

Event Argument Extraction using Causal Knowledge Structures

Debanjana Kar¹, Sudeshna Sarkar², and Pawan Goyal³

Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur.

¹debanjana.kar@iitkgp.ac.in

^{2,3}{sudeshna, pawang}@cse.iitkgp.ac.in

Abstract

Event Argument extraction refers to the task of extracting structured information from unstructured text for a particular event of interest. The existing works exhibit poor capabilities to extract causal event arguments like Reason and After Effects. Furthermore, most of the existing works model this task at a sentence level, restricting the context to a local scope. While it may be effective for short spans of text, for longer bodies of text such as news articles, it has often been observed that the arguments for an event do not necessarily occur in the same sentence as that containing an event trigger. To tackle the issue of argument scattering across sentences, the use of global context becomes imperative in this task. In our work, we propose an external knowledge aided approach to infuse document level event information to aid the extraction of complex event arguments. We develop a causal network for our event-annotated dataset by extracting relevant event causal structures from ConceptNet and phrases from Wikipedia. We use the extracted event causal features in a bi-directional transformer encoder to effectively capture long-range inter-sentence dependencies. We report the effectiveness of our proposed approach through both qualitative and quantitative analysis. In this task, we establish our findings on an event annotated dataset in 5 Indian languages. This dataset adds further complexity to the task by labeling arguments of entity type (like Time, Place) as well as more complex argument types (like Reason, After-Effect). Our approach achieves state-of-the-art performance across all the five languages. Since our work does not rely on any language specific features, it can be easily extended to other languages as well.

1 Introduction

Event argument extraction is a key information extraction task that extracts structured informa-

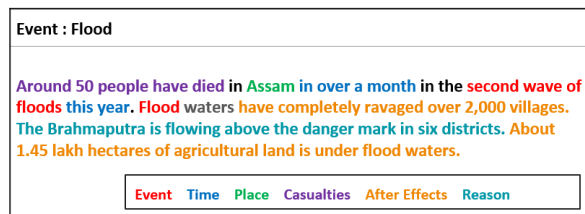


Figure 1: Sample document with annotated events and arguments. The global event category of the document is *Flood* and the words in red indicate the event triggers.

tion from unstructured texts. A widely studied task, it often involves multiple complex sub-tasks like entity retrieval, trigger detection and event-entity linking. In almost all the existing works, the terms event arguments and entities have been used interchangeably. Popular entity types which are frequently mined are that of place, time, organisations, and persons of interest. Our data set contains arguments of entity type, like Time and Place along with more complex argument types, like After Effects and Reason. These arguments with non-entity type attributes have comparatively much fewer instances in the data set than the entity type arguments. Not only that, these arguments also constitute a much more complex lexical and syntactical structure and are hard to capture using the existing architectures. In our work, we propose a novel event feature that helps capture rare complex argument instances more accurately than the existing models on this task. We construct a causal knowledge structure with the aid of external knowledge bases and for a given event, we extract its corresponding causal attribute from the structure.

Context has always been of paramount importance for the task of event argument extraction. Existing literature outlines different paradigms of modeling the contextual scope for this task. The

traditional sentence level event argument extraction tasks performed on popular datasets like ACE 2005 and TAC-KBP 2017 largely restrict their contextual scope to within sentence boundaries (Chen et al., 2015; Subburathinam et al., 2019). However, some researchers identified the need to mine global contextual features to enhance the argument extraction capabilities and modelled the task at a multi-sentence or paragraph level (Yang and Mitchell, 2016; Duan et al., 2017; Zhao et al., 2018), and at a document level (Yang et al., 2018; Zheng et al., 2019). In our work, we also identify the need to explore cross-sentence contextual scope to aid accurate extraction of events and argument spans. For example as illustrated in Figure 1, the sentence "The Brahmaputra is flowing above the danger mark in six districts." could refer to a *Reason* or an *After-Effect* argument in different contexts. But on reading the document, one can understand that the document talks about a *Flood* event, and that the sentence can be tagged as *Reason* argument with considerable confidence. Event arguments do not necessarily co-occur in the sentence containing the event trigger. This behaviour is even more frequently observed in longer documents like news articles. With no thematic signals obtained from event trigger words, it becomes difficult to identify the correct argument labels and their spans in such cases. In our work, we look beyond sentences for stronger contextual clues to better capture the events and their arguments from a document.

In summary, the **contributions** of our work are four fold :

1. We propose a novel event causal feature for improved extraction of complex causal arguments.
2. Through our work, we emphasize on the effectiveness of global vs. local scope of contextual information in this task.
3. We compare different approaches of modeling global information in this task and evaluate the effectiveness of each.
4. We provide a novel end-to-end system for event argument extraction which beats the state-of-the-art model's performance.

1.1 Terminology

We introduce certain terminologies to facilitate better understanding of our work .

- **Event** : An **event** is any real life happening or occurrence which may be denoted by a word or a phrase.
- **Event Trigger** : The particular word or phrase that evokes a particular event type is known as an **event trigger**.
- **Argument** : The words or phrases which provide supporting information about the event are known as **arguments**.
- **Causal Argument** : The causal arguments refer to the **Reason & After Effect** arguments of an event. The *Reason* argument holds information regarding why a particular event happened whereas *After Effects* argument details information about the after-math of an event. It is of importance to note here that the deaths and injuries that occur because of an event are noted under the *Casualties* argument whereas all other effects of an event are covered under the *After Effects* argument .
- **Entity** : Words or phrases which specifically refer to terms that represent real-world objects like people, places, organizations are known as **entities**.

2 Related Work

Event Argument Extraction is a well researched domain, primarily in English (Chen et al., 2015), (Nguyen et al., 2016) and Chinese (Lin et al., 2018), (Yang et al., 2018). While there have been efforts to extend this task to a limited number of languages like Spanish and Arabic (Akbik et al., 2016), (Subburathinam et al., 2019), the task has been hardly explored in Indian languages. In this paper, we make an effort to explore five out of the twenty two commonly spoken languages of India. The existing literature for Indian languages for this task are minimal and have either extracted arguments from unstructured text irrespective of their event links (Ahmad et al., 2019), or have jointly extracted event and argument mentions with rule based approaches (Patel et al., 2019). Each of these works are modelled to focus at an intra-sentence local scope and do not capture inter sentence dependencies. While the sentence level extraction mechanisms outlined by prior works present a promising preface for this task in low-resource languages, the results are indicative of a huge scope of improvement - especially for arguments like *Reason*

which are scant on annotations. In our work, we try to address this gap and extract sentence level events and arguments by infusing document level context. Infusing cross-sentence or global context to extract sentence level information has already shown great improvements in many related works. The work by (Zhang et al., 2020) studies the event-argument extraction task by infusing contextual information from the five sentences surrounding an event trigger. Their work highlights the fact that learnt architectures usually capture arguments from sentences containing event triggers better than from non-event-trigger sentences. Addressing this gap, we adopt a trigger-less event detection approach in our work, similar to (Zheng et al., 2019) and use document-level event information to mitigate the dependence on local event clues obtained from event triggers. The method adopted by (Wadden et al., 2019) uses a graphical span enumeration approach with a document input to model cross-task and inter-sentence dependencies. We model our work to include inter-sentential context to extract event-argument spans from input documents with greater accuracy.

3 Dataset

We have curated an event argument annotated dataset for English and four Indian languages, namely Bengali, Hindi, Marathi and Tamil as part of a collaborative effort. The dataset comprises of documents with sentence-level annotations of events and argument mentions. The data set caters specifically to the disaster domain and covers 32 event types at a fine grain level and 12 event types at a coarse level. The dataset contains annotations for 14 argument types, but in this work we focus on 6 main argument types which are, *Time*, *Place*, *Casualties*, *After-Effect*, *Reason* and *Participant*. Almost all possible man-made and natural disaster event types have been covered in this corpus. Typically used datasets in this task like ACE 2005 and TAC KBP cater to generic events with only entity type arguments. This corpus contains arguments of both entity type and non entity type with widely varying argument boundaries. Non-entity type arguments constitute of complex argument types like *Reason* and *After-Effects* which may constitute one/many entities within it’s span. This makes the task even more challenging and contributes to it’s uniqueness. The raw data for the corpus was collected from the FIRE document

Language	Train	Valid	Test
English	456	56	131
Bengali	699	100	199
Hindi	678	150	194
Marathi	815	117	233
Tamil	1085	155	311

Table 1: The number of documents for each language in their Train, Test and Validation splits.

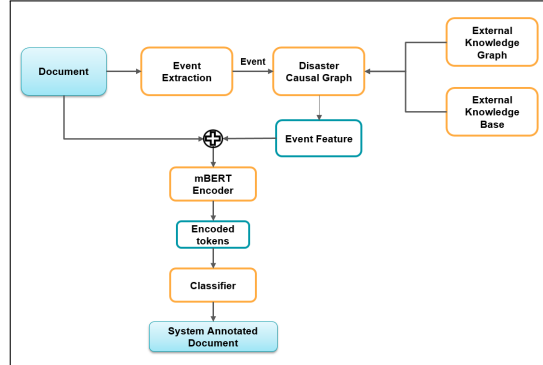


Figure 2: Overview of the general architecture.

repository as well as by crawling popular news websites for the respective languages. The dataset was annotated by eight linguistic experts over a span of two years. The multi-rater Kappa agreement ratio of the annotators has been evaluated as 0.85 approximately. The data distribution is reported in Table 1

4 Approach

In this section we elaborate on the approaches we have taken to accomplish the task of extracting the event labels and it’s corresponding arguments from a document. This is essentially a two-fold task: i) Document Level Event Extraction, and ii) Argument Extraction. The generic overview of our approach is illustrated in Figure 2.

4.1 Problem Formulation

In our work, we treat the task of event-argument extraction as a sequence labeling task. Given N training instances $D = (x_n, y_n)_{n=1}^N$, where x is a sequence of tokens from an input document and y is the corresponding event-argument annotation sequence for x , our aim is to build a neural model which maximises the conditional probability $p_\theta = (y|x)$, parameterized by model parameters θ . To capture events and arguments better, we propose modeling context at a global scope for this task. Infusion of global context in our work is done

Language	Micro F1-Score (%)
English	97.71
Bengali	68.34
Hindi	87.63
Marathi	78.54
Tamil	84.89

Table 2: Document Level Event Extraction Results

in two ways. First, instead of processing each sentence as a training instance, we process an entire document with sentence demarcations. Second, we infuse the document level event information into the model to aid more accurate identification of complex argument types. In the sections to follow, we have described the different approaches we have adopted to process event information into the model. We have conducted our experiments on five Indian languages.

4.2 Document Level Event Extraction

Traditional approaches for event extraction involve extracting event triggers at a sentence level. We adopt a no-trigger approach to label the documents with their corresponding events but instead of adopting a distant supervision method like (Zheng et al., 2019), we adopt a supervised approach with the aid of pre-trained language models in this task. Our dataset comprises of news articles, and most often, it has been observed that the central theme of such documents are divulged in the title of the document itself. Moreover, we find that only 6% of the total corpus contains documents with multi-event instances. Hence, we formulate this task as a multi-class sentence classification task where each document corresponds to one event category. We consider the event labels of the first event trigger instance in each document as the document’s manually assigned event label. We fine-tune a pre-trained multilingual BERT (mBERT) encoder with a linear classification head on top to classify each document title instance to an event category. Since the titles of the documents are not explicitly defined, we take the first two sentences as the title of the news article and use it as our training and testing instances. We use the the pooled output of the encoder as input to the linear classification head in the model. We find this approach to be highly effective with a very high percentage of documents being assigned their correct event labels as shown in Table 2.

4.3 Argument Extraction

The main focus of this sub-task is to extract the sentence level arguments given the document-level event information. To aid the extraction of causal arguments, we introduce a causal feature along with the input document to the argument extraction module. In the following sections, we discuss how we build and use the causal feature for event argument extraction.

4.3.1 Event Causal Feature

Unlike the generic entity type arguments like *Time* and *Place*, causal type arguments follow complex patterns, ranging over variable lengths and show a strong reliance on the event category. This is a natural observation - the cause and effects of an Earthquake will not be similar to that of a flood or a terrorist attack. To make the task even more challenging - these argument types, especially the *Reason* argument, have very few training instances in the corpus. We define a novel event causal feature in our approach to aid the extraction of causal arguments.

Causal Graph Construction We manually construct a disaster event graph ontology based on the structures defined in the annotation guidelines of the corpus as well as background knowledge of linguistic experts. Given a graph $G = (V, E)$, the vertices V correspond to events and their causal argument roles (Reason & After-Effects) and the edges E show the relationships among the various events and their arguments. We manually define the event-event relationships in the graph to aid knowledge transfer across events, such that empty nodes in the graph corresponding to argument roles, can inherit knowledge for that specific argument role.

Node population The event nodes contain the event-types. The children nodes correspond to the causal argument roles. These nodes are populated using words and phrases from external knowledge bases like ConceptNet (Speer et al., 2017) and Wikipedia. For ConceptNet, we have identified a few edge-relations which correspond to our causal argument labels and nodes that correspond to our event types. It is worth mentioning here, that not all the event types could be accessed in the external resources. To mitigate that, we used a list of manually curated synonyms of the event types or merged some related event types (like *Transport*

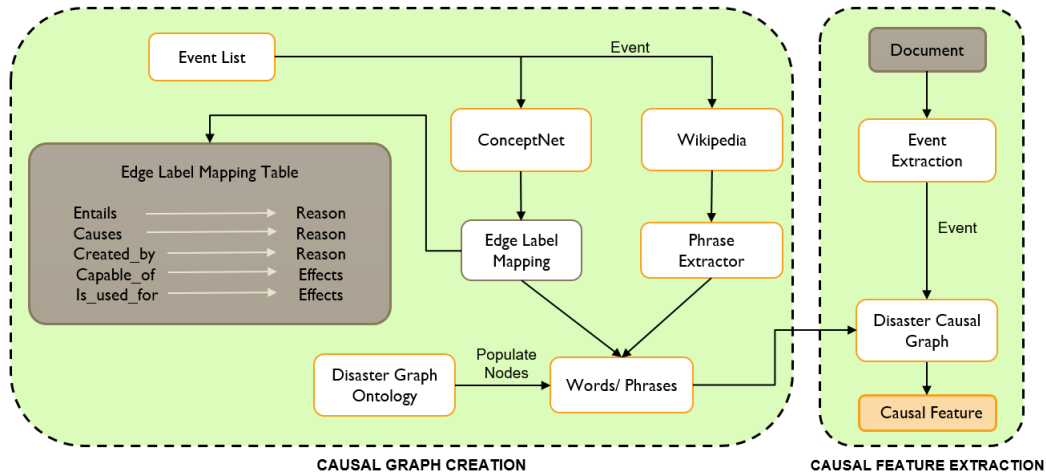


Figure 3: Detailed overview of the causal feature construction process. Since our data is of Disaster domain, we have labeled the graph in the diagram as Disaster Causal Graph.

Hazards and Vehicular Collision) to find relevant resources in the external knowledge bases. For each event type, we query the nodes associated with the identified edges, extract the words and phrases present in the nodes and use that to populate our corresponding causal graph nodes. We also crawl the cause and effects section of relevant Wikipedia pages corresponding to the event types in our corpus. We use an unsupervised language-agnostic phrase extractor (Campos et al., 2020) to extract phrases from the crawled Wikipedia sections and populate the corresponding causal graph nodes with the extracted phrases. Each argument node in the graph maintains a minimum of 3 and a maximum of 10 words or phrases. In the event that a certain argument node in the causal graph has less than 3 words or phrases, we exploit the defined event-event relations in the causal graph and inherit values from related event-argument nodes. The entire process is illustrated in Figure 3.

In an additional final step, we use a sentence template to string together the words and phrases of the argument nodes of a particular event. The constructed event-causal sentence for each event will henceforth be referred to as the event causal feature in the paper.

4.3.2 Argument Span Identification & Classification

Following the pipeline illustrated in Figure 2, for each document, we extract its event category and use the extracted event to obtain the corresponding event causal feature. Each input instance to the argument extraction module is augmented with

the event causal feature at both ends. We train all the 12 encoder layers of the pre-trained bert-base-multilingual-cased model along with an additional linear classification head on top. The linear layer classifies the hidden states output corresponding to each token to one of the six argument types (*Time, Place, Casualties, After-Effects, Reason, Participant*) or to *Others*.

In the sections to follow, we will refer to our approach defined in this section (Section 4) as **Event Causality Augmentation** or **ECA**.

5 Results

5.1 Experimental Settings

We use Huggingface’s bert-base-multilingual-cased model pre-trained on 104 languages¹. Adam (Kingma and Ba, 2014) was used to optimize the parameters. The model was trained with a mini-batch size of 4 on a single Tesla k40-C machine and a maximum sequence length of 512. The models were trained for 20 epochs or until no significant changes were observed in the validation loss, whichever was earlier.

5.2 Baselines

We compare our work with the following enlisted approaches:

1. Patel-Emb: (Patel et al., 2019) are considered the state-of-the-art results on this dataset. The works of (Patel et al., 2019) employ a set of rules to create a rule vector for each token

¹<https://huggingface.co/bert-base-multilingual-cased>

Models	Time	Place	Casualties	After-Effects	Reason	Participant	Avg.
Patel-Emb	83.05	88.41	90.09	55.90	81.56	76.40	79.23
Patel-Parallel	84.65	87.48	89.09	56.48	86.31	75.42	79.90
Patel-KD	81.74	87.19	89.83	47.79	87.35	75.45	78.23
mBERT	91.50	92.80	93.50	55.30	85.90	84.00	83.80
JETAE	90.37	91.84	92.87	47.42	77.07	83.41	80.49
EA	90.90	92.90	93.60	54.90	83.60	86.40	83.71
ECA (Our model)	92.65	93.41	94.22	58.39	90.76	87.29	86.12

Table 3: Argument-Wise comparison of our approach (ECA) with other baseline approaches on the English Corpus using F1-scores. The bold text refers to the best result obtained for each argument as well as overall (Avg.).

which embeds event class information. They use different approaches to infuse the rule vector into the model. In this approach, the rule vector was appended with each sentence token representations and fed to a single layer Bi-LSTM model.

2. Patel-Parallel : In this approach of (Patel et al., 2019), the rule vector and word embeddings were fed to two parallel single layer Bi-LSTM models. Their learnt hidden layer representations were concatenated to learn a joint representation of words and rules.
3. Patel-KD: In this approach of (Patel et al., 2019), the knowledge of rules is distilled to the Bi-LSTM network by using the rule vector to bias the weights of the neural network.
4. mBERT: We adopt the BERT-NER from (Devlin et al., 2019) and fine-tune the mBERT Token Classifier to extract event-argument spans and compare it with our feature based mBERT argument extractor.
5. JETAE: We adapt the mBERT Token Classifier for Joint Event Trigger and Argument Extraction task as defined in (Patel et al., 2019) and report it’s argument extraction capabilities as compared to our approach.
6. EA (Event Augmentation): Instead of augmenting the event causal feature, we augment the event type at both ends of the document and compare it’s results against our’s.

We compare the overall performance in Table 3 for English. For the Indian languages, we compare the F1-scores we have obtained using our approach for each argument type against the corresponding F1 score obtained from (Patel et al., 2019) in Table 5. Since none of the three approaches defined in (Patel

Arg. Type	ECA-gold	ECA-pred
Time	91.84	92.19
Place	71.81	70.82
Casualties	82.96	82.79
After-Effects	53.31	53.13
Reason	42.71	39.88
Participants	55.11	54.00

Table 4: Results to highlight error propagation for the Bengali corpus. We report the F1-scores for each argument type with manually annotated event labels (ECA-gold) and predicted event labels (ECA-pred).

et al., 2019) were reported as the best performing method for the task, for a given natural language, we have considered the highest F1-score obtained for each argument type among the three approaches as the state-of-the-art result for the same. We report our results using precision, recall and f1-scores.

6 Analysis

We perform a thorough analysis of our approaches and enlist their merits and demerits in this section. We compare our models to the state-of-the-art models established on this dataset. As reported in Table 3, we establish new state-of-the-art results on this dataset. We present a comparative analysis of the approaches explored in our work and observe that our approach Event Causality Augmentation (ECA) reports the best performance (Table 3). In the table, we can observe a huge performance boost with the use of fine-tuned contextualised word representations of mBERT. The intended performance boost in the extraction of causal arguments is also observed on introduction of the causal feature. We report the argument extraction performance of ECA with both manually annotated event labels (ECA-gold) and event labels predicted by our event extraction module (ECA-pred) in Table 4. We specif-

Lang.	Arg. Type	SOTA	ECA	$\Delta\%$
Bengali	Time	88.59	92.19	+4.06
	Place	68.20	70.82	+3.84
	Casualties	78.95	82.79	+4.86
	After Effects	41.54	53.13	+27.90
	Reason	17.91	39.88	+122.67
	Participant	49.93	54.00	+8.15
Hindi	Time	68.57	72.54	+5.79
	Place	63.79	73.06	+14.53
	Casualties	68.21	71.93	+5.45
	After Effects	44.67	45.43	+1.70
	Reason	19.25	12.42	-35.48
	Participant	43.18	56.24	+30.25
Marathi	Time	75.59	80.97	+7.12
	Place	69.44	74.74	+7.63
	Casualties	75.69	79.48	+5.01
	After Effects	47.29	59.18	+25.14
	Reason	35.93	35.57	-1.00
	Participant	59.35	64.55	+8.76
Tamil	Time	81.03	89.09	+9.95
	Place	75.86	83.84	+10.52
	Casualties	80.20	87.21	+8.74
	After Effects	72.15	83.82	+16.17
	Reason	42.72	67.94	+59.04
	Participant	65.71	74.61	+13.54

Table 5: Comparison of Argument Extraction Performance of our model (ECA) with the state-of-the-art (SOTA) results ((Patel et al., 2019) for each language. We show the % increase (+) or decrease (-) in F-Scores with respect to the SOTA results.

ically show these results on the Bengali dataset as it had reported the weakest event extraction results among the five languages in Table 2. In table 2 we observe that the F1-scores for Bengali and Marathi are considerably low compared to the other languages. The drop in performance can be attributed to the fact that these two languages contain the maximum number of multi-event documents while our approach is based on the assumption that each document in the corpus is a single event document. Even in multi-event documents, we had observed that the dominant event was mostly present in the title of the document. The Bengali corpus consisted of a few exceptional multi-event news articles which contained news snippets about completely unrelated topics. In such cases the event extraction algorithm performs poorly, thus reporting a signif-

<p>Ground Truth: सड़क निर्माण के दौरान <Time> पहाड़ी के गिरने से मलबे में दबकर दो मजदूरों <After-Effects> की मौत हो गई।</p> <p>Translation: While road construction<Time>, due to landslide, two workers<After-Effects> were crushed under the debris and killed.</p>
<p>Predicted: सड़क निर्माण के दौरान पहाड़ी के गिरने से <Reason> मलबे में दबकर दो मजदूरों की मौत <Casualties> हो गई।</p> <p>Translation: While road construction, due to landslide<Reason>, two workers were crushed<Casualties> under the debris and killed.</p>

Figure 4: Manually annotated and Inferred example from the Hindi dataset highlighting the systemic confusion between *event triggers* and *reason* arguments. It also highlights the annotation errors existing in the corpus, which is discussed in Section 6.4.

icant drop in performance on the Bengali dataset for the document level event extraction task.

We can observe the error from the event extraction module propagating to our argument extraction method in ECA-pred with the effects being visible mostly in the causal arguments in Table 4. We leave the task of a joint event label identification and argument extraction as a future task, which should be able to mitigate this problem. In both Table 3 and Table 5, we can see the desired improvement for the causal arguments. For almost all arguments we notice a major improvement in the performance in Table 5. In Hindi & Marathi though, we observe a dip in the performance for the *Reason* argument. We find that the dip can be attributed to two factors: i) the model confuses event triggers in certain cases as the *Reason* argument - events can be the reason for other events in the document as well; ii) there are *Reason* argument instances which are not annotated but gets detected by the model. We have illustrated (i) through an example in Figure 4. In the example, it can be observed that the model confuses the event trigger *due to landslide* as the *Reason* for the death of two workers. Although, it is correct in its essence, as per the task, the phrase *due to landslide* is an event trigger and not part of a *Reason* argument.

Ground Truth	
<p>ডায়মণ্ড হারবারের এস ডি পি ও দীপশঙ্কর রুদ্র জানান আচমকা বাসটির সামনের চাকা ফেটে যাওয়ায় <Reason> রাস্তা থেকে ছিটকে সেটি পাশের একটি খালে কাত হয়ে পড়ে যায়। <event trigger></p> <p>Translation: Diamond Harbour's SDPO Dipshankar Rudra has informed that due to a sudden burst of the bus's front tires<Reason>, it skid from the road, overturned and fell into a ditch.<event trigger></p> <p><Document Event Type : Transport Hazards ></p>	
Model Type	Predicted Instance
mBERT (Without Event)	No Prediction
EA	<p>ডায়মণ্ড হারবারের এস ডি পি ও দীপশঙ্কর রুদ্র জানান আচমকা বাসটির<Participant> সামনের চাকা ফেটে<Reason> যাওয়ায় রাস্তা থেকে ছিটকে সেটি পাশের একটি খালে কাত হয়ে পড়ে যায়।</p> <p>Translation: Diamond Harbour's SDPO Dipshankar Rudra has informed that due to a sudden burst of the<Reason> bus's<Participant> front tires<Reason>, it skid from the road, overturned and fell into a ditch.</p>
JETAE	No Prediction
ECA	<p>ডায়মণ্ড হারবারের এস ডি পি ও দীপশঙ্কর রুদ্র জানান আচমকা বাসটির সামনের চাকা ফেটে যাওয়ায় রাস্তা থেকে<Reason> ছিটকে সেটি পাশের একটি খালে কাত হয়ে পড়ে যায়।</p> <p>Translation: Diamond Harbour's SDPO Dipshankar Rudra has informed that due to a sudden burst of the bus's front tires<Reason>, it skid from the road, overturned and fell into a ditch.</p>

(a) Example 1

Ground Truth	
<p>কৃষ্ণনগর স্কুলের মাঠে <Place> গাড়ি চালানো শেখার সময় চালক নিয়ন্ত্রণ হারিয়ে ফেলায় <Reason> গাড়িটির চাকায় পিষ্ট হয়ে তৃতীয় শ্রেণীর এক ছাত্রের মৃত্যু <Casualties> হয়েছে।</p> <p>Translation : While taking driving lessons<Time> in Krishna-nagar School grounds<Place>, the driver lost control<Reason> and a 3rd grade student died<Casualties> on getting crushed under the wheels of the car.</p> <p><Document Event Type : Transport Hazards, sentence has no event trigger></p>	
mBERT (Without event)	<p>কৃষ্ণনগর স্কুলের মাঠে <Place> গাড়ি চালানো শেখার সময় চালক <Reason> নিয়ন্ত্রণ হারিয়ে ফেলায় <After Effects> গাড়িটির চাকায় পিষ্ট হয়ে তৃতীয় শ্রেণীর এক ছাত্রের মৃত্যু <Casualties> হয়েছে।</p> <p>Translation: While taking driving lessons in Krishnanagar School grounds<Place>, the driver<Reason> lost<After Effects> control and a 3rd grade student died<Casualties> on getting crushed under the wheels of the car.</p>
EA	<p>কৃষ্ণনগর স্কুলের মাঠে <Place> গাড়ি চালানো শেখার সময় চালক নিয়ন্ত্রণ হারিয়ে<After Effects> ফেলায় গাড়িটির চাকায় পিষ্ট হয়ে তৃতীয় শ্রেণীর এক ছাত্রের মৃত্যু <Casualties> হয়েছে।</p> <p>Translation: While taking driving lessons in Krishnanagar School grounds<Place>, the driver lost control<After Effects> and a 3rd grade student died<Casualties> on getting crushed under the wheels of the car.</p>
ECA	<p>কৃষ্ণনগর স্কুলের মাঠে <Place> গাড়ি চালানো শেখার সময় চালক নিয়ন্ত্রণ <Reason> হারিয়ে ফেলায় গাড়িটির<Participant> চাকায় পিষ্ট হয়ে তৃতীয় শ্রেণীর এক ছাত্রের মৃত্যু <Casualties> হয়েছে।</p> <p>Translation: While taking driving lessons in Krishnanagar School grounds<Place>, the driver lost control<Reason> and a 3rd grade student died<Casualties> on getting crushed under the wheels of the car<Participant></p>

(b) Example 2

Figure 5: Various inferred examples collected for a thorough qualitative analysis. The different approaches compared are our model (ECA), Event Augmentation (EA), without event (mBERT), and Joint Event Trigger and Argument Extraction model (JETAE).

6.1 Importance of event causal information

To investigate the importance of event information in our model for the task of argument extraction, we present our findings in Table 6, where we compare mBERT based argument extraction models without event information (without-event), with event type information (EA) and finally with event causal feature (ECA - our model). We can see a significant improvement in performance with event information, especially with the causal feature. This rise can mainly be attributed to the significant improvement in causal argument extraction capabilities which rely heavily on event information. We also qualitatively analyse our findings in the first example cited in Figure 5. As can be observed, without event information, it is unable to detect the presence of a Reason Argument. With event type information, the model is able to detect the argument's presence, but we observe the model suffering from ambiguity over argument types. This ambiguity often rises with complex argument types being nested among each other, as can be observed in this example. *Bus* is a Participant in this event but in this example, it is part of longer Reason argument span. With addition of the causal feature,

we find causal contexts getting learnt better and as observed in this example, causal argument spans get captured efficiently.

6.2 Importance of document level event information

Extracting document level event information has multiple benefits compared to event trigger extraction: i) It helps capture the overall thematic preface of the document, ii) By not extracting event triggers, we avoid the problem of ambiguous event triggers for multiple event types. For example, the word *explosion* can be an event trigger for event types *Terrorist Attacks*, *Volcano*, as well as *Industrial Accidents*.; iii) It saves annotation labour and cost. Labeling document-level events is much simpler a task than annotating event triggers in each sentence of the document. By taking into account the thematic preface of the document, we will now observe its effects on the task of event argument extraction using the same example as we used above (first example of Figure 5). In JETAE, we often observe the model exhibiting bias towards sentences with event triggers when it comes to the task of argument extraction. In this example, since the

model was unable to detect the event trigger in the sentence, it is unable to detect its corresponding argument as well. Because of the use of document level event context, we can observe that both EA and our model, ECA are able to detect the *Reason* argument span.

6.3 Sentence vs. Paragraph vs. Document

We study the importance of document context in the task of extracting arguments of an event. To investigate the importance of the contextual scope, we have experimented in three different settings by formatting the input instance to either include a single sentence, a paragraph or the entire document. Since paragraph boundaries were not mentioned in the dataset, we have assumed the paragraphs in our work to constitute four sentences. This decision was taken after finding the average length of the documents to be 15 sentences. We present our findings in Table 7 where we can observe a consistent increase in the F1-scores as we widen our contextual scope. The scores in each setting were evaluated by averaging the individual argument scores. While the performances at sentence and paragraph level are comparable, it is in document level context that a significant increase is observed. This increase can be mainly attributed to the effective extraction of the scattered arguments with no localised event clues to aid in their extraction process.

6.4 Case Study

In this section, we analyse a few inferred examples across different approaches adopted in our task. Apart from the points already discussed, we observe a few more challenges and their solutions along with their merits and demerits through our approaches. If we analyse the second example in Figure 5, we find a classic confusion between the *Reason & After Effect* arguments. Reason and after-effects are often interchangeable concepts depending on the contextual premise. As can be observed, addition of causal feature helps resolve this ambiguity. We also observe that the model efficiently captures arguments which were otherwise missed by human annotators like the Participant argument *car* in this example. This is primarily because of the document context and other similar Participant instances being annotated in likewise document context that the model is able to capture the missed annotations in the document as well. We also observe, across examples that the argument boundary

	P	R	F
Without-event	66.32	62.57	63.45
EA	67.21	63.45	64.14
ECA	67.91	65.69	66.20

Table 6: Comparison of different approaches on the Bengali Dataset to establish the importance of event information and event causal feature in our task.

Model	Sentence	Paragraph	Document
EA	59.59	62.59	64.14
ECA	61.20	62.42	66.20

Table 7: Comparison of argument-extraction F1-scores with respect to the contextual scope, for the Bengali dataset of our corpus.

mismatch persists in our approaches as well. The argument window may not be precise and may differ by a few tokens on either ends of the span. Another observation that we make is more in lieu of the annotation errors in the corpus which the model has been observed to overcome in many situations. A confusion among annotators was observed about the semantic scope of *Casualties* and *After Effect* arguments as observed through the example in Figure 4. However, the model in most cases extracts the argument with the correct label (as also illustrated in Figure 4) and thus, exhibits the model’s robustness against noise.

7 Conclusion

In our work we have shown a pipeline approach to mine document level event labels and their arguments from each sentence in a document. We have reported a comparison study across different approaches based on multiple parameters, like input context and event information. In future, we want to extend this task to handle more languages in a multilingual setting. Our approaches have shown a huge improvement over the state-of-the art methods. In conclusion, through our work, we emphasize on the importance of both contextual and event information and report our thorough investigations on the same.

Acknowledgments

The work done 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-1, MHRD, Govt. of India, and MeiTY, Govt. of India.

References

- Zishan Ahmad, Deeksha Varshney, Asif Ekbal, and Pushpak Bhattacharyya. 2019. Multi-linguality helps: Event-argument extraction for disaster domain in cross-lingual and multi-lingual setting.
- Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yonas Kbrom, Yunyao Li, and Huaiyu Zhu. 2016. Multilingual information extraction with polyglotie. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 268–272.
- Ricardo Campos, Vítor Mangaravite, Arian Pasquali, A. Jorge, C. Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Inf. Sci.*, 509:257–289.
- Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Shaoyang Duan, Ruifang He, and Wenli Zhao. 2017. [Exploiting document level information to improve event detection via recurrent neural networks](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 352–361, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Diederik P. Kingma and Jimmy Ba. 2014. [Adam: A method for stochastic optimization](#).
- Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2018. Nugget proposal networks for chinese event detection. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1565–1574.
- Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. [Joint event extraction via recurrent neural networks](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.
- Hrishikesh Patel, Nandan Rathod, Ayush Maheshwari, Ritesh Kumar, Ganesh Ramakrishnan, and Pushpak Bhattacharyya. 2019. Tale of tails using rule augmented sequence labeling for event extraction. *arXiv preprint arXiv:1908.07018*.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: an open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4444–4451.
- Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. Cross-lingual structure transfer for relation and event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 313–325.
- David Wadden, Ulme Wennberg, Yi Luan, and Hananeh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5788–5793.
- Bishan Yang and Tom Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299.
- Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018. [DCFEE: A document-level Chinese financial event extraction system based on automatically labeled training data](#). In *Proceedings of ACL 2018, System Demonstrations*, pages 50–55, Melbourne, Australia. Association for Computational Linguistics.
- Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020. [A two-step approach for implicit event argument detection](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7479–7485, Online. Association for Computational Linguistics.
- Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. [Document embedding enhanced event detection with hierarchical and supervised attention](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 414–419, Melbourne, Australia. Association for Computational Linguistics.
- Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2edag: An end-to-end document-level framework for chinese financial event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 337–346.