 52 | 60 | 30 | 30 | 70 | 73 | 73 |
| Inter-sentence ‘No’ case (%) | 95 | 94 | 97 | 96 | 97 | 86 | 96 | 97 | 97 |
| Bengali | | | | | | | | | |
| Intra-sentence ‘Yes’ case (%) | 99 | 98 | 99 | 97 | 98 | 97 | 98 | 98 | 98 |
| Inter-sentence ‘Yes’ case (%) | 75 | 78 | 75 | 70 | 60 | 60 | 71 | 74 | 74 |
| Intra-sentence ‘No’ case (%) | 73 | 78 | 63 | 78 | 68 | 59 | 76 | 77 | 77 |
| Inter-sentence ‘No’ case (%) | 95 | 94 | 93 | 95 | 93 | 88 | 96 | 96 | 96 |

‘No’ cases. We observe that the accuracy percentage of intra-sentence ‘No’ cases in Hindi is the lowest among all the cases across all languages. In this scenario, the accuracy of the BERT-based model outperforms the other models with significant merging in the Hindi and Marathi dataset. However, Bi-LSTM-CNN and Bi-LSTM-Self-Attention perform
s slightly better than the BERT-based model for the Bengali dataset.

2) EFFECT OF POSITION FEATURE
From the above discussion, we observe that, position information of triggers and the distance information between them is essential for learning the current tasks. In this regard, each position feature plays an important role. Results in Table 6, Table 7 and 8 depict that without position feature, the deep learning models perform poorly except PCNN, as it slices the input sentence into pieces based on the position of event and argument triggers. For CNN and PCNN, position embedding (PE) presents better performance than position indicator (PI). Moreover, for PCNN and CNN, combining PI with PE brings down the final score (no changes in PCNN-Bengali). Similarly, combining PI with PE+SI combination brings down the final score for PCNN and CNN models across languages (no changes in PCNN-Marathi). However, both the position features show a similar result for models with LSTM as the base encoder. Combining PI with PE mainly improves or does not change the model’s performance except few cases where slight deterioration (RCNN-Marathi and LSTM-self-attention-Hindi) of the result is observed. Example 1, 2, 3 and 4 show that, the prediction output of the RCNN model with and without position feature for all the four cases, namely intra-sentence ‘Yes’, inter-sentence ‘Yes’, inter-sentence ‘No’, and intra-sentence ‘No’ cases, respectively. From the examples, we observe that most of the predictions of ‘No’ cases are incorrect without the position feature. In example 1, ‘भूकंप’ (earthquake) is an event trigger of type Earthquake and “6.9 तीव्रता” (6.9 magnitude) is the argument trigger of type Magnitude. In this case, the magnitude is linked to the earthquake event trigger. Here, both the event and argument triggers belong to the same sentence. In example 2, ‘तूफान’ (storm) is the event trigger and “टेलीफोन नेटवर्क तबाह हो गया” (Telephone networks have been destroyed) is the argument trigger. Here, both the triggers belong to different sentences and are linked to each other. In both cases, the RCNN model with and without position feature correctly predicts the class (‘Yes’). In example 3, however, both the event (“लॉबर डे” (Labor Day)) and argument (“लॉबर डे” (Labor Day)) triggers belong to different sentences and they are not related to each other. But RCNN without position features cannot correctly predict the class (‘No’), though RCNN with position features can correctly predict the class. In example 4, both the event (“तूफान” (tsunami)) and argument triggers (“लोगों में दहशत फैला दी” (caused panic among the people)) belong to same sentence and not linked to each other. In this case, RCNN without position feature also fails to predict the correct class (‘No’), whereas RCNN with position feature successfully predicts the correct class.

- Transliteration: indonesia mein lagatar bhookark or tsunami aati rahi hain. 5 august 2018 ko bhi indonasia mein 6.9 tivrata ka bhookark aayaa tha.
- Translation: There have been frequent earthquakes and tsunamis in Indonesia. On August 5, 2018, there was also a 6.9 magnitude earthquake in Indonesia.
- RCNN + PE + PI Architecture Prediction: ‘Yes’
- RCNN + No Feature Architecture Prediction: ‘Yes’
- Example 2 (Inter-Sentence ‘Yes’ Case): तूफान तबाही का तूफान निरीक्षण के दौरान उचित गए। केवल तत्काली का टेलीफोन नेटवर्क तबाह हो गया है।
- Transliteration: tufaan itana teji tha ki kai ped aur bilji k khambhe ukhad gaye. kai ilakon mein telephone net-work tabah ho gaya hai.
- Translation: The storm was so strong that many trees and electric poles were uprooted. Telephone networks have been destroyed in many areas.
- RCNN + PE + PI Architecture Prediction: ‘Yes’
- RCNN + No Feature Architecture Prediction: ‘Yes’
- Example 3 (Inter-Sentence ‘No’ Case): इससे पहले 1988 में ओलिंपिक गिलर्ड नामक तूफान ने सबसे ज्यादा तबाही मचाई थी। इसे नंबर पर लॉबर डे तूफान ने जो 1935 में आया था।
- Transliteration: isase pahale 1988 mein onalee gilabart naamak toophaan se sabse jyada tabahai mchaai thee. doosare numbar par lebar de toophaan ne jo 1935 mein aayaa tha.
- Translation: Earlier in 1988, a hurricane named Only Gilbert caused the most destruction. At number two was the Labor Day hurricane which came in 1935.
- RCNN + PE + PI Architecture Prediction: ‘No’
- RCNN + No Feature Architecture Prediction: ‘Yes’
- Example 4 (Intra-Sentence ‘No’ Case): टूफान। इंडोनेशिया में आई जबाबदार सुनामी से दुनिया अभी उबरी नहीं थी बकरी कृपण ने लोगों में दहशत फैला दी।
- Transliteration: toophaan. indonezhiya mein aai jababdr sanammi se dunia abhi ubari nahi thi, baki krupnar ne logon mein daahsath fiala di.
- Translation: Tokyo. Indonesia mein i jabaradast tsunami se duniya abhi ubari nahi thii ki japan mein aayee ek aur bhookark ne logon mein daahsat faila de.
- Translation: Tokyo. The world had not yet recovered from the tremendous tsunami in Indonesia that another earthquake in Japan caused panic among the people.
- RCNN + PE + PI Architecture Prediction: ‘No’
- RCNN + No Feature Architecture Prediction: ‘Yes’

3) EFFECT OF SEGMENT ID
Another observation from the detailed analysis of the dataset indicates the importance of segment information. From the experiments also, we observe that, in most of the cases (PE vs. PE+SI and PE+PI vs. All Features cases), the introduction of segment ID improves the performance of the systems. For example, in example 5, ‘भूकंप’ (earthquake) is an event trigger and “इस साल सितंबर के महीने” (September this
for cross-lingual experiments than latter by a significant margin. Table 10 depicts the experimental results of cross-lingual experiments. In previous monolingual experiments, we observe that non-BERT-based models with all three features, namely segment ID (SI), position embeddings (PE), and position indicator (PI), perform on par with BERT-based model and sometimes better than that. For cross-lingual experiments, we only consider four models: RCNN, Bi-LSTM-CNN, PCNN, and Bi-LSTM-Self-Attention, with the full feature set to compare the performance with BERT based models. We also assess the usefulness of cross-lingual embeddings [40] over fastText monolingual embeddings. We observe that for a few cases, cross-lingual embeddings perform better than monolingual embeddings. However, performance improvements of multi-lingual BERT-based models are significantly high compared to non-BERT models in both cross-lingual and monolingual experiments. Further analysis reveals that out of four cases viz. intra-sentence ‘Yes’, inter-sentence ‘Yes’, intra-sentence ‘No’ and inter-sentence ‘No’, first one, i.e., intra-sentence ‘Yes’ cases perform well, and the accuracy is similar to the same case in monolingual experiments. However, intra-sentence ‘No’ cases, which are also challenging in monolingual linking, performs poorly, especially in the case of non-BERT-based deep learning models (refer Table 12). Table 12 further reveals that, for some cases, the accuracy is as low as 0. We also observe performance decline of inter-sentence ‘No’ cases. This observation is in line with previous analysis where we observe that the models easily learn and follow the usual scenario, which says nearer argument triggers have a higher probability of being linked than distant argument triggers. In other words, event and argument triggers in the same sentences have higher chances to link to each other (intra-sentences ‘Yes’), and event and argument triggers in different sentences have higher chances not to be linked to each other (inter-sentences ‘No’). Table 11 shows that whatever performance that we achieve in cross-lingual experiments are due to these two categories, which are easier to learn. Among the other two cases viz. inter-sentence ‘Yes’ and intra-sentence ‘No’, the latter one is tough to predict from the position and distance feature and requires semantic knowledge from the text, which comes from the vector representation of the words in the text itself. In monolingual word embeddings, embeddings of both languages are in different spaces. However, cross-lingual word embeddings try to solve the problem by aligning the word vectors in common space (English in our case). But in our case, we observe that the cross-lingual word embeddings do not capture sufficient semantic information to transfer knowledge from one language to another to predict intra-sentence ‘No’ cases. However, we observe the superior performance of multilingual BERT in cross-lingual knowledge transfer. One striking observation from Table 12 is that the performance of the Bi-LSTM-CNN classifier with monolingual embeddings for inter-sentence ‘Yes’ cases is astonishingly higher than the other classifiers. Detailed analysis reveals that the classifier produces a very high recall

year) is the argument trigger. Here, though, both belong to two different sentences, they are very close to each other. In this example, RCNN with position feature cannot predict the correct class (‘No’). Still, RCNN with All feature (which has additional segment ID in addition to RCNN with position model) can predict the correct class successfully. Here, the additional information helps the model in correct prediction. Another example (example 6), in which both the event (‘badh (floods)) and argument (‘Kerala’) belong to the same sentence and they are linked to each other. However, the distance between them is long, leading to the wrong classification. But, segment information helps them predict the correct class (‘Yes’) successfully. There are only a few cases (5 out of 36 cases across all languages) where the performance does not improve and only two (out of 36 cases, namely LSTM-Attention-Hindi and CNN-Marathi) cases where the performance decreases.

- Example 5 (Inter-Sentence ‘No’ Case): रिछले महाने टोफों से करीब 350 किलोमीटर दूर मुख्य होश द्वार पे पुरवतल तपर पर 6.2 तीव्रता का एक शक्तिशाली भूकंप आया था। इस साल सितंबर में महाने में जापान के उत्तरी द्वीप होक्से में आया एक शक्तिशाली भूकंप में 37 लोग मार गए थे।
- Transliteration: pichhale mahine tokyo se karib 350 kilometres door mukhya honshu dweep k purvottar tatar par 6.2 tivrata ka ek shaktishali bhookamp aaya tha. eis saal sitamber ke mahine mein aaye ek shaktishali bhookamp mein 37 log mare gaye the.
- Translation: Last month, a powerful 6.2-magnitude earthquake struck off the northeast coast of the main Honshu island, about 350 kilometers from Tokyo. In September this year, 37 people were killed in a powerful earthquake in Japan’s northern island of Hokkaido.

- RCNN + All Feature Architecture Prediction: ‘No’
- RCNN + PE + PI Architecture Prediction: ‘Yes’

- Example 6 (Intra-Sentence ‘Yes’ Case): अंकने केरल में 14 लाख 52 हजार से ज्यादा लोग राहत शिविरों में रह रहे हैं जबकि बाद ने खेतों में खड़ 43727 हेक्टेर फसल नट कर दी है।
- Transliteration: akele carol mein 14 lakh 52 hazar se jyada log rahat shiviron men rah hain jabki bada ne kheton men khadi 43727 hectare fasal nasht kar di hai.
- Translation: In Kerala alone, more than 14 lakh 52 thousand people are living in relief camps while the floods have destroyed 43,727 hectares of crops standing in the fields.

- RCNN + All Feature Architecture Prediction: ‘Yes’
- RCNN + PE + PI Architecture Prediction: ‘No’

4) CROSS-LINGUAL EVENT-ARGUMENT LINKING

Though the BERT-based models fail to achieve significant improvement over non-BERT-based models like RCNN or Bi-LSTM-CNN, it exhibits superior performance...
value for the ‘Yes’ class, which comes with the expense of a low recall value for the ‘No’ class. We also perform some qualitative analysis of the results. From the result table, we observe that intra-sentence ‘No’ cases are most difficult to learn, and the performance of this scenario is worst among all the scenarios. In example 7, the event (‘hailstorm’) and argument (‘2000’) triggers belong to the same sentence but the argument is not linked to the marked event trigger but to other event trigger (‘storm’) which is not our consideration in this case. Similarly in example 8, the argument trigger (“a huge tree of pakad collapsed on the banks of the canal and fell on a tea shop there”) linked to event trigger “opened fire” (thunderstorm) rather than “hailstorm”). In another example (example 9) from the Bengali dataset, we observe that marked event and argument triggers are in the same sentence but not linked to each other. Here, we have two event of firing from two different participants (“open fire” (SF) and “opened fire” (BDR) respectively) and marked event trigger “opened fire” is not linked to marked argument trigger “open fire” (BDR) but linked to another argument “opened fire” (SF), which is not in our consideration for this case. In the final example (example 10), we observe that both event and argument triggers are very close and belong to the same sentence but are not linked to each other. One common aspect of all the examples is that all are intra-sentence ‘No’ cases that are challenging to predict. In all four cases, the multilingual BERT model in monolingual experiments can correctly predict the class, but the prediction is wrong in cross-lingual experiments.

- **Example 7** (Intra-Sentence ‘No’ Case): हैटी में बाढ़ में मरने वालों की संख्या 2000 हुई। गोनाइवेस, हैटी एपी हैटी के उत्तर पश्चिमी क्षेत्र में दो हज़ार पहले उल्टे तूफान के बाद आई मृत्यु के बाद मरने वालों की संख्या 2000 हो गई है।

- **Translation**: The death toll in the floods in Haiti has reached 2,000. Gonaïves, Haiti (AP) The death toll in the severe flooding that followed the storm that erupted in the northwest region of Haiti two weeks ago has risen to nearly 2,000.

- **Example 8** (Inter-Sentence ‘Yes’ Case): Hindi-Marathi.

- **Translation**: mBERT + Monolingual Experiment Prediction: ‘No’

- **Translation**: mBERT + Cross-lingual Experiment Prediction: ‘Yes’
TABLE 12. Accuracy of various cross-lingual models w.r.t intra and inter sentence event argument triggers. When both event and argument triggers belong to same sentence, then it is called intra-sentence event-argument triggers. When event and argument triggers belong to different sentences, then it is called inter-sentence event-argument triggers.

| Trigger Position                  | Models                                      | RCNN + All Features | Bi-LSTM + CNN + All Features | PCNN + All Features | Bi-LSTM + Self-Atten + All Features | M-BERT + PCNN |
|-----------------------------------|---------------------------------------------|---------------------|------------------------------|---------------------|--------------------------------------|--------------|
| Intra-sentence ‘Yes’ case (%)    | Mono-lingual Embd                            | 99                  | 98                           | 99                  | 99                                   | 99           |
|                                   | Cross-lingual Embd                           | 99                  | 99                           | 100                 | 99                                   | 99           |
|                                   | Hindi; Marathi                              |                     |                              |                     |                                      |              |
| Inter-sentence ‘Yes’ case (%)    | Mono-lingual Embd                            | 50                  | 36                           | 64                  | 55                                   | 52           |
|                                   | Cross-lingual Embd                           | 64                  | 55                           | 52                  | 48                                   | 46           |
|                                   | Hindi; Marathi                              |                     |                              |                     |                                      |              |
| Inter-sentence ‘No’ case (%)     | Mono-lingual Embd                            | 5                   | 6                            | 2                   | 3                                    | 2            |
|                                   | Cross-lingual Embd                           | 6                   | 2                            | 3                   | 1                                    | 4            |
|                                   | Hindi; Marathi                              |                     |                              |                     |                                      |              |
| Inter-sentence ‘No’ case (%)     | Mono-lingual Embd                            | 80                  | 85                           | 67                  | 73                                   | 77           |
|                                   | Cross-lingual Embd                           | 85                  | 73                           | 77                  | 74                                   | 76           |
|                                   | Hindi; Marathi                              |                     |                              |                     |                                      | 88           |

- Example 8 (Intra-Sentence ‘No’ Case): madhubani k lakhnaur prakhand key deep pushchimi panchayat mein andhitufan or olavrishhti ki vajah se nahar kinare pakad ka vishal vriksh dharashayi hawker vahan chay ki dukhan par jaa gira.

- Transliteration: madhubani k lakhnaur prakhand key deep pushchimi panchayat mein andhitufan or olavrishhti ki vajah se nahar kinare pakad ka vishal vriksh dharashayi hawker vahan chay ki dukhan par jaa gira.

- Translation: In Deep West Panchayat of Lakhnaur block of Madhubani, due to thunderstorm and hailstorm, a huge tree of pakad collapsed on the banks of the canal and fell on a tea shop there.

- mBERT + Monolingual Experiment Prediction: ‘No’
- mBERT + Cross-lingual Experiment Prediction: ‘Yes’

- Example 9 (Intra-Sentence ‘No’ Case): haridaspur b es ef camp sahkari commander centil kumar bolun ef gully chalal di oper theke b d aro gully chalay.

- Transliteration: haridaspur b es ef camp sahkari commander centil kumar bolun ef gully chalal di oper theke b d aro gully chalay.

- Translation: In response to the attack, israeli military helicopters attacked gaza city in the early hours of this morning.

- mBERT + Monolingual Experiment Prediction: ‘No’
- mBERT + Cross-lingual Experiment Prediction: ‘Yes’

- Example 10 (Intra-Sentence ‘No’ Case): oih hamlar jababe aaj bhor rate israeli sena helicopter niye akraman chalay gaza shahare.

- Transliteration: oih hamlar jababe aaj bhor rate israeli sena helicopter niye akraman chalay gaza shahare.

- Translation: In response to the attack, israeli military helicopters attacked gaza city in the early hours of this morning.

- mBERT + Monolingual Experiment Prediction: ‘No’
- mBERT + Cross-lingual Experiment Prediction: ‘Yes’

VII. CONCLUSION

This work develops six non-BERT-based and two multilingual BERT-based deep learning models for event-argument linking in monolingual and cross-lingual settings. Non-BERT-based models, namely RCNN, Bi-LSTM-CNN, Bi-LSTM-Self-Attention, Bi-LSTM-Attention, PCNN, and CNN use fastText word embeddings and three features, namely Position Embeddings, Position Indicators, and Segment IDs as input features. The multilingual BERT-based models use similar features. We compare the performance of both types of models. We also perform an ablation study of various features in non-BERT-based deep learning architectures for event-argument linking task. This work aims to understand the effect of masked language model-based input representation over static word embeddings like fastText in both monolingual and cross-lingual knowledge transfer for event-argument linking. We choose the Hindi, Bengali, and...
Marathi language datasets for the experiment. We observe that for monolingual experiments, the non-BERT-based models like RCNN, Bi-LSTM-CNN with full feature set have similar or better performance than BERT-based models. However, BERT-based models outperform the non-BERT-based models for cross-lingual experiments by a significant margin. As the argument of any event trigger span across different sentences, we also analyze four different scenarios viz. intra-sentence ‘Yes’, inter-sentence ‘Yes’, intra-sentence ‘No’, inter-sentence ‘No’ of event-argument linking task. These scenarios are based on the position of event and argument triggers in a pair of sentences. We show that intra-sentence ‘No’ is the most difficult to predict out of these four scenarios, which requires semantic knowledge of word representation and other information like position, distance, and segment information. From the result, we observe that for cross-lingual knowledge transfer, the performance of inter-sentence ‘Yes’ and intra-sentence ‘No’ scenarios have sufficient scope of improvement. Also, predicting the class of events and arguments along with event-argument linking is essential as a future task. In the future, we will also explore the influence of external knowledge to improve performance.

ACKNOWLEDGMENT
The research work reported in this paper is an outcome of the project titled “A Platform for Cross-lingual and Multi-lingual Event Monitoring in Indian Languages”, supported by IMPRINT-I, Ministry of Education, Govt. of India, and MeITY, Govt. of India. Sovan Kumar Sahoo gratefully acknowledges “Visvesvaraya Ph.D. Scheme for Electronics and IT”, under the Ministry of Electronics and Information Technology, Government of India. Asif Ekbal acknowledges the Young Faculty Research Fellowship (YFRF), supported by Visvesvaraya Ph.D. Scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeITy), Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia).

REFERENCES
[1] Y. Kim, “Convolutional neural networks for sentence classification,” 2014, arXiv:1408.5882.
[2] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, “Relation classification via convolutional deep neural network,” in Proc. 25th Int. Conf. Comput. Linguistics: Tech. Papers (COLING), 2014, pp. 2335–2344.
[3] T. H. Nguyen and R. Grishman, “Relation extraction: Perspective from convolutional neural networks,” in Proc. 1st Workshop Vector Space Modeling Natural Lang. Process., 2015, pp. 39–48.
[4] C. Nogueira dos Santos, B. Xiang, and B. Zhou, “Classifying relations by ranking with convolutional neural networks,” 2015, arXiv:1504.05850.
[5] K. Xu, Y. Feng, S. Huang, and D. Zhao, “Semantic relation classification via convolutional neural networks with simple negative sampling,” 2015, arXiv:1506.07650.
[6] Y. Shen and X.-J. Huang, “Attention-based convolutional neural network for semantic relation extraction,” in Proc. 26th Int. Conf. Comput. Linguistics: Tech. Papers (COLING), 2016, pp. 2526–2536.
[7] L. Wang, Z. Cao, G. De Melo, and Z. Liu, “Relation classification via multi-level attention CNN,” in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, 2016, pp. 1298–1307.
[8] B. Li, X. Zhao, S. Wang, W. Lin, and W. Xiao, “Relation classification using revised convolutional neural networks,” in Proc. 4th Int. Conf. Syst. Informat. (ICSM), Nov. 2017, pp. 1438–1443.
[9] F. Ren, D. Zhou, Z. Liu, Y. Li, R. Zhao, Y. Liu, and X. Liang, “Neural relation classification with text descriptions,” in Proc. 27th Int. Conf. Comput. Linguistics, 2018, pp. 1167–1177.
[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2014, arXiv:1412.6572.
[11] D. Zhang and D. Wang, “Relation classification via recurrent neural network,” 2015, arXiv:1508.01086.
[12] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, “Classifying relations via long short term memory networks along shortest dependency paths,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2015, pp. 1785–1794.
[13] Y. Liu, F. Wei, S. Li, H. Ji, M. Zhou, and H. Wang, “A dependency-based neural network for relation classification,” 2015, arXiv:1507.04646.
[14] N. Peng, H. Poon, C. Quirk, K. Toutanova, and W. Yih, “Cross-sentence n-ary relation extraction with graph LSTMs,” Trans. Assoc. Comput. Linguistics, vol. 5, pp. 101–115, Dec. 2017.
[15] R. Zhang, F. Meng, Y. Zhou, and B. Liu, “Relation classification via recurrent neural network with attention and tensor layers,” Big Data Mining Anal., vol. 1, no. 3, pp. 234–244, 2018.
[16] M. Miwa and M. Bansal, “End-to-end relation extraction using LSTMs on sequences and tree structures,” in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, 2016, pp. 1105–1116.
[17] N. Thang Vu, H. Adel, F. Gupta, and H. Schütze, “Combining recurrent and convolutional neural networks for relation classification,” 2016, arXiv:1605.07333.
[18] R. Cai, X. Zhang, and H. Wang, “Bidirectional recurrent convolutional neural network for relation classification,” in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, 2016, pp. 756–765.
[19] X. Zhang, F. Chen, and R. Huang, “A combination of RNN and CNN for attention-based relation classification,” Proc. Comput. Sci., vol. 131, pp. 911–917, Jan. 2018.
[20] Z. He, W. Chen, Z. Li, M. Zhang, W. Zhang, and M. Zhang, “SEE: Syntax-aware entity embedding for neural relation extraction,” 2018, arXiv:1801.03603.
[21] J. Feng, M. Huang, L. Zhao, Y. Yang, and X. Zhu, “Reinforcement learning for relation classification from noisy data,” in Proc. AAAI, 2018, pp. 5779–5786.
[22] D. Zeng, Y. Dai, F. Li, R. Sherratt, and J. Wang, “Adversarial learning for distant supervised relation extraction,” Comput. Mater. Continu., vol. 55, no. 1, pp. 121–136, 2018.
[23] X. Zeng, S. He, K. Liu, and J. Zhao, “Large scaled relation extraction with reinforcement learning,” in Proc. 32nd AAAI Conf. Artif. Intell., 2018, pp. 5658–5665.
[24] M. Faruqui and S. Kumar, “Multilingual open relation extraction using cross-lingual projection,” 2015, arXiv:1503.06450.
[25] Y. Lin, Z. Liu, and M. Sun, “Neural relation extraction with multi-lingual attention,” in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, 2017, pp. 34–43.
[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5998–6008.
[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.
[28] S. Wu and Y. He, “Enriching pre-trained language model with entity information for relation classification,” in Proc. 29th ACM Int. Conf. Inf. Knowl. Manage., Nov. 2019, pp. 2361–2364.
[29] J. Liu, X. Duan, R. Zhang, Y. Sun, L. Guan, and B. Lin, “Relation classification via BERT with piecewise convolution and focal loss,” PRS ONE, vol. 16, no. 9, Sep. 2021, Art. no. e0257092.
[30] D. Zeng, K. Liu, Y. Chen, and J. Zhao, “Distant supervision for relation extraction via piecewise convolutional neural networks,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2015, pp. 1753–1762.
[31] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 318–327, Feb. 2020.
[32] L. Baldini Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski, “Matching the blanks: Distributional similarity for relation learning,” 2019, arXiv:1906.03158.
[33] S. K. Sahoo, S. Saha, A. Ekbal, and P. Bhattacharyya, “A platform for event extraction in Hindi,” in Proc. 12th Lang. Resour. Eval. Conf., 2020, pp. 2241–2250.
[34] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” J. Mach. Learn. Res., vol. 12, pp. 2493–2537, Aug. 2011.

[35] C. Liu, W. Sun, W. Chao, and W. Che, “Convolution neural network for relation extraction,” in Proc. Int. Conf. Adv. Data Mining Appl. Berlin, Germany: Springer, 2013, pp. 231–242.

[36] S. Lai, L. Xu, K. Liu, and J. Zhao, “Recurrent convolutional neural networks for text classification,” in Proc. 29th AAAI Conf. Artif. Intell., 2015, pp. 2267–2273.

[37] J. Zhang, Y. Li, J. Tian, and T. Li, “LSTM-CNN hybrid model for text classification,” in Proc. IEEE 3rd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC), Oct. 2018, pp. 1675–1680.

[38] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” 2014, arXiv:1409.0473.

[39] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, “Learning word vectors for 157 languages,” in Proc. Int. Conf. Lang. Resour. Eval. (LREC), 2018, pp. 3483–3487.

[40] S. L. Smith, D. H. P. Turban, S. Hamblin, and N. Y. Hammerla, “Offline bilingual word vectors, orthogonal transformations and the inverted softmax,” 2017, arXiv:1702.03859.

SOVAN KUMAR SAHOO received the B.Tech. degree from the College of Engineering and Management, Kolaghat, in 2010, and the M.Tech. degree from IIEST, Shibpur, in 2016. He is currently a Senior Research Fellow at the Department of Computer Science and Engineering, IIT Patna. He has published papers in various peer-reviewed conferences of international repute. His main research interests include natural language processing, information extraction, and knowledge graph.

SAUMAJIT SAHA received the B.Tech. degree in computer science and engineering from the Heritage Institute of Technology, Kolkata, in 2017, and the M.Tech. degree in mathematics and computing from IIT Patna, in 2019. He has published research articles in various peer-reviewed conferences of international repute. His research interests include information extraction, natural language processing, and conversation AI.

ASIF EKBAL (Senior Member, IEEE) is currently an Associate Professor with the Department of Computer Science and Engineering, IIT Patna. For the last 14 years, he has been pursuing research in natural language processing, information extraction, text mining, and machine learning applications. He is involved with different sponsored research projects in the broad areas of artificial intelligence and machine learning technologies, funded by government and private agencies. He was a recipient of the Visvesvaraya Young Faculty Research Fellowship (YFRF), Government of India, the JSPS Fellowship Award from the Government of Japan, and the Innovative Project Award from the Indian National Academy of Engineering (INAЕ).

PUSHPAK BHATTACHARYYA is currently a Professor of computer science and engineering at IIT Bombay. He is an outstanding researcher in natural language processing and machine learning. He has contributed to all areas of natural language processing, encompassing machine translation, sentiment and opinion mining, cross-lingual search, and multilingual information extraction. He has held the office of the Association for Computational Linguistics (ACL), the highest body of NLP, as the President. He has received several awards and fellowships, such as a fellow of Indian National Academy of Engineering, the IBM Faculty Award, and the Yahoo Faculty Award.

* * *