Master's Essay

Cross-Document Event Argument Extraction with LLMs

Chihsheng Jin, Aaron Steven White

Department of Linguistics
University of Rochester
zjin22@ur.rochester.edu

In this paper, we extensively evaluated large language models' capability of extract event arguments with multiple document inputs. Given a document pair with an annotated event trigger and a well-documented ontology (FrameNet in our case), we tested two proprietary large language models using zero-shot and few-shot prompting and evaluated the results using CEAF-RME family of metrics. To further explore LLMs' event understanding ability, we implemented several matching algorithms to map the LLM outputs to exact spans in the documents, which led to better extraction results across all metrics. The results suggest that large language models have modest ability to understand events, but applying post-processing methods makes the outputs useful for preliminary event extraction.[1]

Contents

1 The code and data are available in our GitHub repo: https://github.com/ChihshengJ/retrieval-augmented-event-extraction

1. Introduction

Events can be seen as entities represented by a list of arguments. In a document that describes a specific event, the event can be filled with specific spans in the document corresponding to each argument in order to be evaluated linguistically. In this sense, we consider a structured representation of events a list of arguments filled by specific spans from one or more documents that contains information related to the event.

Event argument extraction (EAE) is a critical subtask of information extraction (IE) in natural language processing. Events are one of the key to human language understanding. Understanding events is a complex process with various applications in downstream tasks such as summarization, recommendation, and knowledge graph construction, etc. (Wang et al. 2023; Li et al. 2024). There are various variants that are belongs to the family of EAE tasks, with each having its unique settings, including inputs, outputs, and the models that it implements on. Among these variants, Document-Level EAE is the most relevant to our task, Cross-Document Event Argument Extraction.

1.1 Document-Level Event Argument Extraction

The research on machine event argument extraction started out in the 90s with the development of MUC (Message Understanding Conference) series of datasets, where event argument extraction was seen as a template filling tasks, which is a subtask for obtaining predicate argument structures (Grishman and Sundheim 1996). The idea of event argument extraction was first introduced as a task in 2005 when ACE dataset (Doddington et al. 2004) was first published. ACE, or Automatic Content Extraction dataset defines four tasks related to event extraction, namely, entity recognition, relation recognition, event extraction (detection and characterization), and multimodal extraction. Events were presented by the roles and the entities detected in documents, and the corpus largely consists of news reports and broadcast transcripts.

It was not until 2018 where the idea of retrieving arguments from documents was introduced to the information retrieval scene (Hamborg et al. 2018). Since some domains like news and financial reports are rich in event-related information, many studies put more focus on domain specific document-level event argument extraction (Hamborg, Breitinger, and Gipp 2019; Zheng et al. 2019).

It's worth mentioning that multiple datasets were also constructed for this specific task. SemEval-2010 (Ruppenhofer et al. 2010) is the first dataset that addresses this task in a SRL fashion. It uses both FrameNet ontology and PropBank ontology to represent the structures of events. What is more important is that it presents document-level EAE as the combination of semantic role labeling and co-reference resolution, which influenced many following studies. Another similar dataset is RAMS (Ebner et al. 2019), or Role Across Multiple Sentences. RAMS adopted the event ontology from The DARPA Active Interpretation of Disparate Alternatives (AIDA) program

which covers 139 event types in total. The data source is around 12000 news articles linked to Reddit's *r/politics* sub. What's interesting is that the authors presented the annotators with multiple sentences around the event trigger for annotating the argument spans. RAMS can be seen as the precursor of FAMuS, which is the dataset that we used in this paper.

Powerful Neural Network models allow document-level extraction to be explored more effectively. Around 2020, some new datasets for document-level event extraction were created. WikiEvents (Li, Ji, and Han 2021a) is a dataset that consists of Wikipedia event entry linked news articles. They used KAIROS ontology to represent event structurally, which is a more diversified ontology than what ACE has. Another relevant dataset is DocEE (Tong et al. 2022). The data in DocEE looks similar to the data in RAMS, and it has more than 25000 documents with 356 argument types in total. However, it was built purely on news reports, with an event ontology that is specifically catered to this domain.

From 2020 onwards, this task was generally be conceptualized into four types of problems:

1.1.1 Semantic Labeling Extraction. Since event-related information is usually related to the semantic roles, there are various systems that see EAE tasks as semantic role labeling tasks. Earliest work includes sequence labeling using recurrent neural networks (Nguyen, Cho, and Grishman 2016; Chen et al. 2018). With the emergence of powerful language models like BERT and RoBERTa, some studies also utilize the encoding from these models to push the performance even further (Xu et al. 2022; Yang et al. 2023).

The study most relevant to this paper is LOME (Xia et al. 2021), or Large Ontology Multilingual Extraction. LOME is a system designed to use the FrameNet parser as the base for event-centric information extraction. The FrameNet parser will label a pool of candidate spans as the base for the following parts of the pipeline, then, they use different kinds of models to handle entity co-reference resolution, fine-grained event-typing, and temporal relation extraction. The system can detect and extract multiple events from a document, which is different from our task, where the system needs to extract a singular event from multiple sources. Nevertheless, it was used in part of this paper, which would be discussed in Section 4.

1.1.2 Question Answering Extraction. Some studies proposed question answering based extraction. The question answering approach is a direct ramification from the 5W1H model for EE. For example, Du and Cardie (2021) proposed a framework for EE based on BERT QA and question templates. Ma et al. (2023) found that LLMs are generally inferior to fine-tuned information extraction small language models because of their over-confidence produces false-positive predictions, but by prompting them with multiple choice questions, they can solve hard labeling problems for EAE tasks. More recently, Chen et al. (2024) evaluated whether the question answering paradigm could be directly applied to LLMs and found significant gap between fine-tuned models and LLM in terms of EE tasks.

1.1.3 Span Selection Extraction. Some treat document-level EAE as a span selecting task. Zhang et al. (2020) decomposed an EAE task into two sub-

tasks, a token-pairwise dependency-parsing problem for detecting the head of an argument and a boundary classification task for expanding the head into a span. PAIE (Ma et al. 2022) is a prompt based system that used a BART model to simultaneously extract argument spans from the context. Wei et al. (2021) also proposed a framework that uses frame-aware reasoning and knowledge distillation to improve implicit event argument extraction by leveraging related arguments within an event.

1.1.4 Generation Extraction. More recently some studies used generation based approach. Li, Ji, and Han (2021b) used a seq2seq model that generates arguments by conditioning on both an unfilled template and the document context, enabling cross-sentence reasoning without relying on entity recognition or co-reference resolution, making it adaptable to new event types via zero-shot learning. Du, Li, and Ji (2022) used BART generates argument spans by filling in predefined templates corresponding to the event type. It was further enhanced by a constrained decoding stage, which applies knowledge-based rules to ensure that certain entities are not assigned conflicting roles across different events within the same document.

Moreover, some studies also adopted retrieval augmented generation in their pipeline. Ren et al. (2023) developed an EE system that first retrieves top-k semantically relevant examples based on document context or event schema, then augments the event argument extraction process by generating pseudo-demonstrations sampled from continuous event semantic regions, using a T5 encoder-decoder model to generate final predictions. Liu et al. (2024) proposed a pipeline where the model pre-loads all candidate event frames into a compressive memory and dynamically retrieves relevant information based on the input query, filters out irrelevant data, and uses this refined information to predict event argument roles.

1.2 Cross-Document Event Extraction

Cross-document event extraction was first proposed in Ji and Grishman (2008)'s paper. The idea is that since each verb can have multiple senses in a corpus, to determine the sense of an event trigger (usually a verb) within a set of documents related to the test document, we need to infer it from the related documents. Applying the same rationale to the arguments, they proposed that for a specific event, each entity only plays one role in a collection of documents related by a single event type. They also built a system based on sentence-level event extraction, which is a common approach to document-level extraction at that time.

Similar research has also been conducted extensively on ACE dataset (Ji et al. 2009; Hong et al. 2011; Yang and Mitchell 2016). However, ACE was not built for cross-document event extraction, and since even document-level event extraction is a challenging task, there are not many datasets built for this task until recently, especially since the results can be harvested from document-level EAE, most studies mainly focus on entity co-reference resolution instead of end-to-end extraction.

Most recently, there are some datasets built specifically for cross-document EAE. CLES (Gao et al. 2024) is a dataset that also utilizes the linked document structure of Wikipedia articles. Despite the sheer amount of

documents it contains, the dataset only has 9 event types, and for each event the average number of roles is less than 3, which indicates that the ontology they used (which is OmniEvent (Peng et al. 2023)) is relatively coarse.

FAMuS (Vashishtha et al. 2023), or Frame Across Multiple Sources, is a dataset that used FrameNet ontology (Baker, Fillmore, and Lowe 1998) for the annotation of structured event representations in a collection of report-source document pairs in which both document contain information about the event detected in the report document. It is unique in that FrameNet is a comprehensive ontology that provides detailed list of roles for many abstract events. So essentially it differs from the other datasets like DocEE and WikiEvents because it puts more focus on the predicate instead of the topic, which is a more linguistically informed design choice, and makes it suitable for evaluating language models event understanding on different perspective.

1.3 Key questions

Following these studies, we believe that cross-document EAE task is still an under-tapped field for LLM evaluation, and examining whether LLMs have the capability to perfect this task has substantial implications for our understanding of LLMs' event understanding abilities and extraction abilities. We have three key questions for this study:

- Do LLMs have the ability to understand events in complex documents like humans do?
- Can LLMs extract event arguments from multiple sources accurately and truthfully? If they can't, what is the cause of their errors?
- If LLMs have the abilities to understand events, as shown in many extremely hard benchmarks, can we resort to extrinsic methods to realize their potential in precise event extraction?

2. Task Definition

Since there are two tasks involved in this paper, we define them separately as follows:

- **Report-Only EAE:** Given a document D, and event trigger e which is a span from D, the pipeline would retrieve an event type E_i from an underlying ontology corresponding to e. In each E_i, there would be a set of roles: $\left\{r_1^{(i)}, r_2^{(i)}, \ldots, r_N^{(i)}\right\}$. A document would be divided into various spans in a set S. For every e_i in a D, the pipeline returns a collection of key-value pairs, each with a role $r_j^{(i)}$ from E_i and its corresponding span $s \in S$.
- **Cross-Document EAE:** Given a source document D, a report document R, and event trigger e which is a span from R, the

system should retrieve an event type E_i from an underlying ontology corresponding to e. In each E_i, there would be a set of roles: $\left\{r_1^{(i)}, r_2^{(i)}, \ldots, r_N^{(i)}\right\}$. A set S of spans from the source document D would be generated by a span-finding module. For every e_i in a D, the system returns a collection of key-value pairs, each with a role $r_j^{(i)}$ from E_i and its corresponding span $s \in S$.

3. EAE Metrics

Event argument extraction has always been a challenging task for evaluation due to the fuzzy nature of the ground truth, i.e., different systems have different standards on what should be included in the gold spans (or mentions) and how spans should be organized for each role (or entity). There are various types of metrics used in different studies on EAE tasks. At the beginning of EAE studies, MUC and ACE were the norm for this subfield, and the metrics were relatively simple. A correctly extracted argument should match the event type, offsets, and role in the event match the reference argument mention (see Ji and Grishman (2008), Hong et al. (2011), Li, Ji, and Huang (2013)). However, this kind of evaluation could only work with ACE dataset, and it does not work effectively when there are complex roles for a specific event.

Other popular metrics include Arg-I, Arg-C, MUC (Vilain et al. 1995), B^3(Bagga and Baldwin 1998), and CEAF(Luo 2005). Arg-I and Arg-C essentially treats an EAE task as the combination of an argument identification task and an argument classification task. Arg-I takes in the start and end indices of a span and the event type, while Arg-C also takes the role type to see whether the span match the role type. There is also a variant that takes in the event trigger to ensure they are evaluating the same event (Huang et al. 2023). They might seem straight forward, but they are dependent on the system structure and annotation schema. Moreover, they punish too much on misaligned boundaries and do not take multiple occurrences of the same entity into consideration.

MUC metrics evaluate co-reference resolution by measuring how well a system identifies entities and their mentions across a text. It is "entity-based", so the recall and precision are both based on whether there are common mentions in the reference entities and predicted entities. There are two major problems with MUC, it is not discriminative enough due to its "entity-based" nature, and it favors over-merged entities with wrongfully clustered mentions (Moosavi and Strube 2016). To solve these problems, Bagga and Baldwin (1998) developed B^3 metrics. B^3, unlike MUC metrics, is mention-based, so the precision and recall are based on how mentions from the reference and system predictions in each entity are aligned. B^3 is more discriminative than MUC and also eliminates MUC's preference over overly merged entities, but it also suffers from some pitfalls. It does not take into conflated mentions (co-reference) into consideration, and returns counter-intuitive results for edge cases (Luo 2005).

These two metrics inspired CEAF family of metrics, which is the family of metrics used in FAMuS. CEAF, or Constrained-entity alignment F-measure, was first introduced as a mention-based co-reference resolution metric. It measures the alignment between reference entities and predicted entities. More specifically, it takes a list of reference entities R, which includes all the mentions for each entity, a list of system predicted entities S, and an arbitrary similarity score function that takes in two entities. By using the Kuhn-Munkres algorithm (Kuhn 2010), it maximizes the total similarity between pairs entities in terms of the mentions. By adopting this bi-partite matching algorithm, it solves the co-reference conflation problem in B^3.

Since we need our results to be comparable to the results in the original FAMuS paper, we adopted CEAF as well. To make our results more explainable, the definitions of the metrics are introduced in the following passages.

Let R_m be a subset of R with size m and S_m be a subset of S with size m. to find the maximum similarity score, this problem essentially could be seen as a problem of finding the optimum match between R_m and S_m. Thus, let G_m be a set containing all the possible maps that map each element of R_m to S_m, and $\phi(R, S)$ be the similarity measure. The algorithm could be described in the following equation:

$$g^* = \underset{g \in G_m}{\operatorname{argmax}} \sum_{R \in R_m} \phi(R, g(R)) \tag{1}$$

let $\Phi(g^*)$ denote the total simiarity score of the optimum map g^*, the precision, recall, and F-1 measure can be defined as follows:

$$P = \frac{\Phi(g^*)}{\sum_i \phi(S_i, S_i)} \tag{2}$$

$$R = \frac{\Phi(g^*)}{\sum_i \phi(R_i, R_i)} \tag{3}$$

$$F1 = \frac{2PR}{P + R} \tag{4}$$

The precision measures if the system is over generating mentions. If the system output too many mentions for each role, the denominator will increase, and the overall precision will decrease. The recall measures if the system generates enough mentions for each entity. If the system neglects some of the mentions, the $\Phi(g^*)$ will decrease, lowering the recall.

CEAF-REE, used in Du, Rush, and Cardie (2021)'s paper, is the implementation for role-filler argument extraction, hence the name REE. To fit the EAE task, they changed the similarity measure $\phi(R, S)$ so that the it puts more focus on the one on one matching between system extracted spans and reference spans. In Chen et al. (2023)'s paper, they argued that said similarity measure is too harsh on models that outputs partially incorrect mentions. Thus, they replaced the measure with $\phi(R, S) = |R \cap S|$, and named it CEAF-RMEϕ_3 after Luo's paper, which would be used in the original FAMuS paper as one of the metrics. The other metric used in that paper is CEAF-RMEα, which is another variant of CEAF score in which the similarity measure was replaced with normalized editing distance. This alteration essentially loosened the standard for matching even further, allowing more leniency on the match between reference spans and predictions.

4. FAMuS and FrameNet

FAMuS, as mentioned in Section 1, is a novel dataset that tackles two tasks innate to Information Retrieval from multiple-document, namely, the source identification task and cross-document event argument extraction task. FAMuS was built on top of MegaWika (Barham et al. 2023), which is a multilingual dataset that leverages the data structure of a Wikipedia excerpt and a link in the citation of that excerpt to another article to gather a large collection of these document pairs, where each pair of documents includes some information about the same event. In other words, in each of the document pairs, the Wikipedia excerpt presents a brief description of a specific event, but the information of said event is scattered in both of the documents. To follow suit, we use *report* to denote the Wikipedia excerpt and *source* the linked article.

To structurally extract the event arguments from MegaWika instances, a robust yet comprehensive ontology is necessary to encompass the incredibly diverse events in the dataset. FrameNet is a dataset that contains more than 1,000 event types and detailed event definition, list of core roles, and example annotation of documents for each event types. FAMuS uses the event definition and event core roles from FrameNet to prompt the annotators. Meanwhile, since not all such pairs in MegaWika de facto contains information on the same event, FAMuS used LOME FrameNet parser to detect events, as we've discussed in Section 1. By oversampling document pairs with detected events, 5 examples for each of the 253 event types was ensured after the human annotation. After identifying the documents that contain information about the same event, the cross-document event argument extraction (CDAE) annotation was collected via Amazon Mechanical Turk. Despite the fact that the CDAE annotations were not validated with redundant annotations, the inter-annotator agreement is relatively high, which indicates decent annotation quality.

More specifically, FAMuS not only includes CDAE annotations, but also the detected events and entities predictions from LOME. So for each report-source document pairs, there would be the annotated spans in both documents, a list of spans for each document that represent entities, and a list of detected events. The list of entities would be crucial for the post-processing with will be discussed in section 6.

In terms of the statistics of FAMuS, it contains 253 distinct event types defined in FrameNet in total, presenting different. For each event type, there are 5 unique report-source document pairs, three of them allocated in the train split, the other two are assigned to the dev and test split.

5. Prompting LLMs

To test the state-of-the-art performance of LLMs on CDAE tasks, we used two powerful proprietary LLMs: GPT4-o and Claude 3.5 Sonnet. GPT-4o, released in May 2024, is one of OpenAI's endeavors towards multimodal, multilingual capabilities. It has exhibited performances on par with GPT-4 across many extremely complex intelligence tasks without being prohibitively expensive like the other available models in their lineup (Hurst et al. 2024). Claude 3.5 Sonnet, being Anthropic's counterpart, also shows performances on par with, if not, better than GPT-4o across every mainstream metrics used to evaluate large language models, such as MMLU, DROP, and Big-Bench Hard (Anthropic 2024). Since model comparison is not the main concern of this study, we only compared the models in this stage to choose the better one to work with.

5.1 Setting up LLMs

To prompt LLMs to extract event arguments, there are a few variables that need to be settled in order to get a more representative performance. Due to the budget limit of this study, we only conducted studies on prompting strategies and different temperature settings.

5.1.1 Generating prompts. Prompting has been one of the most important part of deploying large language models in all kinds of systems. Prompt engineering, i.e., explore different prompts to yield better results from LLMs has been proven by many cases to help with exploiting LLMs' latent capabilities (Brown et al. 2020; **?**; White et al. 2023). There are various commonly used strategies, such as few-shot prompting, Prompt chaining, Chain-of-thought prompting. Since there are signs showing that large language models' performance deteriorates over the length of inputs (Li et al. 2023; Levy, Jacoby, and Goldberg 2024; Hsieh et al. 2024), and the average number of tokens in source documents can reach 1000, we believe that few-shot prompting is the only technique that offsets the deficit brought by the increase of input length.

The process of building a prompt is trifold. We adopted a slighted modified system prompt from FAMuS to ensure a better chance of replicating the results. As mentioned in section 4, we only conducted experiments on the test split in the FAMuS dataset, and three splits share the same set of event types. Thus, we utilized the examples in the train split to fill out the few-shot prompts, which consist of three examples per prompt with regard to each event type. For each example being filled in the one prompt, we extract the event type, event definition, and a list of roles from FrameNet using `nltk`, and then the report and document slots would be filled by the example with respect to the task (report-only or cross-document). Last but not least, when using few-shot prompting, the annotations for each role form FAMuS would be used to fill out the answer slot in the demonstrative examples. The prompts used can be found in Appendix 8.2.

5.1.2 Adjusting the temperature. Temperature is a variable used in the Softmax activation function in the output layer of the model, which can be shown in the following equation:

$$Softmax(x)_i = \frac{e^{x_i}/T}{\sum_j^N e^{x_j}/T} \tag{5}$$

As the temperature (denoted by T) rises, the distribution of Softmax across all candidates would become more even, which means that more unexpected outcomes would be generated. In order words, the model becomes more rigid and deterministic when the temperature decreases. Thus, one would assume that it would be best to set temperature to 0. However, the outputs being more rigid and deterministic does not necessarily entail better performance on extraction, since the outputs are still largely dependent on the data the model was trained on. Previous works show no conclusive evidence suggesting setting temperature to 0 would be the best practice (Goel et al. 2023; Vashishtha et al. 2023; Dagdelen et al. 2024; Wei, Gautam, and Huang 2024). Thus, we conducted experiments on three different temperatures, $0, 0.4$, and 0.7 to test out the best setting for this study and build the following experiment on top of it.

5.1.3 Interim results. For this preliminary experiment, we conducted the first experiment by manipulating four variables: model, task, prompt strategy, and temperature. Table 1 and Table 2 each represents the results in two task settings (Report-Only extraction, Cross-Document extraction) from two models using the metrics introduced in section 3.

From the two tables we can see that temperature has little effects on the EAE task performance in general for both models. The differences are negligible, and with manual examination, we believe these fluctuations were caused by some of the deprecated extractions where models failed to output the predictions in the instructed format (the model answering the prompt with redundant sentences and forgot to output the extracted spans), which means they do not necessary suggest models' understanding of events are affected by temperature.

Noticeably, while few-shot prompting increases model's performance in the Report-Only setting, it is not necessarily beneficial to the cross-document EAE task for both models. This could be the result from lengthy prompts in this specific setting since the average token length of the source document is over 1000, so the three examples plus the actual prompt would add up to more than 4000 tokens in the prompt, which could lead to difficulties for model to retrieve spans correctly.

The most important trend in all the results is that the recall is significantly higher than precision, and CEAF-RMEϕ_α scores are always higher than CEAF-RMEϕ_3 scores, both indicating that the models have some degree of understanding of the events, but cannot extract the spans precisely. Thus, we adopted 0.7 as the default setting for all the other experiments in this study to investigate whether adding extrinsic methods to the pipeline can bring out the true potential of LLMs for our task.

Some other findings in this experiment also gave us insights on what to choose for the experiments ahead. First, Claude 3.5's overall performance is slightly better than GPT-4o. By examining the outputs from both models, we also observed that Claude 3.5 Sonnet tend to stick to the format given

Table 1: Performance Metrics for Report-Only

(a) Zero-Shot Prompting

Model	Temp	CEAF-RMEϕ_3			CEAF-RMEϕ_α		
		P	R	F1	P	R	F1
GPT-4o	0.0	18.00	32.55	23.18	30.59	55.32	39.40
	0.4	17.39	31.45	22.40	30.56	55.26	39.35
	0.7	17.54	31.72	22.59	31.01	56.08	39.94
Claude 3.5 Sonnet	0.0	18.99	34.34	24.46	**32.87**	**59.43**	**42.33**
	0.4	18.99	34.34	24.46	32.74	59.20	42.16
	0.7	**20.29**	**36.69**	**26.13**	32.74	59.20	42.16

(b) Few-Shot Prompting

Model	Temp	CEAF-RMEϕ_3			CEAF-RMEϕ_α		
		P	R	F1	P	R	F1
GPT-4o	0.0	21.05	38.07	27.11	31.70	57.32	40.82
	0.4	20.75	37.52	26.72	31.81	57.53	40.97
	0.7	20.75	37.52	26.72	32.28	58.37	41.57
Claude 3.5 Sonnet	0.0	22.73	41.10	29.27	33.88	61.26	43.63
	0.4	**23.11**	**41.79**	**29.76**	**34.29**	**62.01**	**44.16**
	0.7	22.58	40.83	29.08	33.34	60.29	42.94

by the prompt better than GPT-4o. On the other hand, the recall scores are significantly better than precision for all trials, and the recall scores are even higher than some of the performances shown in the original FAMuS paper. This indicates that the models have a general understanding of the events, but are unable to extract the spans precisely. The prevailing low precision scores also validate this observation. By examining the outputs closely and comparing them to the annotations, we found that models tend to extract longer spans, to which the metrics that we are using are very sensitive, since they are comparing the similarity between strings in one way or another.

Thus, all the following experiments would be done with Claude 3.5 Sonnet with the temperature set to 0.7, but the two prompt settings are still being controlled since there could be many out-of-distribution cases where event types does not the corresponding document pairs in the training set for few-shot prompting.

Table 2: Performance Metrics for Cross-Document

(a) Zero-Shot Prompting

Model	Temp	CEAF-RMEϕ_3			CEAF-RMEϕ_α		
		P	R	F1	P	R	F1
GPT-4o	0.0	12.89	19.45	15.50	28.49	42.99	34.27
	0.4	12.89	19.45	15.50	28.64	43.21	34.45
	0.7	13.58	20.48	16.33	29.37	44.30	35.32
Claude 3.5 Sonnet	0.0	**15.33**	**23.13**	**18.44**	**32.62**	**49.22**	**39.24**
	0.4	14.49	21.86	17.43	32.05	48.35	38.55
	0.7	13.88	20.94	16.70	28.74	43.36	34.57

(b) Few-Shot Prompting

Model	Temp	CEAF-RMEϕ_3			CEAF-RMEϕ_α		
		P	R	F1	P	R	F1
GPT-4o	0.0	12.81	19.33	15.41	27.80	41.94	33.44
	0.4	13.12	19.79	15.78	28.03	42.29	33.71
	0.7	11.52	17.38	13.85	26.53	40.02	31.91
Claude 3.5 Sonnet	0.0	13.58	20.48	16.33	**29.71**	**44.83**	**35.74**
	0.4	13.73	20.71	16.51	29.35	44.28	35.31
	0.7	**14.34**	**21.63**	**17.25**	29.36	44.30	35.31

6. Post-processing LLM outputs

The high precision, low recall results from the experiment on models and temperature differences leads to another question. Since LLMs generally output too much for precise EAE, and that they tend to not use the exact words in the documents, a mapping from model outputs to actual spans in the documents could be used to ameliorate this problem. To match the output spans with actual spans in the documents, the map should be a function that takes in a pool of candidate spans, a predicted span, and output one span from the document for one specific argument. If we see documents as a list of tokens, a document with m tokens would have $\frac{m(m-1)}{2}$ spans for the system to match. Thus, using algorithms to conduct efficient matching, or create a smaller pool of candidate spans is necessary for efficient EAE. In this study, we use two methods to post-process the raw outputs from LLMs to achieve better performances on EAE task. Namely, they are *Literal Match* with Smith-Waterman algorithm and *Cross-Encoder Ranking* with a cross-encoder model.

6.1 Literal Match

Literal match takes documents as a list of tokens and a predicted span as a list of tokens to align the predicted span with the most similar sublist in the list of tokens from the document and output the index of the sublist. There are many ways to achieve this, but we chose Smith-Waterman algorithm for its efficiency, and its fit for the constraints of our case. In our task, the documents can be very lengthy, but the spans can be extremely short, so algorithms that utilize token frequency are not necessary suitable for the purpose since some of the words can be in multiple spans across the entire document. Smith-Waterman algorithm (Smith and Waterman 1981), which is essentially a local sequence alignment algorithm, is a good choice for the task since it can handle the situation where one input sequence is significantly shorter than the other. And the algorithm itself is also parameterized by three parameters (Namely, the match reward, the mismatch penalty, and gap penalty) to allow fuzzy matching, so even if some tokens in the predicted span do not align with the overall span in the document, they can still be matched.

We used `difflib`'s `SequenceMatcher` in Python to calculate the similarity between two spans represented by a list of tokens. The `ratio` method was used to calculate the similarity score of two input sequences, and the similarity threshold was set to 0.5. The match reward was set to 2, the gap penalty and the mismatch penalty were both set to -1.

6.2 Semantic Match

The downside of literal match is that it does not take semantic similarity into consideration, which could be untapped by merely matching strings based on character similarity. In order to utilize the semantic information in the inputs while keeping the pipeline efficient, we need a system to not only cut down the pool of candidate spans in the document, but also contextually embed the spans and prediction spans to use better embeddings. Thus, we proposed the pipeline shown in Figure 1.

In this pipeline, a span finder model would extract preliminary candidate spans for ranking. The pool of candidate should consists of meaningful linguistic units, such as entities, phrases, and clauses. The other route is similar to the one we used in the first experiment, where we extract the event (frame) related infromation from FrameNet and prompt large language models to predict spans from each role. The last step in the pipeline is that we pair the predicted span with all the candidate spans and use a cross-encoder model trained on evaluating the semantic similarity between texts to predict the similarity within the pair. Then we use the ranking to extract the top 1 candidate to make sure that the final output comes directly from the document.

There are many pre-trained cross-encoder models available on `Hugginface`, but most of them were trained on two datasets: MS MACRO (Bajaj et al. 2016) and STSB (Cer et al. 2017). MS MACRO is a dataset curated based on Bing user questions as queries paired with human written answers, more importantly, each query also has about 10 passages retrieved by Bing that are considered containing information relevant to the answer of the query. Other than the positive examples for each query, the dataset also

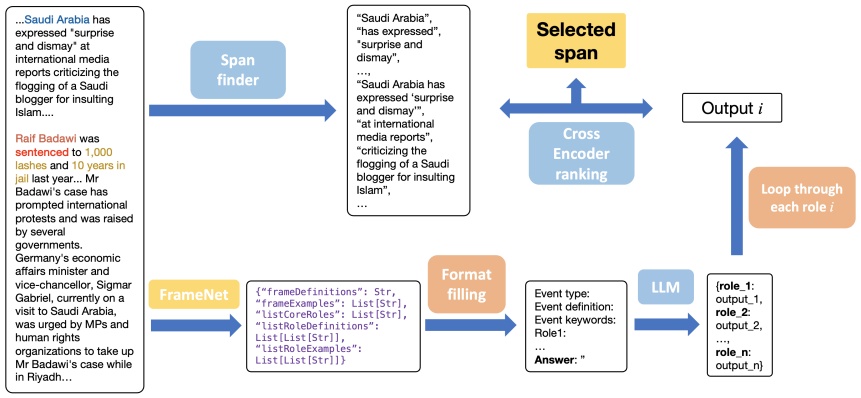

Figure 1: The pipeline used for semantic matching strategy

assigns negative examples to them, and a score from a BERT$_{CAT}$ Ensemble model that can be used for knowledge-distillation training for other models (Hofstätter et al. 2020) is also attached to the triplet.

STSB, on the other hand, is a much simpler dataset that is composed of pairs of genre-diverse short sentences with human rated similarity scores per pair. The sentences come from news headlines, and content captions. Since the form of examples in STSB is closer to our data (span pairs), we chose `stsb-distilroberta-base`, which is a distilled model based on RoBERTa (Liu et al. 2019) and trained on STSB as the cross-encoder model for the semantic matching.[2]

Cross encoder architecture is an effective design of scoring semantic similarity between a query and a passage first introduced as an implementation of BERT (Nogueira and Cho 2019). Essentially, cross-encoders are encoder-only based transformers trained on examples where the query and the passage are concatenated with separator tokens and a [CLS] head was attached to the sequence. The model with a single layer neural network which takes the [CLS] vector as the input were trained on a bi-classification task based on the passage's relevancy to the query.

2 We conducted experiments on `distilroberta-large`, `distilroberta-base`, and `MiniLM-L6-v2` trained on MS MACRO but found that `distilroberta-base` was the best one out of the three models. Since this observation was consistent, we do not report the results in this paper.

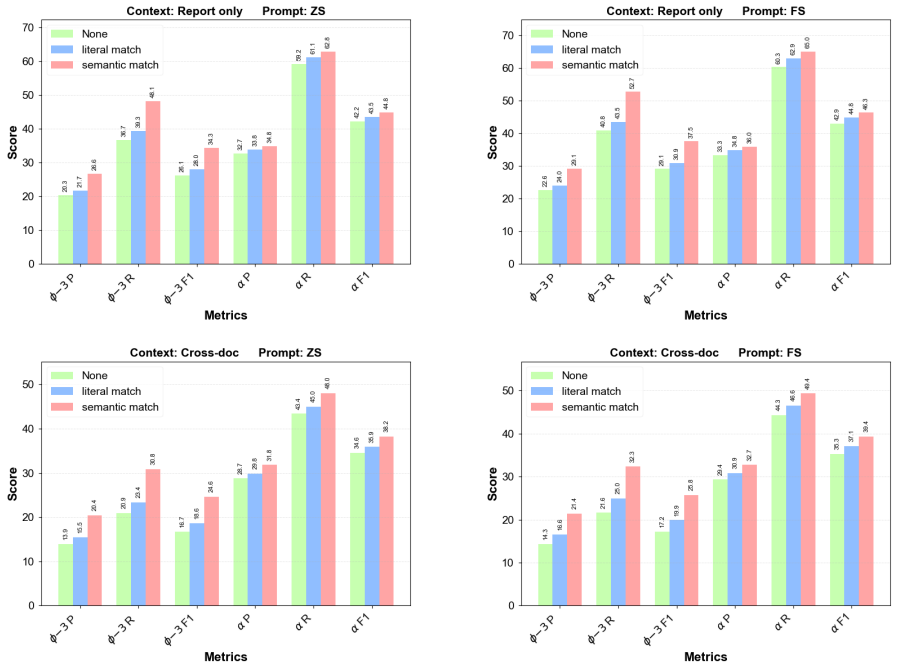

Figure 2: Model performance comparison across different matching strategies

7. Experiments

We show the results from our experiments in two scopes, the comparison between matching strategies and the comparison between our best model and the previous results from the original FAMuS paper.

7.1 Comparison across matching strategies

The results of both tasks from three post-processing strategies (including no post-processing) are shown in Figure 2.

The most significant trend in the results is that the recall of the outputs seems to be astonishingly good compared to the precision, which also led to an increase in the F1 scores. We believe that this phenomenon can be credited to LLMs' capability of understanding complex events at least at a modest level. The possible reason for low precision is that the gold annotations in FAMuS were crowdsourced on Amazon Mechanical Turk, and their instructions on the boundaries of annotated spans were not entirely clear, so it could be that the annotators did not have a consensus on this issue, which leads to the discrepancy between LLM predictions and human annotations, further lowering the precision.

On top of that, few-shot and zero-shot prompting do not make a significant difference for all three matching strategies. There is a consistent marginal gain in all metrics, but it is hard to say if few-shot prompting merely

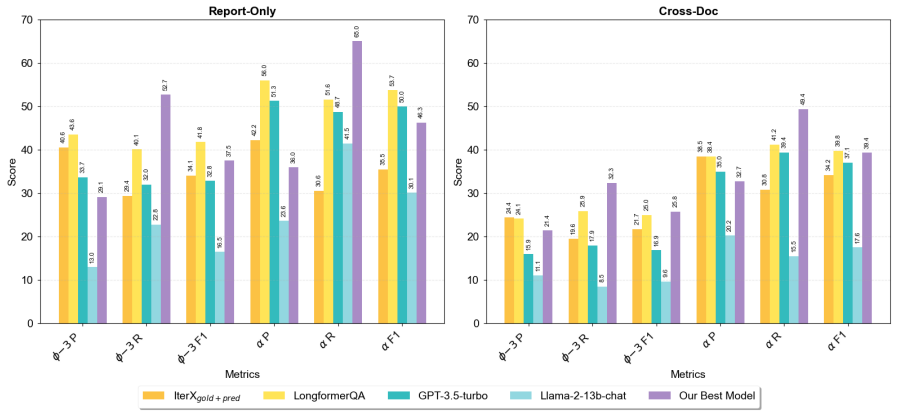

Figure 3: Model performance across our best model and previous models for FAMuS CDAE

helped LLMs to better understand the output format instead of the meaning of each role of the event, which indicates that LLMs do not necessarily need few-shot prompting to improve their event understanding.

Moreover, in comparison to other two matching strategies, semantic matching brings significant boost to all ϕ_3 scores, while the boosts in ϕ_α scores are marginal, which further suggests that LLMs outputs can be utilized in a carefully crafted pipeline, while literal match alone do not solve the issue.

Thus, we believe that LLMs do not have human-level event understanding, especially when information related to an event is scattered throughout a complex document, but it is not entirely true that they are not effective information extraction tools, and the zero-shot capability indicates that they could be useful for preliminary event extraction.

7.2 Comparison to the previous results

From Figure 2 we can clearly see that Claude 3.5 Sonnet with temperature set to 0.7 and prompted with few-shot prompting yield the best performances across the board after its results being matched based on semantic similarity through our cross-encoder model, which makes it our best model. Figure 3 shows the comparison between our best model and the previous results from the original FAMuS paper.

There are four models from the original paper that are suitable for the comparison. The first model is IterX (Chen et al. 2023), which introduced in Section 1. In the original FAMuS paper, the authors used three types of inputs to train and test IterX, based on the nature of the system, which only fills out the event role template with a pool of candidate spans. The only comparable one is the one that was trained on the gold annotation spans, spans extracted by LOME FrameNet parser, and entity spans detected by NER module in the Stanza library (Qi et al. 2020), since our cross-encoder model also select candidate spans from the pool of spans generated by the same parser. The

Longformer-QA is a Longformer model (Beltagy, Peters, and Cohan 2020) trained on question-answer pairs. The questions consisted of the name of the event (or frame in FrameNet's vocabulary), the name of a role, and the event trigger in the context, while the context being the document to be extracted. It is the best model in the original paper. Last but not least, the two LLM models were both few-shot prompted using the same method presented in this paper.

For report-only EAE tasks, our best model is even inferior to ChatGPT (GPT-3 turbo). This can be explained by the difference in the pool of candidate spans. Our best model essentially only select spans from the pool of candidate spans generated from the LOME parser, which tend to differ from how human annotate spans in the documents, while in the original paper they only used models' output directly with few-shot prompting, so it does not necessarily suggest that the LLMs we used are inferior to ChatGPT in terms of event understanding abilities. On the other hand, it's clear that for cross-document EAE tasks, our best model is on par with Longformer-QA for most metrics and even surpasses it in terms of recall scores. We believe that the LLMs strength in understanding longer context made a difference here. Combining the results in these two tasks, since the models in the original paper all have a hard time processing longer documents, it is unsure whether LLMs simply understand events better, or whether the performance should be solely credited to their ability to process longer documents better than smaller models since their context lengths are much larger.

Overall, our results show interesting potentials in LLMs event extraction abilities. Our system does not include any model fine-tuned to our dataset, yet the performance still holds up to the best model in the original paper, and even achieved the best performance for the cross-document EAE tasks. We believe that it demonstrates LLMs' capabilities of event understanding to some extent, and the importance of combining post-processing with LLM's NLP capabilities to achieve better performance.

8. Discussion and Conclusions

8.1 Summary

In conclusion, we conducted comprehensive experiments to test proprietary large language models' capability of event extraction from multiple sources. We show that temperature does not significantly impact model's performance generally and even for powerful LLMs like Claude 3.5 Sonnet and GPT-4o, accurate and truthful cross-document event extraction tasks still pose challenges. We then used two post-processing methods to test whether matching outputs from LLMs to actual spans in the documents can ameliorate the imprecision of the models when extracting information. The results suggest that using semantic match greatly improve the overall performances of LLMs, making zero-shot prompted LLMs even comparable to fine-tuned models.

However, the precision of outputs from LLMs still left space for improvement. Overall, LLMs achieve astonishing recall scores, suggesting that they can extract enough information about the event, but much of it is redundant. This phenomenon showcases that LLMs do not have human-level event extraction ability because if they did, the precision should have been much better, especially with few-shot prompting enabled, while the reality shows otherwise. Moreover, even with post-processing with match strategies, the precision scores are still lower than some of the fine-tuned small models, and LLMs even performed worse than those models in report-only tasks, which are much simpler than cross-document tasks. These findings indicate that LLMs do not have human-level event understanding yet.

Thus, our results primarily answer the three questions we raised in Section 1. Our conclusions are:

- LLMs have the ability to understand complex events in long documents where information related to a specific event is scattered across the entire document. But their ability to understand events is still significantly inferior to humans'.
- LLMs only have modest ability to extract event arguments from multiple sources accurately and truthfully. Their errors stem from hallucination, incorrect copying, and plain misunderstanding of complex events.
- Extrinsic methods such as matching LLM predictions to candidate spans in the document significantly increase the performance of EAE tasks for LLMs. However, these methods do not solve the underlying problem, that is, LLMs do not necessarily have human-level understanding of events.

8.2 Limitations and Future work

There are certain limitations to this study that we want to point out. First, since the documents are sourced from Wikipedia, the LLMs may have already trained on them in advance. Despite the fact that FAMuS came out after the two models were trained and released, which means the data used

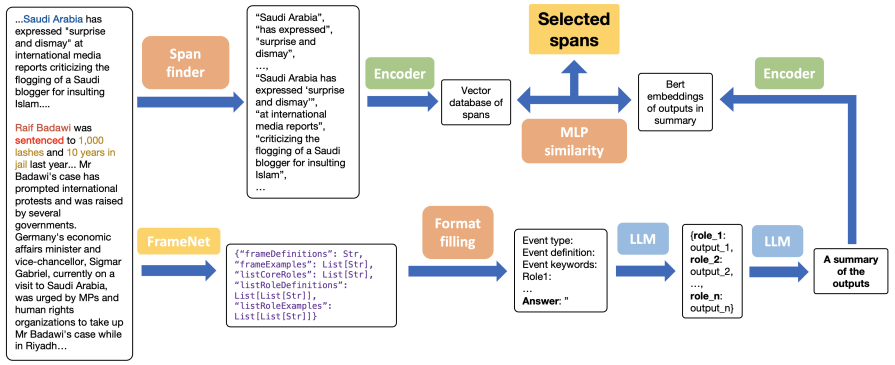

Figure 4: Pipeline scheme of **RAEE** (Retrieval-Augmented Event Extraction)

for training these large language models does not include the FAMuS dataset itself, the exposure would nevertheless pose questions on the rigidity of the study in terms of whether it can evaluate large language models event understanding ability. However, since the LLMs exhibited stellar performance on supposedly much harder benchmarks such as Big-Bench and MMLU, the mediocre results from EAE tasks in our study shows that even when the models were exposed to the documents during training, they still can not have a clear and truthful understanding of the complex events presented in those documents. So the exposure does not necessarily pose challenges to our conclusions.

Another limitation is that we did not train our own cross-encoder. In fact, we have tried using direct outputs to fine-tune a cross-encoder ourselves so that we can further refine the LLM predictions with better semantic matching precision. The paradigm can be shown in Figure 4 where we trained a Longformer on a bi-classification task with an MLP module on direct LLM outputs and human annotations[3]. So essentially, we treated direct LLM outputs as if they had the correct event argument information but with minor alteration in the wording. However, after trying out many hyperparameters in the MLP module, we failed to train a single model that have better than guessing results in this bi-classification task, proving that our premise of using LLM outputs as spans that contain the "right message" is false. We believe that if we have the budget to annotate on the LLM outputs to train the model, the performance would be better than the out-of-shelf cross-encoder model, but since the pipeline is dependent on three models, namely, the span-finding model, the LLM, and the cross-encoder, we can not be sure which one would turn out to be the bottleneck of the pipeline. Nevertheless, there is still room for improvement in terms of cross-encoder model fine-tuning to fit the task constraints better.

3 The code can be found in our Github repo.

Acknowledgments
This work was supported by the on-going research project related to FAMuS and SEAMuS. The conclusions and views of this work should only be interpreted as those of the authors. This work is heavily inspired by SEAMuS and benefited from the discussions the authors had with FACTsLab's members at the University of Rochester.

Appendix A: Prompts

The system prompt that we used is largely the same as the one used in the original FAMuS paper:

```
You are a system that generates high quality role annotations of one
    or two documents describing an event, based on given event roles.
The following inputs are given to you:
1. Event Type: A Frame name from the FrameNet ontology (eg: Hiring,
    Arrest, etc.)
2. Event Definition: Definition of the event type along with an
    optional example.
3. Event keywords: A span in the document that pinpoint the event.
    There could be no keywords.
4. Roles: All roles (or participants) of the event type (or frame)
    followed with its definition.
5. Report: A document that provides a description of the event,
    usually shorter than the Source.
6. Source: (Optional) A document that potentially provides more
    detail to the event presented in the Report.
You should output the exact text spans from either the Report (if
    Source is None) or the Source (if both are present) for each
    role in the order listed in the "roles" section.
Note that if a Source is present (not None), please only extract
    roles from the Source with regard to the event described in the
    Report.
If there are multiple candidates for one role, return the most
    informative one, i.e. "March 1st" is more informative than
    "Thursday", "Michael Jackson" is more informative than "the pop
    star".
Then, use the exact spans you predicted that are not N/A to generate
    a 1 - 2 sentence summary describing the event.
If there are multiple candidates for a span, please choose the most
    informative one.
Please answer with the following format (the Role1 and Role2 are
    placeholders):
"Role1: span1
Role2: span2
...
Summary: 1-2 sentence summary of the event"
Note that you can leave an N/A if you are not certain whether there
    is any span representing the role in the document.
```

The templates we used for prompting are also similar to the last section of the system prompt:

```
FEW_SHOT_PREFIX = """
Event type: {event_type},
Event definition: {event_definition}
Event keywords: {event_trigger}
Roles:
{roles}
Report: {report}
Source: {source},
Answer: {answers}
"""
```

```
PROMPT_TEMPLATE = """
Event type: {event_type},
Event definition: {event_definition}
Event keywords: {event_trigger}
Roles:
{roles}
Report: {report}
Source: {source},
Answer:
"""
```

The event type, event definition, and the list of roles (including their definitions) would be retrieved from FrameNet using `nltk`'s FrameNet dataset, and the event trigger, the report and the source (and answers if we were using few-shot prompting) would be retrieved from the FAMuS dataset. For Report-Only taksks, the source would be filled with `None`.

References

Anthropic. 2024. Claude 3.5 sonnet model card addendum.

Bagga, Amit and Breck Baldwin. 1998. Algorithms for scoring coreference chains.

Bajaj, Payal, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Baker, Collin F., Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Barham, Samuel, Orion Weller, Michelle Yuan, Kenton Murray, Mahsa Yarmohammadi, Zhengping Jiang, Siddharth Vashishtha, Alexander Martin, Anqi Liu, Aaron Steven White, Jordan Boyd-Graber, and Benjamin Van Durme. 2023. Megawika: Millions of reports and their sources across 50 diverse languages.

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

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Cer, Daniel, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Chen, Ruirui, Chengwei Qin, Weifeng Jiang, and Dongkyu Choi. 2024. Is a Large Language Model a Good Annotator for Event Extraction? *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17772–17780.

Chen, Yubo, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1267–1276, Association for Computational Linguistics, Brussels, Belgium.

Chen, Yunmo, William Gantt, Weiwei Gu, Tongfei Chen, Aaron White, and Benjamin Van Durme. 2023. Iterative document-level information extraction via imitation learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1858–1874, Association for Computational Linguistics, Dubrovnik, Croatia.

Dagdelen, John, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418.

Doddington, George, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, European Language Resources Association (ELRA), Lisbon, Portugal.

Du, Xinya and Claire Cardie. 2021. Event Extraction by Answering (Almost) Natural Questions. ArXiv:2004.13625.

Du, Xinya, Sha Li, and Heng Ji. 2022. Dynamic Global Memory for Document-level Argument Extraction. ArXiv:2209.08679.

Du, Xinya, Alexander Rush, and Claire Cardie. 2021. GRIT: Generative role-filler transformers for document-level event entity extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages

634–644, Association for Computational Linguistics, Online.
Ebner, Seth, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2019. Multi-sentence argument linking. *arXiv preprint arXiv:1911.03766*.
Gao, Qiang, Zixiang Meng, Bobo Li, Jun Zhou, Fei Li, Chong Teng, and Donghong Ji. 2024. Harvesting events from multiple sources: Towards a cross-document event extraction paradigm. *arXiv preprint arXiv:2406.16021*.
Goel, Akshay, Almog Gueta, Omry Gilon, Chang Liu, Sofia Erell, Lan Huong Nguyen, Xiaohong Hao, Bolous Jaber, Shashir Reddy, Rupesh Kartha, Jean Steiner, Itay Laish, and Amir Feder. 2023. Llms accelerate annotation for medical information extraction. In *Proceedings of the 3rd Machine Learning for Health Symposium*, volume 225 of *Proceedings of Machine Learning Research*, pages 82–100, PMLR.
Grishman, Ralph and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.
Hamborg, Felix, Corinna Breitinger, and Bela Gipp. 2019. Giveme5W1H: A Universal System for Extracting Main Events from News Articles. ArXiv:1909.02766.
Hamborg, Felix, Soeren Lachnit, Moritz Schubotz, Thomas Hepp, and Bela Gipp. 2018. Giveme5W: Main Event Retrieval from News Articles by Extraction of the Five Journalistic W Questions. In Gobinda Chowdhury, Julie McLeod, Val Gillet, and Peter Willett, editors, *Transforming Digital Worlds*, volume 10766. Springer International Publishing, Cham, pages 356–366. Series Title: Lecture Notes in Computer Science.
Hofstätter, Sebastian, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. *arXiv preprint arXiv:2010.02666*.
Hong, Yu, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1127–1136, Association for Computational Linguistics, Portland, Oregon, USA.
Hsieh, Cheng-Ping, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.
Huang, Kuan-Hao, I Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Premkumar Natarajan, Kai-Wei Chang, Nanyun Peng, Heng Ji, et al. 2023. Textee: Benchmark, reevaluation, reflections, and future challenges in event extraction. *arXiv preprint arXiv:2311.09562*.
Hurst, Aaron, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
Ji, Heng and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Association for Computational Linguistics, Columbus, Ohio.
Ji, Heng, Ralph Grishman, Zheng Chen, and Prashant Gupta. 2009. Cross-document event extraction and tracking: Task, evaluation, techniques and challenges. In *Proceedings of the international conference RANLP-2009*, pages 166–172.
Kuhn, Harold W. 2010. *The Hungarian Method for the Assignment Problem*. Springer Berlin Heidelberg, Berlin, Heidelberg.
Levy, Mosh, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models.
Li, Jiaqi, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2023. Loogle: Can long-context language models understand long contexts? *ArXiv*, abs/2311.04939.

Li, Qi, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Association for Computational Linguistics, Sofia, Bulgaria.

Li, Qian, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, and Philip S. Yu. 2024. A Survey on Deep Learning Event Extraction: Approaches and Applications. *IEEE Transactions on Neural Networks and Learning Systems*, 35(5):6301–6321. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

Li, Sha, Heng Ji, and Jiawei Han. 2021a. 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, Association for Computational Linguistics, Online.

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

Liu, Wanlong, Li Zhou, Dingyi Zeng, Yichen Xiao, Shaohuan Cheng, Chen Zhang, Grandee Lee, Malu Zhang, and Wenyu Chen. 2024. Beyond Single-Event Extraction: Towards Efficient Document-Level Multi-Event Argument Extraction. ArXiv:2405.01884 version: 1.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Luo, Xiaoqiang. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Association for Computational Linguistics, Vancouver, British Columbia, Canada.

Ma, Yubo, Yixin Cao, YongChing Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! *arXiv preprint arXiv:2303.08559*.

Ma, Yubo, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for Extraction? PAIE: Prompting Argument Interaction for Event Argument Extraction. ArXiv:2202.12109.

Moosavi, Nafise Sadat and Michael Strube. 2016. Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In *Proceedings of the 54th annual meeting of the association for computational linguistics*, volume 1, pages 632–642, Association for Computational Linguistics.

Nguyen, Thien Huu, 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, Association for Computational Linguistics, San Diego, California.

Nogueira, Rodrigo and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Peng, Hao, Xiaozhi Wang, Feng Yao, Kaisheng Zeng, Lei Hou, Juanzi Li, Zhiyuan Liu, and Weixing Shen. 2023. The devil is in the details: On the pitfalls of event extraction evaluation. *arXiv preprint arXiv:2306.06918*.

Qi, Peng, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Ren, Yubing, Yanan Cao, Ping Guo, Fang Fang, Wei Ma, and Zheng Lin. 2023. Retrieve-and-Sample: Document-level Event Argument Extraction via Hybrid Retrieval Augmentation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–306,

Association for Computational Linguistics, Toronto, Canada.
Ruppenhofer, Josef, Caroline Sporleder, Roser Morante, Collin Baker, and Martha Palmer. 2010. SemEval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 45–50, Association for Computational Linguistics, Uppsala, Sweden.
Smith, T.F. and M.S. Waterman. 1981. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197.
Tong, MeiHan, Bin Xu, Shuai Wang, Meihuan Han, Yixin Cao, Jiangqi Zhu, Siyu Chen, Lei Hou, and Juanzi Li. 2022. DocEE: A large-scale and fine-grained benchmark for document-level event extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3970–3982, Association for Computational Linguistics, Seattle, United States.
Vashishtha, Siddharth, Alexander Martin, William Gantt, Benjamin Van Durme, and Aaron Steven White. 2023. FAMuS: Frames Across Multiple Sources. ArXiv:2311.05601.
Vilain, Marc, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.
Wang, Xiaozhi, Hao Peng, Yong Guan, Kaisheng Zeng, Jianhui Chen, Lei Hou, Xu Han, Yankai Lin, Zhiyuan Liu, Ruobing Xie, et al. 2023. Maven-arg: Completing the puzzle of all-in-one event understanding dataset with event argument annotation. *arXiv preprint arXiv:2311.09105*.
Wei, Kaiwen, Xian Sun, Zequn Zhang, Jingyuan Zhang, Guo Zhi, and Li Jin. 2021. Trigger is Not Sufficient: Exploiting Frame-aware Knowledge for Implicit Event Argument Extraction. 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 4672–4682, Association for Computational Linguistics, Online.
Wei, Kangda, Aayush Gautam, and Ruihong Huang. 2024. Are llms good annotators for discourse-level event relation extraction? *arXiv preprint arXiv:2407.19568*.
White, Jules, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.
Xia, Patrick, Guanghui Qin, Siddharth Vashishtha, Yunmo Chen, Tongfei Chen, Chandler May, Craig Harman, Kyle Rawlins, Aaron Steven White, and Benjamin Van Durme. 2021. LOME: Large Ontology Multilingual Extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 149–159, Association for Computational Linguistics, Online.
Xu, Runxin, Peiyi Wang, Tianyu Liu, Shuang Zeng, Baobao Chang, and Zhifang Sui. 2022. A two-stream amr-enhanced model for document-level event argument extraction. *arXiv preprint arXiv:2205.00241*.
Yang, Bishan and Tom Mitchell. 2016. Joint extraction of events and entities within a document context. *arXiv preprint arXiv:1609.03632*.
Yang, Yuqing, Qipeng Guo, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2023. An amr-based link prediction approach for document-level event argument extraction. *arXiv preprint arXiv:2305.19162*.
Zhang, Yunyan, Guangluan Xu, Yang Wang, Daoyu Lin, Feng Li, Chenglong Wu, Jingyuan Zhang, and Tinglei Huang. 2020. A Question Answering-Based Framework for One-Step Event Argument Extraction. *IEEE Access*, 8:65420–65431. Conference Name: IEEE Access.

Zheng, Shun, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2edag: An end-to-end document-level framework for chinese financial event extraction. *arXiv preprint arXiv:1904.07535*.