Document Level Event Extraction By Conditional Generation

Rajeev Kashetti rkashett@gmu.edu
Jagadish Ramidi jramidi@gmu.edu
Sai Srujan Dandyala sdandyal@gmu.edu
Vamsi Krishna Peddi vpeddi3@gmu.edu

1 Introduction

Building machine reading systems, as well as downstream activities like information retrieval, knowledge base populating, and trend analysis of actual world events, all require an understanding of the text's events. Machines may be able to better comprehend the underlying storylines by extracting events from literary works. Therefore, a reliable event extraction method is essential for comprehending tales in their whole. To tackle this issue we implemented document level neural event argument extraction model. By formulating the task as conditional generation following event templates.

1.1 Task / Research Question Description

Can the model be able to extract multiple related events from a document? this is the research question we were trying to address as part of this project on (Li et al., 2021) paper. By utilizing memory (Du et al., 2022) paper achieved this task and our initial idea was on this line. We wanted to tackle the bottleneck of (Du et al., 2022) paper. As the document length and number events per documents grow the performance of (Du et al., 2022) paper tends to go down. Can we extract arguments if length of document and number of events per document is high.

1.2 Motivation and Limitations of existing work

As BART(Lewis et al., 2019) model used in (Du et al., 2022) can take input lengths of 512 tokens this creates a limitation where the model might have a chance of missing some relevant events. To overcome this shortcomings. (Du et al., 2022) has added memory component to the model. But still (Du et al., 2022) clearly effects when the distance between informative arguments and gold standard triggers substantially increase. As per the error analysis we made on subset of the dataset where informative arguments is atleast 100 words far from gold standard triggers we found that there were model could not identify most of the arguments. So we wanted to make the model get context of the texts even for lengthy documents utilizing Longformer.

1.3 Proposed Approach

Our Idea is to send whole document as a input to the model to extract arguments rather than sending 512 tokens at a time. But, Transformer based models are unable to process long sequences due to their self attention operation, which scales quadratically with the sequence length. To address this limitation, we implemented the Longformer which has an attention mechanism that scales linearly with sequence length, making it easy to process documents of thousands of tokens or longer. Longformer attention mechanism is a drop in replacement for the standard self attention for our apporach as it combines a local windowed attention with a task motivated global attention. Our Idea is that by using long former we can get context of larger documents with arguments over substantial distance from triggers. As per our error analysis on the baseline model (Du et al., 2022), the model is unable to identify informative arguments which are over 100 words apart from trigger words. We observed significant increase in the Performance on informative argument extraction after applying Longformer(Refer Table 3). We observed 7% increase in head match and 4% increase in Coref Match for Argument Identification. We also observed 6% increase in Head Match and 4% increase in Coref Match for Argument Clssification.

2 Approach

We worked on challenging task of extracting informative arguments (Name entity mentions are recognzed as more informative than nominal mentions). Each event consists of (1) Trigger expression of event type ”E” which is continous span which we get from ontology. (2) a set of arguments each of which has a role predefined in the ontology. (3) for each event type E, we will also have a template . Our goal is to extract all argument spans to fill in the role of Events [E1, E2,....En]given a long document <Trigger1>.....<Trigger2>.....<Trigger n>. i.e. Trigger 1 triggers event of type E1. Then we have to fill the arguments in the template related to E1. For example as we can see in Figure 1, when the model encounters trigger word unfilled template for the event related trigger of event type Transaction.exchangeBuySell is extracted and filled by the argument model is returned.

The event extraction consists of two sub tasks 1. Trigger extraction 2. Argument Extraction One template per event type given in the Ontology along with set of event types and roles for each event. So, no further human intervention is required.

2.1 Trigger Extraction

As mentioned above trigger has to be detected first to have input ready for our argument extraction model. Any trigger extraction model can be used. But here we used Trigger extraction model designed to work with only keyword-level supervision. For example, Damage event we use two key words damage, harm as initial supervision with no mention level annotation.

2.2 Argument Extraction Model

We use conditional generation model for argument extraction, where the condition is an unfilled template and a context. The template is a sentence that describes the event with <arg>placeholders. The generated output is a filled template where placeholders are replaced by concrete arguments. Previous work (Li et al., 2021) and (Du et al., 2022) used BART for argument extraction. As the BART Transformer consisted of an encoder-decoder architecture intended for sequence to sequence tasks((Sutskever et al., 2014)), such as summarization and translation. Compared to encoder-only(eg. (Lewis et al., 2019)) Transformers that are effective on a variety of NLP tasks, pretrained encoder decoder models have achieved strong results on tasks like summarization. Yet, such models cant efficiently scale to seq2seq tasks with longer inputs.

To facilitate modelling long sequences for seq2seq((Sutskever et al., 2014)) learning, we implemented a Longformer Variant that has both encoder and decoder Transformer stacks but instead of the full self-attention in the encoder, it uses the efficient local+global attention pattern of LongFormer. The decoder uses the full self attention to the entire encoded tokens and to previously decoded locations.

Since pre-training LED is expensive (Beltagy et al., 2020) implemented by initializing LED parameters from the BART, and follow BART’s exact architecture in terms of number of layers and hidden sizes. The only difference is that to process longer inputs, they extended position embedding to 16K tokens(up from BART’s 1K tokens) and initialized the new position embedding matrix by repeatedly copying BART’s 1K position embeddings 16 times but they mentioned further improvements should be possible through pretraining of LED.

We tried implemented two variants of pretrained Longformer encoder decoder, 1) **allenai/led-large-16384**, 2) **patrickvonplaten/led-large-16384-pubmed** which is fine tuned on allenai/led-large-16384. Later version is fine tuned upto an input length of 8k tokens. We got better results with the later one. In addition to the usual attention mask, LED can make use of an additional global attention mask defining which input tokens are attended globally and which are attended only locally.

To utilize the encoder-decoder LM for argument extraction, we can construct an input sequence of <s>template </s>document </s>. All argument names (arg1, arg2, etc.) in the template are replaced by a special place holder token <arg>. The ground truth sequence is the filled template where the placeholder token is replaced by the argument span whenever possible. In the case where there are multiple arguments for the same slot, we connect the arguments with the word ”and”. The generation probability is computed by taking the dot product betweeen the decoder output and the embeddings of tokens from the input. To prevent the model from hallucinating (Li et al., 2021) proposed we restrict the vocabulary of words to the set of tokens in the inputs and we have followed the same. The model is trained by minimizing the negative loglikelihood over all instances in the Dataset.

3 Experiments

3.1 Datasets

We are considering our sole benchmark dataset as WIKIEVENTS(Li et al., 2021). WIKIEVENTS focuses on annotations of informative arguments and for multiple events in the document level event extraction setting, and is the only benchmark dataset for this purpose. It contains real-world news articles annotated with the DARPA KAIROS ontology. The distance between informative arguments and event triggers is 10 times larger than the distance between local/uninformative arguments and event triggers. This demonstrates more needs for modelling long document context and event dependency and thus it requires a good benchmark for evaluating our proposed models. The statistics of the dataset are shown below. We use the same data split and preprocessing step as in the previous work.

	Train	Dev	Test
Documents	206	20	20
Sentences	5262	378	492
Avg. numbers of events	15.73	17.25	18.25
Avg. numbers of tokens	789.33	643.75	712.00

Table 1: Dataset Statistics

3.2 Evaluation Metrics

As for evaluation, we use the same criteria as in previous work. We consider an argument span to be correctly identified if itsoffsets match any of the gold standard informative arguments of the current event and it is correctly classified if its semantic role also matches. To judge whether the extracted argument and the gold-standard span match, since the exact match is too strict that some correct candidates are considered spurious e.g. ”the police arrived” and ”police arrived” do not match under the exact match standard. Following (Yang et al., 2021) and (Li et al., 2021), we use head word match F1. We also report performance under a more lenient metric ”Coref F1”: the extracted argument span gets full credit if it is coreferential with the gold standard arguments. The coreference links information between informative arguments across the document are given in the gold annotations.

3.3 Baselines

We compared our framework with the current state of the art model (Du et al., 2022) global neural generation based framework for document level event argument extraction which constructs a document memory store to record the contextual event information and leveraging it to implicitly and explicitly help with decoding of arguments for later events. We also compare our framework with (Li et al., 2021) which uses conditional neural text generation model for the document-level argument extraction problem, it handles each event in isolation with BART-Gen. As mentioned in previous works, neural generation based models are superior in this document-level informative argument extraction task as compared to sequence labeling based approaches. we will be comparing only with neural based models.

The neural generation-based models (BART-Gen and our framework) outperform the sequence labeling-based approaches in this challenge of document-level informative argument extraction. Additionally, generation-based techniques only need one pass as opposed to span enumeration-based methods, which need two.

With respect to the raw BART-Gen, memory-based training, which makes use of previously nearest derived event information, significantly improves precision (P) and F-1 scores. Recall scores, particularly under Coref Match, see a lesser but still significant gain.

Compared to memory based training our long-former encoder decoder gave good results as it precision recall and f1 score have improved. Particularly in argument identification which is relatively a difficult task.

3.4 Error Analysis

The Figure 2, depicts the length of words per document affects the performance of the model. Similar to (Du et al., 2022) we still see a drop in F1 score as the document length grows, but we can see that with long former our model performs better than the baselines.As we see while our framework maintains a large advantage when document is longer than 750 words, our models performance is not affected much, while the baselines performance drops.

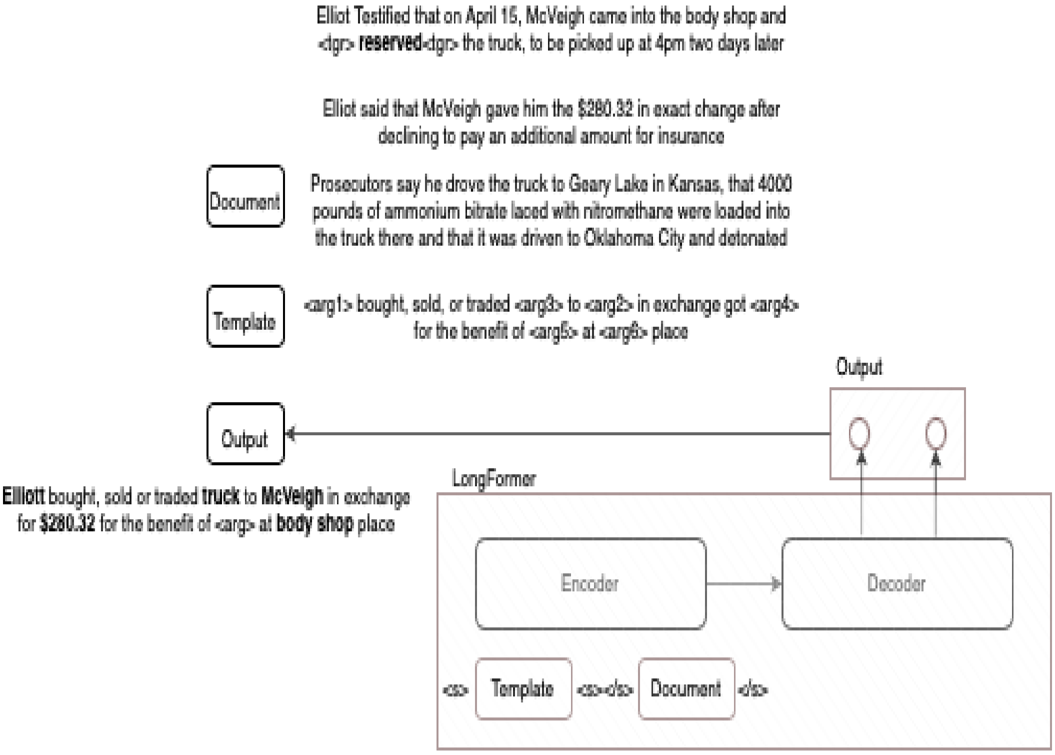
Elliot Testified that on April 15, McVeigh came into the body shop and <tgr> reserved<tgr> the truck, to be picked up at 4pm two days later
Elliot said that McVeigh gave him the $280.32 in exact change after declining to pay an additional amount for insurance
Document
Prosecutors say he drove the truck to Geary Lake in Kansas, that 4000 pounds of ammonium bitrate laced with nitromethane were loaded into the truck there and that it was driven to Oklahoma City and detonated
Template
<arg1> bought, sold, or traded <arg3> to <arg2> in exchange got <arg4> for the benefit of <arg5> at <arg6> place
Output
Output
Elliott bought, sold or traded truck to McVeigh in exchange for $280.32 for the benefit of <arg> at body shop place
LongFormer
Encoder
Decoder
<s>
Template
<s></s>
Document
</s>

Figure 1: Approach

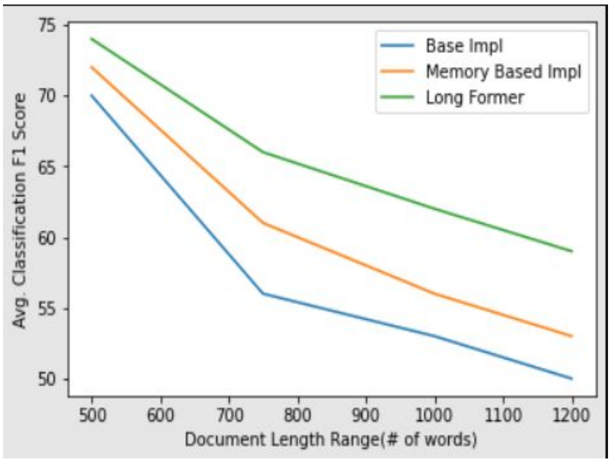

Figure 2: Results comparisoin

Looking at Table 3 In the example we can see that instead of picking up the argument value the model picks destination as the target instead of considering it as the place. To mitigate this problem, models should be able to identify certain noun phrase boundaries with external knowledge . plus, the improvement of data annotation and evaluation is also required.

3.5 Qualitative Analysis

In this the event(Table 2) triggered by **Charged**, it is hard to find arguments court and man for the memory based constrained decoding model as there was more than 100 word distance from the trigger word and the argument. But our model was able to extract the argument correctly as shown in Table 2. Similary as seen in example two we observe wierd performance model as the distance between trigger and argument increases by large.

3.6 Results

In Table 4, we present the main results for the document-level informative argument extraction. The score for argument identification is strictly higher than argument classification since it only requires span offset match. In Table 3, We have populated the results of BERT-CRF, BART-Gen and Du's paper results along with our model performance.

The results populated in Figure 3 are with while whole documents is passed input while the other three pass 512 tokens. From the table we can see that pLongformer encoder decoder model helped in significant increase in the score compared to baselines. We have utilized the hyper parameters presented in Table 5 to achieve the results.

Gold Template	Memory Enhanced Training with Constrained Decoding	Longformer Encoder Decoder
gold": " court charged or indicted men before <arg>court or judge for <arg>crime in <arg>place"	"predicted": " court charged or indicted men before <arg>court or judge for <arg>crime in <arg>place	"predicted": " <arg>charged or indicted Bilal Mohammed and Mieraili Yusufu before <arg>court or judge for <arg>crime in <arg>place",
"gold": " <arg>attacked Nicol0̆0e1s Maduro using drones and explosives at <arg>place"	"predicted": " government attacked Nicol0̆0e1s Maduro using drones at <arg>place <arg>passive <arg><arg>analysis backs claim drones were used to attack Venezuela <arg>place <arg><arg>signal <arg>"	"predicted": " <arg>attacked Nicol0̆0e1s Maduro using drones at <arg>place",

Table 2: Extracted Arguments by Memory enhanced Training with constrained decoding vs our model

Gold Template	Longformer Encoder Decoder
"gold": " attackers detonated or exploded <arg>explosive device using <arg>to attack <arg>target at place"	"predicted": " attackers detonated or exploded <arg>explosive device using <arg>to attack <arg>target at destination

Table 3: Extracted Arguments by gold vs our model

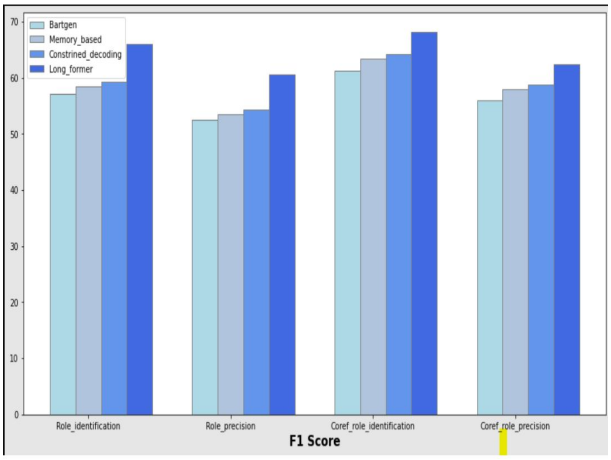

Figure 3: Results comparisoin

4 Related works

4.1 Document level event extraction

Document level event extraction can be traced back to role filling tasks that required retrieving participating entities and attribute values for specific scenario.

It is generally underexplored topic due to lack of datasets. Most of the datasets available are small in size and only covered small set of carefully selected predicates. One of the popular dataset for document level event extraction is RAMS dataset with cross sentence argument annotation, only annotates one event per document which is not so good. The other famous dataset ACE is good for sentence level argument extraction.

(Li et al., 2021) proposed a method for document level event argument extraction using conditional generation which was something new at the time. They also came uo with an bench mark wiki events dataset which has much richer event ontology specifically for argument roles. Distance between event trigger and arguments is also high in wiki events dataset when compared to other previous datasets, which helps in both local and informative argument extraction. But the approach modelled each event independently and ignored global context partially because of length limitation and lack of attention for distant context of pretrained models.

The work done to this point where either done on sentence level argument extraction or document level extraction focusing on modelling each event independently. Previous works often overlooked the consistency between extracted event structures across the long document. (Du et al., 2022) Introduced how we can leverage previously extracted events as additional context for training the text generation -based event extraction model to help the model automatically capture event dependency knowledge. To explicitly help the model satisfy the global event knowledge based constraints, also proposed a dynamic decoding process with world knowledge based argument pair constraints. But still it suffers when the distance between trigger and argument is greater than 100 words or large enough.

In our framework we are using Long Former en-

	Argument Identification						Argument Classification					
	Head Match			Coref Match			Head Match			Coref Match		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BERT-CRF	-	-	52.71	-	-	58.12	-	-	43.29	-	-	47.70
BART-Gen	58.62	55.64	57.09	62.84	59.64	61.19	54.02	51.27	52.61	57.47	54.55	55.97
Memory-based Training	61.07	56.18	58.52	66.21	60.91	63.45	55.93	51.45	53.60	60.47	55.64	57.95
w/knowledge constrained decoding	62.45	56.55	59.35	67.67	61.27	64.31	57.23	51.82	54.39	61.85	56.00	58.78
Longformer	72.90	60.43	66.08	75.27	62.39	68.23	66.88	55.44	60.62	68.82	57.04	62.38

Table 4: Performance on the informative argument extraction task

train batch size	2
eval batch size	1
learning rate	3e-5
accumulate grad batches	4
training epochs	5
warmup steps	0
weight decay	0
gpus	1

Table 5: Hyperparameters

coder decoder which helps is utilizing larger documents which helps in providing greater context to the model negating the document length limitation.

5 Conclusion and Future Work

In this we examined the effect of longformer encoder decoder on document level informative event argument extraction. In this new framework we proposed a method to bypass the document length limitation while extracting the argument. Experiments demonstrate that our approach achieves a bit improvement above prior works and large advantage when document length and number events increase. For future work we plan to investigate how to extend our method to multi document event extraction cases. We also feel that incorporating more ontological knowledge produced more accurate extractions.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Xinya Du, Sha Li, and Heng Ji. 2022. Dynamic global memory for document-level argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5264–5275.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. *arXiv preprint arXiv:2104.05919*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. 2021. Document-level event extraction via parallel prediction networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6298–6308.