QAEvent: Event Extraction as Question-Answer Pairs Generation

Anonymous ACL submission

Abstract

We propose a novel representation of document-level events as question and answer pairs (QAEvent). Under this paradigm: (1) questions themselves can define argument roles without the need for predefined schemas, which will cover a comprehensive list of event arguments from the document; (2) it allows for more scalable and faster annotations from crowdworkers without linguistic expertise. Based on our new paradigm, we collect a novel and wide-coverage dataset. Our examinations show that annotations with the QA representations produce high-quality data for document-level event extraction, both in terms of human agreement level and high coverage of roles comparing to the pre-defined schema. We present and compare representative approaches for generating event question answer pairs on our benchmark.

1 Introduction

Event extraction (EE) is a challenging yet important task in information extraction research (Sundheim, 1992). The task aims at extracting event information from unstructured texts into a structured form, which mostly describes attributes such as "who", "when", "where", and "what" of real-world events that happened (Li et al., 2022). The task involves extracting the trigger (predicate) for an event and identify its arguments for certain role from a sentence or a document (Li et al., 2013; Nguyen et al., 2016; Du and Cardie, 2020; Du and Ji, 2022).

However, highly skilled and trained annotators with linguistic expertise are required for labeling the event structures in the document (Doddington et al., 2004; Li et al., 2021), especially for domain-specific documents. Plus, for each new domain, schema-induction and curation require even more efforts (Du et al., 2022). It involves determining a fixed and limited set of argument roles for each event type, which takes a significant amount of

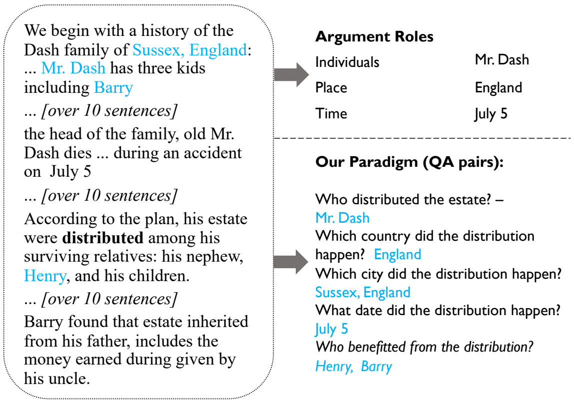


Figure 1: Extracting event structures from long documents according to the close schema (upper) vs. our paradigm of generating QA pairs (bottom). The event is triggered by **distributed** in this example.

efforts. Usually the definition of argument roles is ambiguous and causing challenges in the annotations and relatively low agreements (Linguistic Data Consortium, 2005).

Motivated by all these, we propose a new method based on annotating more complete representations of the event structures, where arguments of an event trigger might spread across the entire document. It can be easily done by non-experts. More specifically, we propose question-answer pair representation for events (QAEvent). It represents each event trigger-argument structure of a document as a set of question-answer pairs. For example in Figure 1, we can ask questions regarding the event triggered by "distribution", such as "who benefited from the distribution", and whose answer consists of one or multiple phrase spans in the document (e.g. "Henry" and "Barry"). Enumerating all such QA pairs help obtain a comprehensive set of attributes of the specific event. Our paradigm QAEventprovides several benefits, (1) it does not rely on and limited to a pre-defined set of argument roles, non is there any requirement for curated schema as in previous work; Nonetheless, the

QA-based arguments still cover almost all schema-based arguments; (2) annotated QA pairs under this paradigm can capture more nuanced/implicit attributes such as "why" and "how", instead of only general roles such as in FrameNet (Baker et al., 1998; Liu et al., 2019). (3) the annotation process is layman-friendly and cost-efficient, especially under the document-level setting. The resulting QA pairs are of relatively good quality – with high agreement scores among annotators. Also they can be easily examined and modified by data collectors.

We present an approach for collecting comprehensive/high quality event QA pairs in an efficient and scalable way. We crowdsourced question answer pairs annotators (e.g. STEM students) without linguistic background. For each event (represented by one trigger), we ask the annotator to ask questions about as many attributes as possible of the event. The requirement is that (1) the answer should be a phrase (i.e. a span) is the document; (2) follow a general template which is designed to speed up and increase mutual agreement. Through our QAEVENT paradigm and annotation strategy, we quickly obtain QA pairs set with high coverage and quality. Plus, the time cost is much smaller as compared to previous work (Li et al., 2021), especially consider our document-level extraction setting. We elaborate on the crowdsourcing and the quality control process, next we conduct comprehensive analysis of the dataset collected.

Finally, we benchmark different models on our dataset. We first propose an information extraction (IE) pipeline and template-based question generation method; Further, we also benchmark the large language model (LLMs) performance on this complex task which requires document global understanding and instruction following. Finally introduce a multi-step prompting-based framework including QA pair over generation and self-examination for refinement. During the refinement, QA pairs that are not consistent or not following the template are filtered out. Through thorough experiments, we demonstrate the advantages of our approach in terms of both consistency and performance.

2 Related Work on Semantic QA Approaches

Using QA structures to represent semantic propositions has been proposed as a way to generate "soft" annotations, where the resulting representation is formulated using natural language, which is shown to be more intuitive for untrained annotators (He et al., 2015). This allows much faster and more large-scale annotation processes (FitzGerald et al., 2018) and when used in a more controlled crowdsourcing setup can produce high-coverage quality annotations for *sentence-level* tasks(Roit et al., 2020; Pyatkin et al., 2020). Both QASRL (He et al., 2015) and QAMR (Michael et al., 2018) collect a set of QA pairs, each representing a single proposition, for a sentence. In QASRL the main target is a predicate, which is emphasized by replacing all content words in the question besides the predicate with a placeholder. The answer constitutes a span of the sentence. The annotation process itself for QASRL is very controlled, by suggesting questions created with a finite-state automaton. QAMR, on the other hand, allows us to freely ask all kinds of questions about all types of content words in a sentence. In our QAEVENT work, we introduce a new paradigm based on the QA representation of *document-level* events to achieve high coverage of event arguments, which is the first work in the information extraction community.

3 Dataset Collection

We describe our annotation process in detail, and discuss agreement between our QAEVENT annotations and the corresponding standard event extraction annotations in WikiEvent (Li et al., 2021).

3.1 Annotation Design

We annotate the event structures with question answering pairs in the document. Each event structure is represented by one trigger word. Trigger words for the events are a set of words which most accurately describe the occurrence of the events. These trigger words correspond to one event type as listed in the schema of WikiEvent (Li et al., 2021). For example, the word "distributed" triggers the DISTRIBUTION event in Figure 1.

Given a document d and set of triggers $T = \{t_1, ...t_i\}$, the annotators write a set of wh-questions that contain one of the triggers t_i whose answer is a continuous span in d. Furthermore, we also ensure that there shall not include any inference question, i.e. the questions should not require multi-hop or logical reasoning. To speed up annotation and increase agreement between annotators, we used the question template as suggested in (He et al., 2015). This template restraints the question

Document	Argument Role	Questions	Answers
(1) She offers compelling, if circumstantial, indications that Iraqi operatives helped to plot, prepare and execute murderous **attacks** in Oklahoma City (and perhaps against other targets in the United States) [...]	PLACE ATTACKER	(a) Where were the attacks carried out? (b) Who helped to plot, prepare and execute the attacks?	Oklahoma City Iraqi operatives
(2) Maduro has **jailed** and sidelined many opposition activists, regularly accusing them of plotting to overthrow him [...]	DETAINEE JAILER	(a) Who has been jailed? (b) Why were they jailed? (c) Who jailed them?	opposition activists plotting to overthrow Maduro Maduro
(3) In a country where 98% of **crime** goes unpunished, government sleuths resolve this kind of case in a matter of hours [...]	PLACE	(a) Which country has 98% of crime go unpunished? (b) Which crimes are solved quickly? (c) What percent of crime goes unpunished in the country?	Venezuela alleged assassination 98
(4) Pérez was **killed** in a shootout six months later[...]		(a) When did the shootout with Oscar Perez happen? (b) Where did the shootout with Oscar Perez happen?	six months later Caracas
(5) Ms. Davis has also found witnesses who say McVeigh and his convicted co-conspirator, Terry Nichols, had **consorted** with former Iraqi soldiers [...]	PARTICIPANT ARTIFACT	(a) Who consorted with former Iraqi soldiers? (b) With whom did the former Iraqi soldiers consort?	McVeigh and his convicted co-conspirator, Terry Nichols a Palestinian
(6) Venezuela's president, Nicolás Maduro, has survived an apparent and – if true – audacious assassination attempt when, according to official reports, drones loaded with explosives flew towards the president while he was **speaking** at a military parade in Caracas [...]	COMMUNICATOR PLACE	(a) Who was speaking when the assassination attempt occurred? (b) Where was the president speaking?	the president, Nicols Maduro at a military parade in Caracas
(7) In each of these cases, there is reason to believe that Saddam Hussein and his minions played some role in the **murder** of Americans [...]	TARGET ATTACKER	(a) Who was murdered? (b) Who is accused of playing a role in the murder?	Americans Saddam Hussein and his minions
(8) He will use it to concentrate power, whoever did this David Smilde Fire fighters **interviewed** by the Associated Press claimed that the bangs heard were caused by a gas tank explosion in a nearby apartment [...]	PARTICIPANT PLACE PARTICIPANT	(a) Who was interviewed? (b) Where did the explosion occur? (c) Who interviewed the firefighters? (d) Who backed up the firefighters?	Firefighters in a nearby apartment Associated Press Local Press

Table 1: Examples of question answer pairs capturing various WikiEvent argument roles, which are annotated with based on the highlighted trigger word and the document. QAEVENT align well with the schema, and meanwhile capture more comprehensive aspects of event arguments.

q to a format with seven tokens where $q \in$ **WH** $\times$ **AUX** $\times$ **SBJ** $\times$ **TRG** $\times$ **OBJ1** $\times$ **PP** $\times$ **OBJ2**, where **WH** token is the question word which can be from *Who, Whom, What, When, Where, Why, How*; **SBJ** refers to the entity that performs the action; **OBJ1** and **OBJ2** are the entities that are being acted upon. We also use **PP** to show direction, time, place, location, spatial relationships, or to introduce an object. Apart from the **WH** and **TRG** not every field must be included. Based on our preliminary study, the template is sufficient to cover most of the event argument questions (>90%).

Questions can have multiple answer spans. Overall, one example question is "What was Mr. Dash expected to have ?" with the answer being "kindness, confidence".

3.2 Data Preparation and Annotation

We annotate a total of 154 documents which comprise of many different events from the WikiEvent-Dataset (Li et al., 2021). We followed their Train, Dev and Test Splits. Each document contains a set of triggers for which annotators wrote a set of question and answers. The statistics for the final dataset is shown in Table 2.

3.3 Annotation Process

We set up a crowd sourcing job on Amazon Mechanical Turk to obtain QA pairs. In order to help the annotators, we provide some bootstrap QA pairs generated using GPT-4 which is used in many downstream NLP tasks (Liu et al., 2023). Though GPT-4 questions are prone to many problems such as low coverage and inaccuracy, it acts as a good reference point to the annotators. Figure 6 in Appendix shows the Amazon Mechanical Turk interface which we used to collect the QA pairs. It can be seen that we have a set of triggers T and questions are created by following the template for each of the triggers (highlighted).

Datasplit	Documents	Sentences	Event (triggers)	QA pairs (arguments)
Train	130	3586	1319	2117
Validation	12	320	199	223
Test	12	251	110	132
Overall	154	4157	1628	2472

Table 2: Summary of Data Statistics. QA pairs are annotated by our annotators.

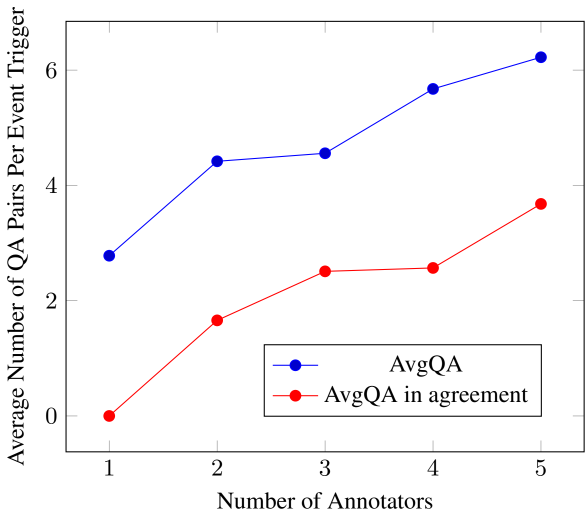


Figure 2: Inter-annotator agreement on five documents containing 50 events. A QA pair is considered agreed if it's written by two or more annotators.

After reading the annotation guideline (Figure 5), the annotators were asked to complete a Qualification Test (five documents) as a part of the screening process. The results were then reviewed by the authors before they start to annotate all the documents. Finally, we recruited five annotators who are native speakers with at least a high school degree. We record the timings to find out the average time required to annotate the document with a series of Questions and Answers based on triggers. It takes an average of 16 minutes 22 seconds for annotating each document (with a maximum being around 20 minutes and a minimum of around 10 minutes). This difference in time, accounts for the variety of documents, with different length, complexity, number of events and topics. Compared to WikiEvent, annotation under their paradigm is much more costly (around 30 minutes per document), which demonstrate the benefits of our QAG paradigm.

3.4 Inter-Annotator Agreement

To judge the reliability of the data, we calculate inter-annotator agreement on a subset of the annotated dataset of five documents. Five annotators write the question answer pairs after passing the qualification test. This calculation becomes more difficult since a particular question for an event trigger can be phrased in many ways. On the other hand, the answer spans generally remain highly overlapping for a particular type of question. For example, for a trigger word *custody* one annotator asks the question *"Who remains in custody?"* while another annotator asks the question *"Who is in custody?"*, however, the answer span coincides heavily.

To calculate the agreement, for each event we consider two QA pairs (arguments) to be same if they have the same Wh-word and have an overlapping answer span. A QA pair is considered to be agreed upon if at least two annotators agree on the pair (He et al., 2015). We calculate the average number of QA pairs per trigger t_i and also kept a track of average number of QA pairs agreed. Figure 2 shows how the average number of QA pairs and agreed QA pairs increases as the number of annotators increases. It shows that after five annotators the number starts to asymptote. We also find that one annotator finds around 60% of agreed QA pair that are found by five annotators. This implies that a high recall can be achieved and if we want to improve the process further. In future, we can have annotators answer others questions instead of making their own pairs.

4 Dataset Analysis

In this section, we show that QAEvent has high coverage of event arguments and uses a rich vocabulary to label fine-grained and nuanced event attributes.

4.1 Compare the QAEvent Coverage of Event Arguments with WikiEvent

The recall and heatmap, together, imply that annotations made by crowdsourcing can contain much of the information made by experts and are easily understandable too.

Table 1 shows the comparisons between examples from QAEvent and original fixed schema WikiEvent examples (Li et al., 2021). Our annotation mechanism captures different information from WikiEvent schema, however, we can find a lot of similarity between the two. To measure this, we try to find the overlap between the answers in our generated QA pair arguments, and the WikiEvent

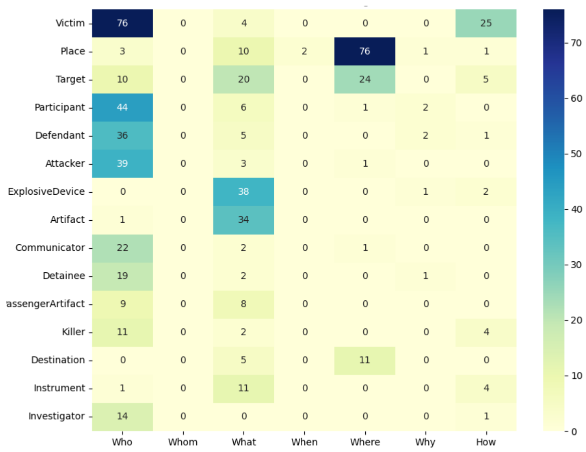


Figure 3: Co-occurrence of Wh-word in QAEVENT annotations and WikiEvent argument.

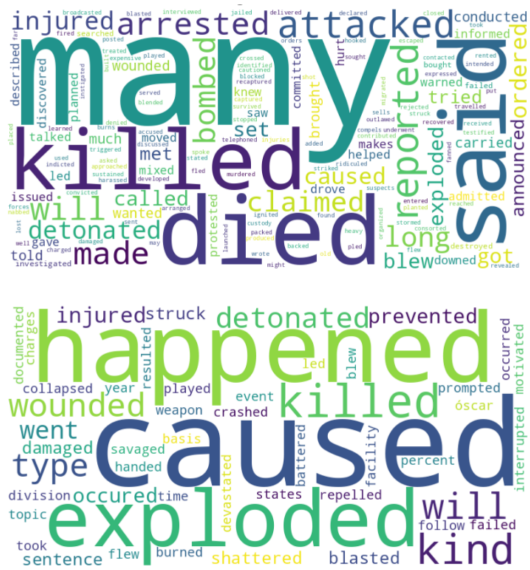


Figure 4: Words which appear after Wh-word. Upper word cloud shows the words that appear after Who, Whom & How; bottom shows the words that appear after What, When, Where & Why.

arguments provided.

We consider the matches if the WikiEvent argument overlaps with the answer span. An argument is considered to be overlapping if any of the word in the argument appears in the answer span. This is also supported by the fact that our guidelines ask the annotators to select answer spans from the document. We calculate the precision as a proportion of QA pairs that match a WikiEvent argument. The recall is calculated as the proportion of WikiEvent arguments which are covered by the QA pairs. The precision is **51.62%**, the recall is **78.01%**, and the F1 is **62.13%**. A loss in recall is observed due to some erroneous inputs by the annotators. The annotators also tend to skip some of the triggers are highly overlapping. For example if the trigger word attack comes twice in the sentence in two different form, the annotator skips one of the triggers. This is not necessarily a bad thing as this opens scope of study on optimizing the number of triggers to form an ideal set of QA pairs. The precision explains that QA-based annotation is more informative as compared to WikiEvent arguments.

Figure 3 shows a heatmap based on the Top 15 WikiEvent argument *roles* which correspond with the QAEVENT Wh-word. It is evident from the heatmap that "Who" is related to roles at personal level such as VICTIM, PARTICIPANT, DEFENDANT etc. Similarly "Where" is almost always related to some locative argument roles such as PLACE, DESTINATION and TARGET. The Wh-word "What" is often used to reason about the cause and it is clear from the heatmap that annotators used this word with argument roles such as ARTIFACT and EXPLOSIVE DEVICE. These are logical and unsurprising correlations which support the claim that our annotations help to create a more understandable annotations.

4.2 Vocabulary

The annotators are asked to follow the template and the vocabulary which they can use is open, apart from the Wh-word and the trigger. This leads to an interesting finding of the words which immediately follow the Wh-word words. For example the question *"Who thwarted the attack?"* contains the word *"thwarted"* which was not present in the corresponding document but occurs in the question. We mostly believe this is because that annotators use synonyms quite often as their level of familiarity with words vary.

The upper word cloud of Figure 4 includes phrases which come immediately after the "Who", "Whom" and "How". "How" is often associated with the quantity and it is also observed from the word cloud that "many" appears as one of the most frequent words. "Who" and "Whom" are generally related to person which explains the occurrence of words such as "killed", "died" etc. Similarly, the bottom word cloud of what follows "What", "When", "Where", and "Why". The results are in lieu with the observation of previous studies that mention "When" and "Where" to be associated with temporal and spatial entities (He et al., 2015; Michael et al., 2018). "What" is often associated

```
[System (M_1)] You help provide questions and answers to annotate passages
[User (M_2)] {Prompt:  "You are an assistant that reads through a passage and provides
all possible question and answer pairs to the bolded word.  The bolded word is the event
trigger, and the questions will help ascertain facts about the event.  The questions
must be in this template:wh* verb subject trigger object1 preposition object2 Wh* is a
question word that starts with wh (i.e.  who, what, when, where).  The subject performs
the action.  The object is the person, place, or thing being acted upon by the subjectś
verb.  A preposition is a word or group of words used before a noun, pronoun, or noun
phrase to show direction, time, place, location, spatial relationships, or to introduce
an object.  Answers MUST be direct quotes from the passage.  Do not ask any inference
questions.Please make sure to provide an answer for every question and limit the maximum
number of question answer pair to 5"}
[User (M_3)] {"This is a demonstration of what I want {demonstration}"}
[User (M_4)] {Here is the passage:  {passage}.  The trigger is:  {trigger}'}
```

Table 3: Discussion template for a user to prompt ChatGPT model to generate question and answer pairs.

with reason and it can be seen in the word cloud that words such as "caused" and "happened" occur frequently.

5 Question Answer Pair Generation

In this Section, we present the various Question Answer Pair Generation (QAG) methods. Formally, given a document D, for every trigger t_i in D, we aim to generate Question Answer Pairs $\{(Q_1, A_1), ...(Q_j, A_j)\}$ to annotate arguments of triggers t_i, where each QA pair represents one argument of the event. A_j is supposed to be the answer corresponding to Q_j.

5.1 Methods

This subsection discusses the ideas and details for the various baseline methods.

Rule-based Question Generation The general idea is that we first apply an event extraction (IE) system to obtain the arguments of the trigger word. Then treat the argument as the answer and generate its corresponding question.

We first create a mapping $f : r_i \rightarrow$ Wh* between the WikiEvent argument roles and the set of Wh-words based on its detailed schema[1]. Then for question generation, we first apply the Gen-IE system (Li et al., 2021) which applies BART model (Lewis et al., 2019) for extracting the event arguments under the WikiEvent schema. For each WikiEvent argument role r (e.g. ATTACKER, PLACE), we have extracted arguments as $A_1, ..., A_n$. Then we treat each argument A_i as the answer span, map from its role r to a Wh-word, and generate the question based on the Wh-word and the trigger t following the template in Section 3.1. For example, if the extracted argument is "Mr. Dash" and "estate", and the trigger is "distributed", we can generate the QA pair as ("who distributed the estate?", "Mr. Dash").

[1] https://github.com/raspberryice/gen-arg/blob/main/event_role_KAIROS.json

Prompting based Question Generation We also investigate prompting large language models (LLMs) for generating QA pairs. The general prompt we use is illustrated in Table 3. The prompt P consists of several messages which enable the LLM model to generate QA pairs. We initially ask the model to help generate question and answers which is considered as M_1; M_2 consists of the main instruction which helps the LLM to follow our guidelines to generate QA Pair. We also set the specific requirements on avoiding multi-hop questions; M_3 consists a sample document followed by a set of QA pairs (a demonstration); The last message M_4 corresponds to the actual input which is the document followed by event trigger in consideration. In our study on the training set, LLM generates many QA pairs which is not controllable and far beyond our requirements, we restrict the number of pairs to be five by adding this constraint in P.

The general prompt is used for our baseline **Q-First (ChatGPT)** by default. In order to investigate the influence of answer span to question when generation the QA pair, we also propose **A-First (ChatGPT)**. Intuitively the model first extracts potential answer spans and ask questions based on it (similar to rule-based method above). In terms of prompt, this method mainly differs with question first based prompt in the fact that we force the LLM to generate the answer first followed by the question. In M_2 to prompt it to "generate answer question pairs", and change the order of question and answer in the demonstration. Our **Q-First (GPT-4)** use a prompt similar to Q-First (ChatGPT). Q-First (GPT-4) uses GPT-4 for query processing and it has been established to be more suited to follow detailed and complex instructions (Takagi et al., 2023). In our trials, we find that GPT-4 tends to generate even more complicated questions, so in demonstration we provide more representative single-hop questions for each trigger.

5.2 Experiments

Metrics and Setups We report recall, precision and F1 scores based on the matching between our generated questions and gold questions. By matching we use maximal intersection over union (IOU), a QA pair is aligned with another pair that IOU >= threshold on a token-level, we report results using two thresholds which are 0.5 and 0.4 (Pyatkin et al., 2020). The recall is proportion of gold questions that are matched by any of the generated question; the precision is the proportion of generated questions that can match to any of the gold question. Recall is more important for our task, because of task's nature on extracting more comprehensive arguments of the events.

We also see the performance variation based the context provided as the input to various model. We consider two settings: (1) Under Entire Document Context and (2) Under Sentence level context. For the sentence level context, we calculate the metrics if and only if the answers lie within the context. This helps us to understand how questions generated for the entire context (document Level) is beneficial to annotate the document.

	Prec	Recall	F1
IOU>0.5			
Rule_Based	0.23	0.17	0.19
Q-first (ChatGPT)	0.06	0.10	0.07
A-first (ChatGPT)	0.08	0.14	0.10
Q-first (GPT-4)	0.20	**0.39**	0.26
IOU>0.4			
Rule_Based	0.37	0.27	0.31
Q-first (ChatGPT)	0.11	0.18	0.13
A-first (ChatGPT)	0.15	0.27	0.20
Q-first (GPT-4)	0.27	**0.52**	0.36

Table 4: QG performance within the document-level context. Performance is substantially lower than the sentence-level performance (Table 5), demonstrating our task setting is more challenging than prior work.

	Prec	Recall	F1
IOU>0.5			
Rule_Based	0.23	0.44	0.30
Q-first (ChatGPT)	0.06	0.05	0.06
A-first (ChatGPT)	0.12	0.23	0.16
Q-first (GPT-4)	0.28	**0.85**	0.42
IOU>0.4			
Rule_Based	0.40	**0.77**	0.53
Q-first (ChatGPT)	0.10	0.08	0.09
A-first (ChatGPT)	0.27	0.51	0.36
Q-first (GPT-4)	0.35	**1.00**	0.52

Table 5: QG performance under the within sentence-level context.

Results We discuss the performance of all the baseline models across the two settings: **(1) Document-level Context**: Top part of Table 4 shows the results for IOU with threshold of 0.5 with the document-level context. We get the maximum recall for GPT-4 based baseline which is expected since GPT-4 understands multi-step instructions better than other baselines. A good precision is also seen for rule based method because that these questions are shorter and often include phrases in golden questions which is generated based on the template. Bottom part of Table 4 shows the results for IOU-0.4. Relaxing the threshold level increases the number of matches (resulting in higher precision and recall). A similar trend is seen in terms of recall being highest for GPT-4 based baseline. In general, an interesting result is that A-first based prompts results in a recall higher than Q-first based prompts. We believe this is because we constrain our guidelines more so that an answer is phrased such that it keeps the question somewhat similar to set of golden questions. On the other hand apart from Wh-word and trigger no other field has a restricted domain of words. **(2) Sentence-level Context**: We also inspect the quality of questions based on a sentence-level context. In this setting we only consider the set of generated questions and golden questions whose answers are within one sentence containing the trigger word. The results all grow significantly, proving the lower difficulty of the sentence-level task (i.e. as in previous work of QA-SRL, QAMR and QADisourse). At IOU-0.5, we see an increment in the recall for all the baselines as compared to the document-level setting. This happens due to the fact a restricted set of generated and golden questions (within one sentence) results in more overlaps among the questions. A substantial improvement is seen for the recall of GPT-4 baseline ascertaining the fact that GPT-4 can follow the prompt instructions better as compared to other baselines. For IOU-0.4, relaxing the IOU threshold level results in an increase of both precision and recall for all the models. At this level, GPT-4 generates all the golden questions. Rule-based baseline has more substantial improvements as compared to ChatGPT based models. We speculate this happens because rule-based generation gives us a shorter length questions with a high possibility of the word occurring in the context.

6 Answer Identification (based on Golden Questions)

6.1 Methods

We design a QA system also with LLM. More specifically, ChatGPT to generate the answers for each golden question in the test set. Table 7 in the Appendix shows the prompt that we use to generate the answer based on question. Basically, given the input, we design the prompt such that it enables LLM to frame an answer based on the messages in it. In the system message M_1, we initially instruct the system, to give us one answer based on the context. M_2 is the main instruction to the LLM model in that we specify the constraints on the answer generated. After manual inspection of several generated answers we also provide the span of answer and the format of output. After this message we add a demonstration M_3.

6.2 Experiments

Metrics and Setups For evaluating the quality of answer identification (question answering) methods, we report precision, recall, F1 , and exact match (EM) based on the metric calculation in (Yang et al., 2018)

	Precision	Recall	F1	EM
ChatGPT	0.45	**0.70**	0.50	0.24
ChatGPT w/ demo.	0.47	0.62	0.49	0.27

Table 6: Results of Answer Identification.

Results Table 6 presents the results of the experiments for answer identification. **LLM with Demo** enables in-context learning (Dong et al., 2023) which is a paradigm where the LLM generate the results based on context and small set of examples.

We observe that LLM with demo has a higher recall as compared to LLM without demo. This indicates that answers generated by LLM with demo is closer to the set of golden questions. However, LLM without demo has a higher precision because the answers are more similar to LLM without demo. LLM without demo achieves higher exact match as compared to LLM with demo, but this does not confirm that the answer generated by LLM with demo is wrong. For example, If the question is *"Who is accused of playing a role in the murder?"* and answer generated by the LLM with demo is *"Hussein and his minions"* whereas the golden answer is *"Saddam Hussein and his minions"*, EM metric will return 0.

7 Conclusion

In this work we show that document-level events can be represented using question and answer pairs. This representation results in a scalable and fast annotations from crowd sourcing without much linguistic background. We present a set of guidelines which can be used to collect event QA pairs and conducted crowdsourcing for collecting a QAEVENT corpus. We found that: (1) annotation is more efficient under our paradigm, it takes much shorter time as compared to the original WikiEvent annotation; (2) our annotations align well with WikiEvent event arguments, and in addition cover more nuanced and fine-grained arguments/attributes. Finally we establish both rule-based and LLM-based baselines on our benchmark.

Limitations

The current QAEVENT based annotation has a good coverage and can be used to annotate passages quickly and efficiently. However, we observe that sometimes the annotations does not cover certain WikiEvent argument roles. Ex(5) in Table 1 represents one such scenario. In this case we do not have a question and answer pair for this role. Further investigation is required to understand this behavior.

Based on the current proposed methods for question generation we generate a set of question and answers based on template based mapping which sometimes results in grammatically incorrect answers. For example- based on the trigger word "speaking" and the WikiEvent role to be an artifact then the rule based question generation will result in "What speaking?" Future work will involve adding some kind of pruning mechanism to both restrict the number of questions and generating grammatically correct ones. The current prompts generate questions and answers which have a good recall, however it is observed that LLM based models generate QA Pairs which do not follow the guidelines or are inference based.

References

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

George Doddington, 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)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

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.

Xinya Du and Heng Ji. 2022. Retrieval-augmented generative question answering for event argument extraction. In *EMNLP*.

Xinya Du, Zixuan Zhang, Sha Li, Pengfei Yu, Hongwei Wang, Tuan Lai, Xudong Lin, Ziqi Wang, Iris Liu, Ben Zhou, Haoyang Wen, Manling Li, Darryl Hannan, Jie Lei, Hyounghun Kim, Rotem Dror, Haoyu Wang, Michael Regan, Qi Zeng, Qing Lyu, Charles Yu, Carl Edwards, Xiaomeng Jin, Yizhu Jiao, Ghazaleh Kazeminejad, Zhenhailong Wang, Chris Callison-Burch, Mohit Bansal, Carl Vondrick, Jiawei Han, Dan Roth, Shih-Fu Chang, Martha Palmer, and Heng Ji. 2022. RESIN-11: Schema-guided event prediction for 11 newsworthy scenarios. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, pages 54–63, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Nicholas FitzGerald, Julian Michael, Luheng He, and Luke Zettlemoyer. 2018. Large-scale qa-srl parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2051–2060.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 643–653.

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.

Qi Li, 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, Sofia, Bulgaria. Association for Computational Linguistics.

Qian Li, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, et al. 2022. A survey on deep learning event extraction: Approaches and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

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.

(LDC) Linguistic Data Consortium. 2005. English annotation guidelines for events. `https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf`.

Xiao Liu, Heyan Huang, and Yue Zhang. 2019. Open domain event extraction using neural latent variable models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2860–2871, Florence, Italy. Association for Computational Linguistics.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, and Bao Ge. 2023. Summary of chatgpt/gpt-4 research and perspective towards the future of large language models.

Julian Michael, Gabriel Stanovsky, Luheng He, Ido Dagan, and Luke Zettlemoyer. 2018. Crowdsourcing question-answer meaning representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 560–568.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.

Valentina Pyatkin, Ayal Klein, Reut Tsarfaty, and Ido Dagan. 2020. QADiscourse - Discourse Relations as QA Pairs: Representation, Crowdsourcing and Baselines. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2804–2819, Online. Association for Computational Linguistics.

Paul Roit, Ayal Klein, Daniela Stepanov, Jonathan Mamou, Julian Michael, Gabriel Stanovsky, Luke Zettlemoyer, and Ido Dagan. 2020. Crowdsourcing a high-quality gold standard for qa-srl. In *ACL 2020 Proceedings, forthcoming*. Association for Computational Linguistics.

Beth M. Sundheim. 1992. Overview of the fourth Message Understanding Evaluation and Conference. In *Fourth Message Uunderstanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

Soshi Takagi, Takashi Watari, Ayano Erabi, Kota Sakaguchi, et al. 2023. Performance of gpt-3.5 and gpt-4 on the japanese medical licensing examination: comparison study. *JMIR Medical Education*, 9(1):e48002.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

A Crowdsourcing details

See Figure 5.

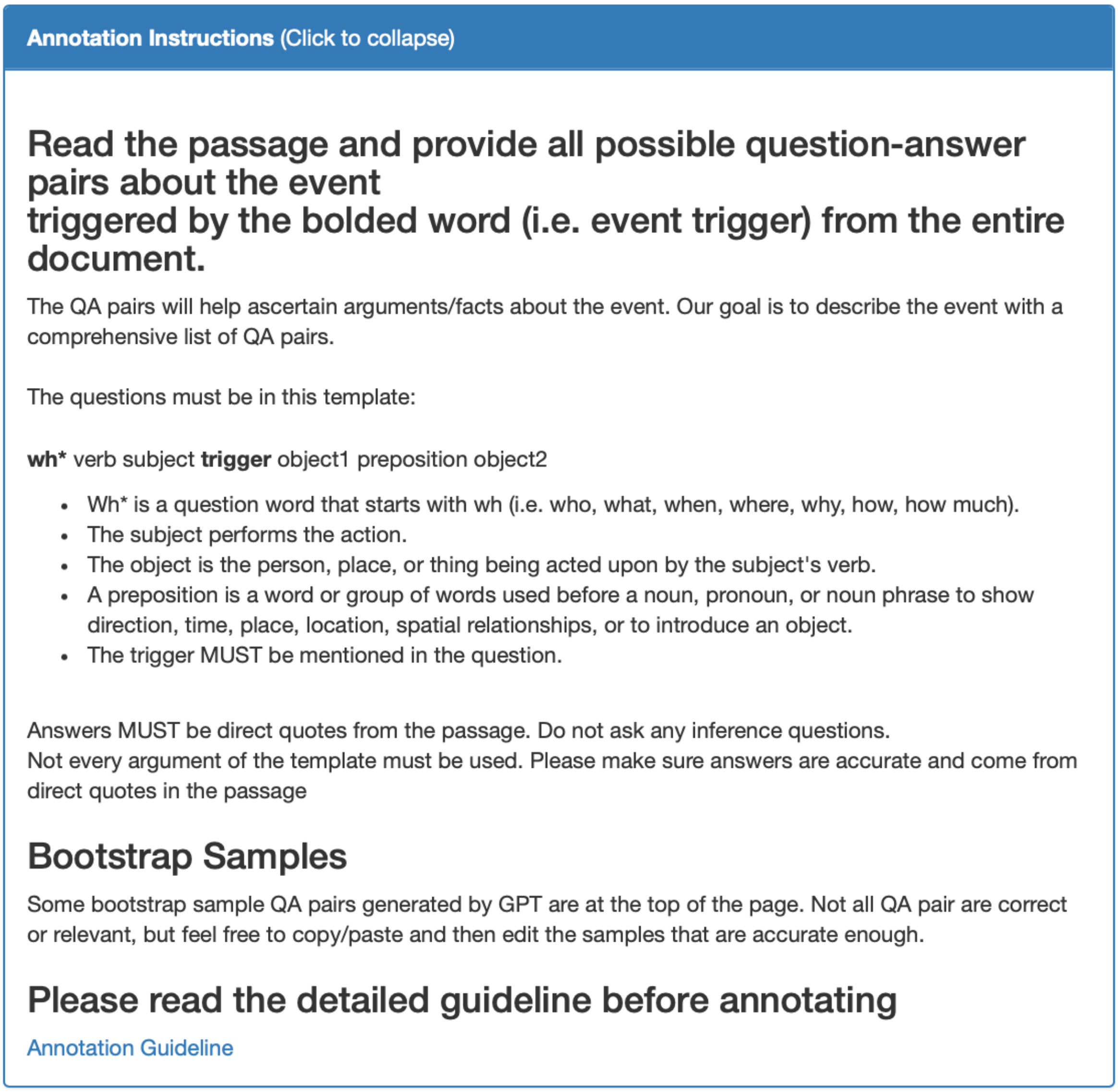
A.1 Full annotation guidelines given to workers

B Interface for Annotation Task

Refer to Figure 6.

C Answer Identification Prompt

Refer to Table 7.



Annotation Instructions (Click to collapse)

Read the passage and provide all possible question-answer pairs about the event triggered by the bolded word (i.e. event trigger) from the entire document.

The QA pairs will help ascertain arguments/facts about the event. Our goal is to describe the event with a comprehensive list of QA pairs.

The questions must be in this template:

wh* verb subject **trigger** object1 preposition object2

- Wh* is a question word that starts with wh (i.e. who, what, when, where, why, how, how much).
- The subject performs the action.
- The object is the person, place, or thing being acted upon by the subject's verb.
- A preposition is a word or group of words used before a noun, pronoun, or noun phrase to show direction, time, place, location, spatial relationships, or to introduce an object.
- The trigger MUST be mentioned in the question.

Answers MUST be direct quotes from the passage. Do not ask any inference questions.
Not every argument of the template must be used. Please make sure answers are accurate and come from direct quotes in the passage

Bootstrap Samples

Some bootstrap sample QA pairs generated by GPT are at the top of the page. Not all QA pair are correct or relevant, but feel free to copy/paste and then edit the samples that are accurate enough.

Please read the detailed guideline before annotating

Annotation Guideline

Figure 5: Annotation Guidelines.

[System (M_1)] You help provide one answer of length not more than len(answer) to the question based on context
[User (M_2)] {Prompt: "You are an assistant that reads through a passage and provides the answer based on passage and trigger. The bolded word is the event trigger. Answers MUST be direct quotes from the passage. Make sure to generate the answers based on the context, the trigger and corresponding question.In a new line, output the answer. Do not output anything else other than the answer in this last line."}
[User (M_3)] {"This is a demo of what I want demo"}
[User(M_4)] {Context: passage Trigger: trigger Question: question Answer: }

Table 7: Discussion template for a User to query GPT 3.5 Turbo model to generate answer

Figure 6: Screenshot of the Crowdsourcing User Interface.