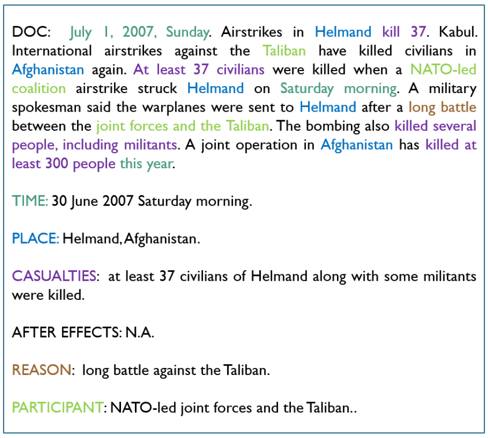
ArgGen: Prompting Text Generation Models for Document-Level Event-Argument Aggregation

Debanjana Kar [*]
IBM Research
Bengaluru, India
debanjana.kar1@ibm.com

Sudeshna Sarkar and **Pawan Goyal**
IIT Kharagpur
Kharagpur, India
{sudeshna, pawang}@cse.iitkgp.ac.in

Abstract

Most of the existing discourse-level Information Extraction tasks have been modeled to be extractive in nature. However, we argue that extracting information from larger bodies of discourse-like documents requires more natural language understanding and reasoning capabilities. In our work, we propose the novel task of document-level event argument aggregation which generates consolidated event-arguments at a document-level with minimal loss of information. More specifically, we focus on generating precise document-level information frames in a multilingual setting using prompt-based methods. In this paper, we show the effectiveness of prompt-based text generation approach to generate document-level argument spans in a low-resource and zero-shot setting. We also release the first of its kind multilingual event argument aggregation dataset that can be leveraged in other related multilingual text generation tasks as well: https://github.com/DebanjanaKar/ArgGen



DOC: July 1, 2007, Sunday. Airstrikes in Helmand kill 37. Kabul. International airstrikes against the Taliban have killed civilians in Afghanistan again. At least 37 civilians were killed when a NATO-led coalition airstrike struck Helmand on Saturday morning. A military spokesman said the warplanes were sent to Helmand after a long battle between the joint forces and the Taliban. The bombing also killed several people, including militants. A joint operation in Afghanistan has killed at least 300 people this year.

TIME: 30 June 2007 Saturday morning.

PLACE: Helmand, Afghanistan.

CASUALTIES: at least 37 civilians of Helmand along with some militants were killed.

AFTER EFFECTS: N.A.

REASON: long battle against the Taliban.

PARTICIPANT: NATO-led joint forces and the Taliban..

Figure 1: Illustrative example of the Event Argument Aggregation Task. The sentence-level event argument mentions have been highlighted in the document with colours corresponding to their argument roles (like TIME, PLACE). Multiple sentence-level arguments in the same colour in the document indicate high redundancy of information for that particular argument role.

1 Introduction

Discourse-based Information Extraction (IE)is a well-explored NLP task. Most of these works (Yang et al., 2018; Zheng et al., 2019) rely on extractive approaches to mine relevant event-argument spans for specific argument roles. However, there are two main challenges in this effort. First, extractive argument spans may miss implicit information at a document-level. For example in Figure 1, the *Time* mentions in the document include the publishing date of the document and the day of the week the event occurred. An extractive approach will not be able to accurately determine the date of the event. We aim to address this challenge using a conditional text generation approach. Second, sentence-level argument mentions in the document are often scattered and may contain similar yet distinct information. For example, the *Casualties* argument mentions like 'kill 37', 'At least 37 civilians', 'killed several people, including militants' in the example (Figure 1) contain repetitive but slightly distinct information. An extractive method extracting such document level arguments may again miss key information as they employ elimination strategies to select the key argument mention at the document level. The approach we propose addresses this challenge by leveraging argument specific prompts with conditional text generation methods.

In this paper, we provide a fresh perspective to discourse-based IE and propose the task of Event Argument Aggregation. Event Argument Aggregation is a challenging natural language understanding task that aims to consolidate document-level structured information from given unstructured text. Closely related to the task of document-level event argument extraction, event argument aggregation emphasizes on filtering redundant and irrelevant

[*] Work done as a student at IIT Kharagpur

Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022, pages 399–404
November 20–23, 2022. ©2022 Association for Computational Linguistics

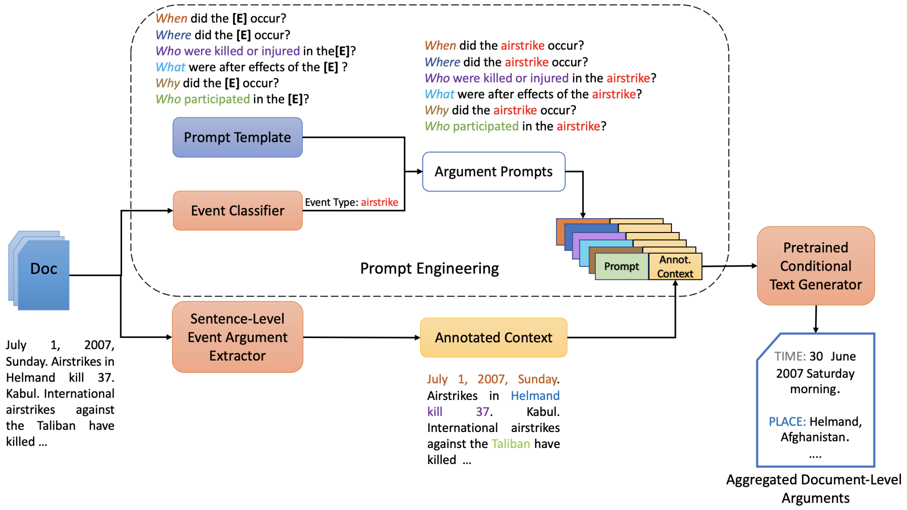

When did the [E] occur?
Where did the [E] occur?
Who were killed or injured in the[E]?
What were after effects of the [E] ?
Why did the [E] occur?
Who participated in the [E]?
When did the airstrike occur?
Where did the airstrike occur?
Who were killed or injured in the airstrike?
What were after effects of the airstrike?
Why did the airstrike occur?
Who participated in the airstrike?
Prompt Template
Event Classifier
Event Type: airstrike
Argument Prompts
Prompt
Annot. Context
Prompt Engineering
Doc
Pretrained Conditional Text Generator
July 1, 2007, Sunday. Airstrikes in Helmand kill 37. Kabul. International airstrikes against the Taliban have killed ...
Sentence-Level Event Argument Extractor
Annotated Context
July 1, 2007, Sunday. Airstrikes in Helmand kill 37. Kabul. International airstrikes against the Taliban have killed ...
TIME: 30 June 2007 Saturday morning.
PLACE: Helmand, Afghanistan.
....
Aggregated Document-Level Arguments

Figure 2: Illustration of the architecture for training our desired event-argument generation model.

argument mentions to generate precise document-level information frames. In our work, we focus on producing the document-level information frames using prompt-based generative approaches.

In our work, we adopt (Du and Cardie, 2020; Feng et al., 2020)'s idea of reducing our related task of Document-Level Event Argument Aggregation to that of Natural Language Question Answering. A very recently published work related to the task of Event Argument Generation is that of (Li et al., 2021). Like our approach, they too employ conditioned text generation to generate document-level event arguments. However, the argument spans they extract at a document level are much shorter and explicit in nature than our argument mentions. Prompt-based methods have recently gained popularity in a number of related tasks like entity extraction(Wang et al., 2022), question answering(Liu et al., 2022) and text generation(Li et al., 2022). In this paper, we show the effectiveness of prompt-based methods to aggregate event-arguments at a document-level. We evaluate our models on low-resource settings as well as more challenging zero-shot settings. We discuss and analyse the effectiveness of the proposed model in the following sections.

The key contributions of our work are enumerated as follows: i) We propose a fresh perspective to discourse-based IE through our proposed task of Event Argument Aggregation. ii) We are the first to explore prompt-based conditional text generation to aggregate event-arguments at a document level. Our proposed model provides state-of-the-art results on this task. iii) We are the first to release an annotated, multilingual event-argument aggregation dataset. The corpus consists of 346 annotated documents in English, Hindi and Bengali.

2 Event-Argument Aggregation

In this section, we detail the approaches we propose for the task of Document-Level Event Argument Aggregation. The framework primarily involves three steps: i) MRC Pre-training, ii) Prompt Engineering, iii) QA-based Argument Generation.

2.1 MRC Pre-training

For the model to generate informative aggregated argument mentions at a document-level from scattered sentence-level argument mentions (as demonstrated in Fig 2), the model requires strong comprehension and reasoning capabilities. For example, given the publishing date of the article and the day of the week on which the event occurred, the model should be able to comprehend and render the correct date of the event which is not explicitly mentioned in the input document. This requirement for infusing natural language understanding and reasoning capacities in the model necessitates the machine reading comprehension (MRC) pre-training step in our proposed approach. MRC usually comprises of NLP tasks like question-answering, textual-entailment, numerical reasoning, etc. We pre-train our model on an amalgamated QA dataset (Multi_QA, Section 3.1) which consists of reasoning QA data samples in English along with other QA data samples in Hindi and Bengali. The conditional text generator we use for this task is a transformer-based encoder-decoder ar-

Dataset	DROP	MLQA	XQuAD	TyDI	Multi_QA
#Train	77409	4918	96340	3585	1,82,252
#Test	9536	507	2374	113	12,530
Q Len	10.83	9.31	11.01	5.61	10.78
A Len	1.38	3.62	3.91	3.78	2.77
P Len	202.44	155.24	136.86	87.23	165.70

Table 1: Dataset Statistics for the amalgamated QA corpus Multi_QA along with it's constituent datasets. The first two rows enumerate the number of train test instances across the datasets. *Q, P, A Len* refer to the average lengths of Questions, Passage and Answers respectively.

chitecture which takes as input an input passage P and a query q, and is trained to generate an answer a of abstractive nature. We use the multilingual variant of the T5 model as our backbone model for this task. After training the small and base variants of mT5 and mBART-50, we find that mt5-base performs the best with an F1-score of 62.45%

2.2 Prompt Engineering

Prompting the QA-based argument aggregator is fairly intuitive. Given the argument roles, we design templates for the prompts like *When did [E] happen?* where $[E] \in$ disaster-based events like $\{earthquake, flood, terrorist\_attack, ..\}$. Since the number of argument-roles are limited, we manually define the prompts instead of generating them automatically for greater accuracy. We define our prompts using 5W words (*When, Where, What, Who, Why*) and it has been observed empirically that the prompts with 5Ws work better in such QA-based frameworks(Liu et al., 2022). To fill the event mask $[E]$ in the prompt, we define a document classifier which identifies the event type of the document. A classification head on top of multilingual BERT is trained iteratively to map the correct event-type to the input document instance. Since for each of m argument roles, we define a specific prompt, we hence refer to the prompts as *Argument Prompts*.

2.3 QA-based Argument Generation

Given a document, we parse the document to annotate sentence-level argument mentions. We extract sentence-level argument information from the document using the state-of-the-art event argument extraction method for this dataset (Kar et al., 2020). It uses causal knowledge structures to accurately detect the low-resource event argument mentions in the document's sentences. We mark the sentence-level argument spans in the document with special argument role tokens to generate our annotated context. We avoid marking duplicate argument mentions and mentions with very similar surface form in the document to curtail redundancy in the model. Using fuzzy string match techniques (Levenshtein, 1965), we only mark the longer argument span in case of redundancy. The annotated context is concatenated with an argument-specific prompt and used as the input to the pre-trained conditional text generator. The conditional text generator, pretrained with an MRC objective in the previous step, is fine-tuned with few examples to generate the desired document-level aggregated argument mentions for a specific argument role. Our results and analysis in the following sections highlight that our proposed framework effectively generates meaningful aggregated argument mentions even after seeing only a few examples for each language.

3 Dataset

In the sections to follow, we discuss the details of the datasets we created for i) the MRC pretraining task and ii) Multilingual Event Argument Aggregation (*ArgGen* dataset).

3.1 MRC Pretraining Dataset

Most of the works in the domain of Natural Language Question Answering are of extractive nature. However, for the task of MRC Pretraining (as discussed in Section 2.1), we required an abstractive multilingual question answering dataset. We curate such a dataset by collating the following datasets: i) DROP Dataset (Dua et al., 2019) which is an abstractive, reasoning QA dataset with a special focus on numerical reasoning; ii) Hindi annotated instances of MLQA (Lewis et al., 2020) and XQuAD (Artetxe et al., 2020) datasets and iii) Bengali annotated instances from TyDi QA dataset (Clark et al., 2020). Although the multilingual datasets collated are extractive in nature, we use them in generative pretraining along with the abstractive DROP

Dataset	Eng.	Ben.	Hindi	Multi
# Docs	129	75	142	346
#Train Inst.	619	360	681	1660
#Test Inst.	155	90	171	416
Avg. Ans Len	7.2	9.3	11.0	9.3
Avg. Pas. Len	209.8	142.1	296.8	230.8

Table 2: Dataset Statistics for the ArgGen corpus. The terms *Multi, Inst., Ans, Pas.* refers to Multilingual, Istances, Answer and Passage in the table. Eng. and Ben. refer to English and Bengali respectively.

dataset so that the model doesn't learn to reason in a singular language resulting in a bias. The statistics of the amalgamated QA dataset *Multi_QA* [1] is given in Table 1.

3.2 ArgGen Dataset

We curate the first multilingual event argument generation dataset in English and two morphologically rich Indian languages, Hindi and Bengali. The dataset consists of abstractive aggregated argument mentions for each of the six argument roles, that is, *Time, Place, Casualties, After Effects, Reason, Participant*, in three different languages. While we use the same English documents as those used in the *ArgFuse* dataset (Kar et al., 2021), we source the Hindi and Bengali documents from reputed news websites. The news articles have been crawled from different time periods (2016-2020) to have diversity in the event types of the documents. [2]

For each document, the topic or event of the document is annotated. The documents cater specifically to the disaster domain and can correspond to 32 event types at a fine grain level and 12 event types at a coarse level. For a given document in the corpus, for each of the six argument roles, the annotator was asked to compose an aggregated argument mention in his/her own words. The aggregated argument mention should consolidate all available information from the given passage and present an informative, yet precise piece of text. All argument roles may not be populated for each and every document. Such roles are then filled with an 'N.A.' value. The corpus was annotated by two linguistic experts with good knowledge about data curation and had working/native proficiency in the languages of the documents. The statistics of the dataset is presented in Table 2. While we have create a low-resource multilingual NLG dataset, we have observed that our Hindi and Bengali corpus comprise of more challenging aggregated argument mentions.

Model	Scores		
	R-L	MTR	BScr
English			
GPT-2	36.12	10.22	75.7
mT5-base	32.91	6.78	74.9
Our model	58.24	18.94	84.4
Bengali			
mT5-base	6.05	10.26	64.9
Our model	32.22	21.09	77.4
Hindi			
mT5-base	28.40	3.31	71.6
Our model	18.71	2.89	68.6
Multilingual			
mT5-base	44.03	13.53	74.7
Our model	39.75	9.85	77.6

Table 3: Document-Level Event-Argument Generation Results across languages (train and test languages are same). R-L, MTR and BScr denote ROUGE-L, METEOR Scores and BERTScore respectively as %.

4 Discussion

We have used mT5-base[3] (Xue et al., 2021) model at the core of our experiments. In Table 3, we present our event argument generation results across languages using ROUGE-L, METEOR [4] and BertScore [5]. We find that the results improve by a major margin by following our pretraining + finetuning recipe, infused with sentence-level argument information. However, given the model is trained on a large amount of English corpus, we find the best results being reported for English. We report the importance of each of the elements proposed in our framework in Table 4. We can observe that pre-training our model on reasoning data helps a lot in improving the generation capabilities of the model. Infusion of argument prompts can also be observed as a major point of guidance for the model. This highlights and justifies the necessity of our proposed pipeline framework instead of an end-to-end one.

[1] We access all the constituent datasets of Multi_QA from https://www.tensorflow.org/datasets/catalog/overview

[2] https://www.anandabazar.com/, https://epaper.bhaskar.com/

[3] https://huggingface.co/google/mt5-base

[4] https://github.com/Maluuba/nlg-eval

[5] https://github.com/Tiiiger/bert_score

BertScore	English	Bengali	Hindi	Multilingual
English	84.4	69.5	63.9	75.2
Bengali	68.3	77.4	62.7	69
Hindi	67.1	62.1	68.6	68.7
Multilingual	92.7	83.3	71.6	77.6

ROUGE-L	English	Bengali	Hindi	Multilingual
English	58.2	14.9	0.3	30.6
Bengali	31.8	32.2	1.9	24.4
Hindi	20.4	5.7	18.7	20.8
Multilingual	79.1	53.2	28.4	39.8

Figure 3: Crosslingual & Multilingual Analysis of Event Argument Generation using our model on ArgGen. The y-axis & x-axis labels correspond to the language of the training and test sets respectively where all scores are reported as %. The spectrum of values is represented with various shades, with the minimum values highlighted using peach and the maximum values highlighted using violet.

We present our results of the crosslingual and multilingual analysis in Figure 3. We analyse both at the surface level and at the contextual level using ROUGE-L and BERTScore respectively. We observe that for all the test cases, both at the surface-level as well as contextual, the model trained on the multilingual corpus performs the best. This can be regarded to the fact that the multilingual corpus with the combined, enlarged count of training samples provides the model a scope to train on additional data and learn from a variety of samples from different languages in a common embedding space. We also find that English, among all the other languages reports the best performance. We attribute this to i) the bias in training data of the core model for English compared to the other languages and ii) most of the aggregated mentions in the English corpora are of extractive nature, thus making it easier for the model to generate. The Hindi and Bengali corpus comprises of more challenging aggregated argument mentions which require advanced reasoning capabilities. We also find that Hindi reports the poorest performance compared to all the languages. We observed that the i) mT5-base model itself performs poorly when fine-tuned on the Hindi corpora, ii) our large Hindi pre-training corpora is of extractive nature. Although our Bengali pre-training corpora is also of extractive nature, the size of the data is lower by many orders compared to the Hindi corpora and hence we do not see such drastic effects. Our hypothesis is that i) It would help to pretrain on multilingual reasoning dataset of abstractive nature like DROP instead of large multilingual corpora of extractive nature, ii) for complex generation corpora like the Hindi corpora, larger and more complex models can help learn the synthesis better.

Setting	ROUGE-L	METEOR
Our model	58.24	18.94
- MRC pre-training	32.91	7.38
- argument prompts	31.62	9.56

Table 4: Ablation Study on the English corpus of ArgGen.'$-$' represents minus a particular setting. Scores have been reported as %.

5 Conclusion

We have presented *ArgGen*, a low-resource, prompt-based multilingual framework which aggregates event argument mentions at a document-level. We have also presented a fresh perspective in the domain of multilingual IE through our proposed challenging task of document-level event argument aggregation. We provide access to a novel multilingual event argument aggregation dataset which can also be leveraged for other related natural language generation tasks: https://github.com/DebanjanaKar/ArgGen. Our proposed model not only generates syntactically and semantically relevant aggregated argument mentions but demonstrates similar effectiveness in a zero-shot setting as well. In the future, we want to explore this task across more languages and documents.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Rui Feng, Jie Yuan, and Chao Zhang. 2020. Probing and fine-tuning reading comprehension models for few-shot event extraction.

Debanjana Kar, Sudeshna Sarkar, and Pawan Goyal. 2020. Event argument extraction using causal knowledge structures. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 287–296.

Debanjana Kar, Sudeshna Sarkar, and Pawan Goyal. 2021. ArgFuse: A weakly-supervised framework for document-level event argument aggregation. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 20–30, Online. Association for Computational Linguistics.

Vladimir I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Junyi Li, Tianyi Tang, Jian-Yun Nie, Ji-Rong Wen, and Xin Zhao. 2022. Learning to transfer prompts for text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3506–3518, Seattle, United States. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Andy T. Liu, Wei Xiao, Henghui Zhu, Dejiao Zhang, Shang-Wen Li, and Andrew Arnold. 2022. Qaner: Prompting question answering models for few-shot named entity recognition.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022. PromDA: Prompt-based data augmentation for low-resource NLU tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4242–4255, Dublin, Ireland. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

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.

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, Hong Kong, China. Association for Computational Linguistics.