Asking and Answering Questions to Extract Event-Argument Structures

Md Nayem Uddin[♠] **Enfa Rose George**[♦] **Eduardo Blanco**[♦] **Steven R. Corman**[♠]
muddin11@asu.edu, enfageorge@arizona.edu, eduardoblanco@arizona.edu, steve.corman@asu.edu

[♠]Arizona State University, Tempe, AZ.
[♦]University of Arizona, Tuscon, AZ.

Abstract

This paper presents a question-answering approach to extract document-level event-argument structures. We automatically ask and answer questions for each argument type an event may have. Questions are generated using manually defined templates and generative transformers. Template-based questions are generated using predefined role-specific wh-words and event triggers from the context document. Transformer-based questions are generated using large language models trained to formulate questions based on a passage and the expected answer. Additionally, we develop novel data augmentation strategies specialized in inter-sentential event-argument relations. We use a simple span-swapping technique, coreference resolution, and large language models to augment the training instances. Our approach enables transfer learning without any corpora-specific modifications and yields competitive results with the RAMS dataset. It outperforms previous work, and it is especially beneficial to extract arguments that appear in different sentences than the event trigger. We also present detailed quantitative and qualitative analyses shedding light on the most common errors made by our best model.

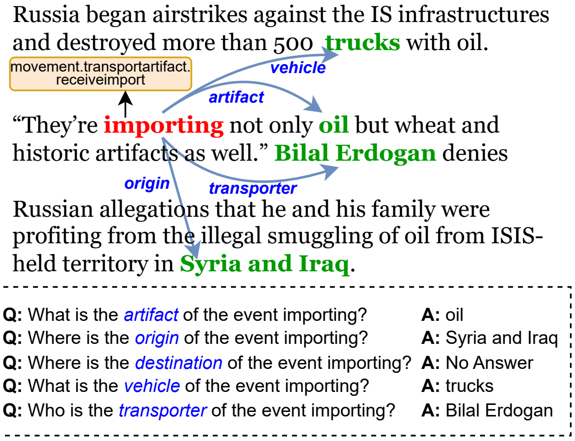
Keywords: Event-Argument Extraction, Information Extraction

1. Introduction

Extracting event-argument structures is an important problem in natural language understanding (Doddington et al., 2004; Aguilar et al., 2014). At its core, it is about identifying entities participating in events and specifying their role (e.g., the *giver*, *recipient*, and *thing given* in a *given* event). Event triggers (i.e., words instantiating events) include both nouns (e.g., *election*, *speech*), and verbs (e.g., *vote*, *talk*). Regardless of specific events and relations, event-argument structures are beneficial for applications such as news summarization (Li et al., 2016) and coreference (Huang et al., 2019; Zhang et al., 2015).

Traditionally, corpora are limited to arguments within the same sentence an event belongs to. Inter-sentential arguments are more challenging and have received less attention (Gerber and Chai, 2010; Ruppenhofer et al., 2010). Figure 1 presents an example from RAMS (Ebner et al., 2020), the largest corpus annotating multi-sentence event-argument structures. Two out of four event-argument relations cross sentence boundaries.

In this paper, we tackle the problem of extracting event-argument structures. As exemplified in Figure 1, we cast the problem as a question-answering task. We ask one question for each argument an event may have, and rely on transformers to find answers pinpointing the text corresponding to the argument in the input document (or, alternatively, indicate that there is no answer).



Russia began airstrikes against the IS infrastructures and destroyed more than 500 **trucks** with oil.

"They're **importing** not only **oil** but wheat and historic artifacts as well." **Bilal Erdogan** denies Russian allegations that he and his family were profiting from the illegal smuggling of oil from ISIS-held territory in **Syria and Iraq**.

Q: What is the *artifact* of the event importing?	**A:** oil
Q: Where is the *origin* of the event importing?	**A:** Syria and Iraq
Q: Where is the *destination* of the event importing?	**A:** No Answer
Q: What is the *vehicle* of the event importing?	**A:** trucks
Q: Who is the *transporter* of the event importing?	**A:** Bilal Erdogan

Figure 1: Event trigger (*importing*) and its arguments in the same (artifact and transporter) and surrounding sentences (vehicle and origin). We cast the problem of extracting the arguments of an event as a question-answering task. Questions are automatically generated (and answered) for each argument an event may have.

The main contributions of this paper are:

- Two approaches to formulate the questions: template- and transformer-based;
- Data augmentation strategies to improve the extraction of inter-sentential arguments;
- Quantitative results showing the benefits of our approach, including transfer learning; and

- Error analysis shedding light into the most challenging event-argument relations.

The framework presented in this paper does not depend on any annotation framework, set of event types or argument types, domain, or corpora. The only requirement is a list of argument types an event may have. Most event-argument annotation efforts satisfy this requirement, including PropBank (Palmer et al., 2005), NomBank (Meyers et al., 2004), FrameNet (Baker et al., 1998), RAMS, ACE (Doddington et al., 2004), and WikiEvents (Li et al., 2021). All the examples and experiments in this paper, however, draw from the RAMS dataset (Section 3). We reserve for future work experimenting with other corpora.

2. Previous Work

Extracting event-argument structures, also referred to as event extraction (Ahn, 2006), includes identifying event triggers and its arguments. The task has a long history in the field (Grishman and Sundheim, 1996; Doddington et al., 2004). Initially, datasets focused on extracting arguments within the same sentence than the verb (Palmer et al., 2005; Walker et al., 2006). There are also corpora focused on inter-sentential arguments (Gerber and Chai, 2010; Ruppenhofer et al., 2010; Ebner et al., 2020; Li et al., 2021). Early models were based on handcrafted features (Li et al., 2013; Liao and Grishman, 2010; Hong et al., 2011). Like most NLP tasks, models to extract event-argument structures experienced a transformative shift building on word embeddings, RNNs, and CNNs (Chen et al., 2015; Nguyen et al., 2016).

Transformer-based approaches are currently the best performing. Some efforts assume event triggers and argument spans are part of the input and present classifiers to identify the argument type (Ebner et al., 2020; Chen et al., 2020). Unlike them—and like the remaining previous works discussed below—we only assume event triggers. At a high-level, efforts to identify argument spans and argument types can be categorized into sequence labeling, casting the problem as a question-answering task, and using generative models. Sequence label classifiers approach the problem with the traditional BIO encoding (Ramponi et al., 2020). Framing the problem in terms of questions and answers is popular (Du and Cardie, 2020; Liu et al., 2020; Li et al., 2020). Doing so enables zero-shot (Lyu et al., 2021) and few-shot (Sainz et al., 2022) predictions. Li et al. (2021), Ma et al. (2022) and Du et al. (2021) leverage generative language models (Raffel et al., 2020; Lewis et al., 2020). Language generation facilitates a more flexible extraction by *generating* the arguments rather than identifying spans in the input document. Transfer learning has also been explored, including semantic roles (Zhang et al., 2022), abstract meaning representations (Xu et al., 2022), and frame-aware knowledge distillation (Wei et al., 2021). Our approach casts the problem as a question answering task. We introduce (a) a template- and transformer-based approach to generate questions and (b) streamline transfer learning for extracting event-argument structures.

Supervised models demand annotated examples. To mitigate this need, unsupervised learning (Huang et al., 2016; Yang et al., 2018) and weakly supervision (Chen et al., 2017; Kar et al., 2021) have been proposed. Data augmentation approaches (Liu et al., 2021; Gao et al., 2022) have been reported useful. We also explore data augmentation. Unlike previous works, our augmentation strategies target additional inter-sentential arguments. Surprisingly, we show that arguably the simplest strategy yields the best results.

3. The RAMS Dataset

Roles Across Multiple Sentences (RAMS) (Ebner et al., 2020) is a dataset annotating event-argument structures. The source texts are news articles. The annotations follow the AIDA-1 ontology.[1] This ontology contains a 3-level event hierarchy (e.g., *transaction.transfermoney.payforservice*) and the argument types each event type may have (e.g., *giver*, *recipient*, *beneficiary*, *money*, and *place*). The ontology contains 139 events types and 65 argument types (some are relevant to many event types, e.g., *place* appears with many events).

The RAMS annotations include (a) event triggers (i.e., words instantiating an event type) and (b) the arguments of that event trigger (i.e., the word spans for each argument type). We use the term *event-argument structure* to refer to an event trigger and its arguments. An event-argument structure need not include all the argument types in the AIDA-1 ontology (e.g., *importing* is missing the *destination* argument in Figure 1). Event triggers need not belong to the bottom level in the event hierarchy (e.g., an event trigger may belong to *transaction.transfermoney* if no child is a good fit).

The event-argument structures in RAMS are annotated across sentences. First, annotators identified event triggers. Second, they identified arguments (as defined in AIDA-1) up to two sentences before or after, as arguments are rarely found outside this window. In practical terms, this means that documents in RAMS are 5 sentences long; the only exceptions are event-arguments structures whose event trigger was found at the very beginning or end of the source news articles.

[1] LDC{2019E04, 2019E07, 2019E42, 2019E77}

	Train	Dev	Test
# documents	3,194	399	400
# events	7,329	924	871
# arguments	17,026	2,188	2023
intra-sentential	14,018	1,811	1,667
inter-sentential	3,008	377	356
# arguments per event	2.23	2.36	2.32

Table 1: Basic statistics of the RAMS dataset.

Table 1 presents basic statistics of the RAMS dataset. It includes 9,124 event-argument structures annotated in 3,993 documents. There are 21,237 arguments in these structures. 3,741 (18%) of these argument are inter-sentential.

4. Asking and Answering Questions for Event-Argument Extraction

We cast the problem of extracting event-argument structures as a question answering problem and explore two approaches: (a) supervised learning with traditional transformers (Section 4.1) and (b) zero- and few-shot prompts with GPT3 (Section 4.2). As we shall see, the latter obtains much worse results. Inspired by previous work (Du and Cardie, 2020; Liu et al., 2021), we generate questions for each argument an event may have according to the AIDA-1 ontology. Answers are *No answer* if an argument is not present. Our novelties are as follows:

- Combining two approaches to generate questions: template-based and transformer-based;
- Exploring data augmentation strategies; and
- Showing that our approach can easily accommodate transfer learning with other corpora.

4.1. Supervised Question-Answering

Using supervised learning requires us to transform the RAMS event-argument structures into a set of questions and answers. We also explore data augmentation and transfer learning, two strategies that can be easily incorporated into our approach.

4.1.1. Generating Questions

We explore two *automated* approaches to generate questions asking for the arguments of an event trigger. The only requirement is to know a priori the name of the arguments an event may have, an assumption we share with previous work. Let us consider the event trigger in Figure 1, *importing*, which belongs to *movement.transportartifact.receiveimport*. This event has up to five arguments in AIDA-1: *transporter*, *artifact*, *vehicle*, *origin*, and *destination*.

Template-Based Generation. We use a straightforward template to generate questions: *Wh-word is the [argument type] of event [event trigger]?*, where *Wh-word* is selected from the following: *what*, *where*, *who*, and *how*. Questions are generated regardless of whether the argument is present in the input document. The answer is the text corresponding to the argument if it exists without any leading text (e.g., *The answer is [argument]*, *The [argument type] is [argument]*). If the argument does not exist, the answer is *No answer*. Five question-answer pairs are generated for our running example in the Figure 1:

1. Q: *Who is the transporter of the event importing?* A: Bilal Erdogan
2. Q: *What is the artifact of the event importing?* A: oil
3. Q: *What is the vehicle of the event importing?* A: trucks
4. Q: *Where is the origin of the event importing?* A: Syria and Iraq
5. Q: *Where is the destination of the event importing?* A: No answer

Transformer-Based Generation. The template-based approach results in correct question-answer pairs. These pairs, however, lack linguistic diversity. In order to alleviate this issue, we experiment with T5 (Raffel et al., 2020) to generate questions. Specifically, we use a version of T5 pre-trained with SQuAD (Rajpurkar et al., 2016) to generate questions (Wang et al., 2020).

The input is a document (5 sentences) and an argument. The output is a question whose answer is the argument. Following with the example in Figure 1, here are the questions generated by T5:

1. Q: *Who denied Russian allegations?* A: Bilal Erdogan
2. Q: *What did Russia destroy 500 trucks with?* A: oil
3. Q: *What type of vehicles did Russia destroy?* A: trucks
4. Q: *Where did ISIS hold territory?* A: Syria and Iraq

Note that T5-generated questions may be irrelevant to the event at hand. Indeed, none of the questions above are about *importing*. Additionally, some questions are wrong. For example, T5 struggles with the prepositional phrase attachment in question (2): the *oil* was carried by the *trucks*—it is not what the trucks were destroyed with. Despite these limitations, leveraging transformer-based questions yields better results (Section 5).

There is a caveat when generating questions with T5: we only generate questions for arguments that are present, not for all arguments in AIDA-1. As a result, transformer-based question generation is only applicable at training time.

4.1.2. Data Augmentation

RAMS includes inter-sentential arguments, but most of them are intra-sentential. Previous work has consistently reported worse results with inter-sentential arguments (Wei et al., 2021; Liu et al., 2021), so we designed six novel data augmentation strategies to improve results with inter-sentential arguments. We group the strategies into three categories: *Simple Swapping, Leveraging Coreference Resolution and Leveraging LLMs.* Figure 2 shows a gold example from the RAMS dataset and the results of six data augmentation strategies. We provide further details in Appendix C.

Simple Swapping. Our first strategies are the most straightforward and result in ungrammatical documents. We shift intra-sentential arguments outside their sentence and change the gold argument to point to the new position after moving to a different sentence. We use two strategies:

- *Plain*. Move the intra-sentential argument into a random sentence boundary including the beginning and end of the document (6 options for 5-sentence documents).
- *Verbose*. Copy the intra-sentential argument into a random sentence boundary including the beginning and end of the document. We use the following template to generate the text to be pasted: *The [argument type] of [event] is [argument]*.

For each event-argument structure in the original training split, each of these strategies result in as many additional event-argument structures as intra-sentential arguments in the original instance.

Leveraging Coreference Resolution. Transforming intra-sentential arguments into inter-sentential ones can be achieved by manipulating coreference chains. We follow two strategies:

- *Random mention*. Update intra-sentential arguments with an inter-sentential mention randomly selected from its coreference chain.
- *Most Meaningful mention*. Same as *Random mention* but selecting the most meaningful mention. We consider mentions that have more tokens and named entities (first and second criterion) to be more meaningful.

For each event-argument structure in the original training split, each strategy results in as many additional event-argument structures as intra-sentential arguments that are part of a coreference chain with at least one mention belonging to another sentence. A drawback of both strategies leveraging coreference resolution is that errors in coreference resolution lead to noisy augmented data.

Leveraging LLMs. Given the recent success of Large Language Models (LLMs), including efforts to use them for data augmentation (Yoo et al., 2021), we also experiment with them. Unlike the previous strategies, using LLMs has the potential to generate unconstrained augmented data.

First, we use paraphrasing without any prompting or customization. Specifically, we use PEGASUS (Zhang et al., 2020a) fine-tuned for paraphrasing (Zhou and Bhat, 2021). Given the input document, this model returns a paraphrased version.

Second, we experiment with GPT-3 prompting (Brown et al., 2020). After several refinements, we came up with the following prompt: *Rewrite the story like a newspaper article in N sentences. Include the event triggering word [event trigger] and event arguments [argument$_1$], [argument$_2$], [...], [argument$_n$] in the generated article*., where N is the number of sentences in the original document.

Rewriting the input document requires us to modify the gold span positions. The process is conceptually simple, but neither PEGASUS nor GPT3 are guaranteed to keep the tokens for each argument in the generated text. The mapping process inevitably results in unmapped events and argument. Out of 7,079 events and 17,026 arguments, we successfully map 66.6% events and 74.8% arguments with PEGASUS and 90.2% events and 93.5% arguments with GPT3.

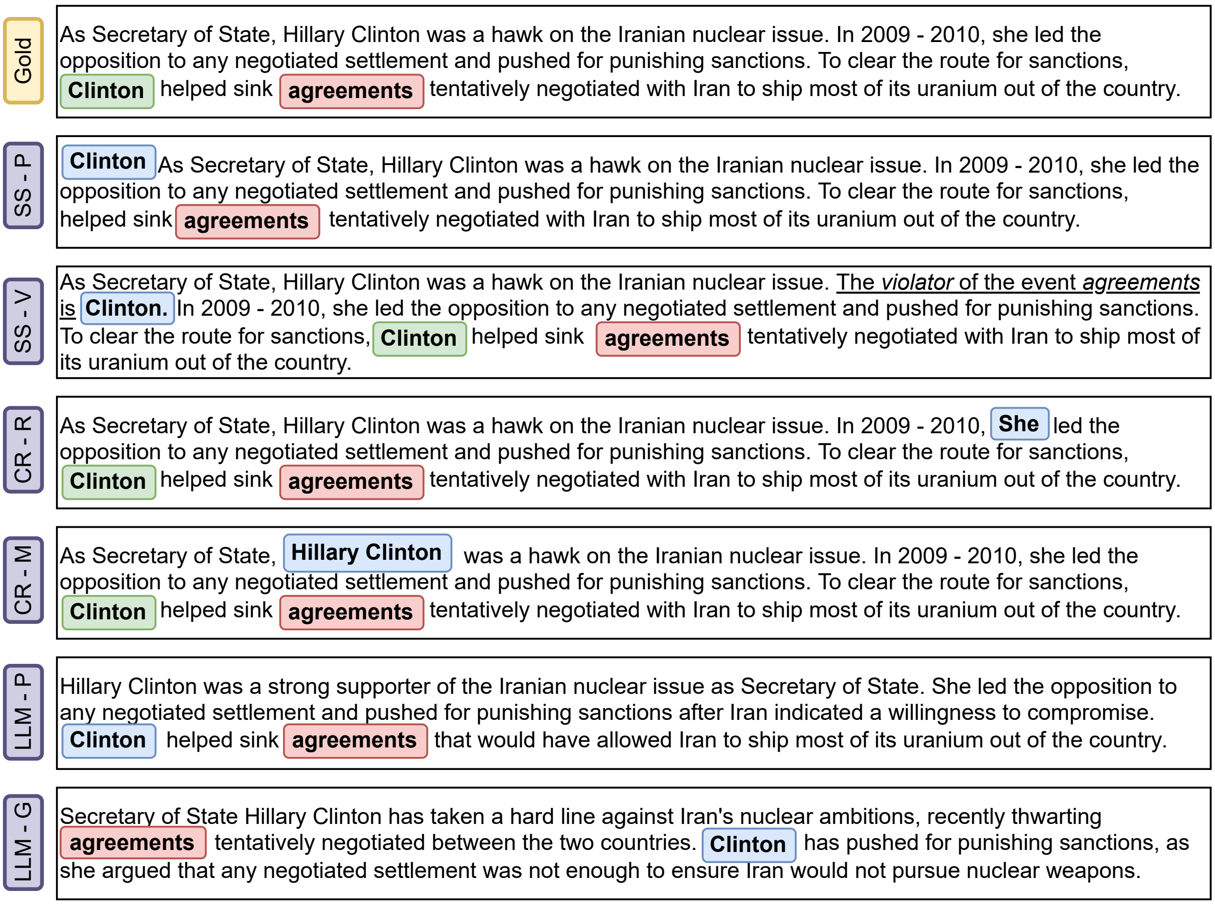
4.1.3. Transfer Learning

Transfer learning has been shown useful for extracting event-argument structures among others. Previous efforts project annotations (Huang et al., 2018) or reuse existing corpora in a specialized manner (Zhang et al., 2022). We take a streamlined approach: transform existing corpora into questions and answers using the *same* methods described in Section 4.1.1. We work with:

ACE (Walker et al., 2006). It contains 5,349 event triggers annotated in broadcast conversations and news, newsgroups, phone conversations, and weblogs. ACE considers 8 event types, 33 event subtypes, and 36 argument types.

WikiEvents (Li et al., 2021). It contains 3,951 event triggers annotated in Wikipedia pages. It includes 50 event types and 59 argument types.

QA-SRL (He et al., 2015). It includes crowdsourced questions and answers encoding predicate-argument structures. Argument do not have a specific type; the question wording captures their role. QA-SRL includes 299,308 question-answer pairs.



Gold: As Secretary of State, Hillary Clinton was a hawk on the Iranian nuclear issue. In 2009 - 2010, she led the opposition to any negotiated settlement and pushed for punishing sanctions. To clear the route for sanctions, **Clinton** helped sink **agreements** tentatively negotiated with Iran to ship most of its uranium out of the country.

SS - P: **Clinton** As Secretary of State, Hillary Clinton was a hawk on the Iranian nuclear issue. In 2009 - 2010, she led the opposition to any negotiated settlement and pushed for punishing sanctions. To clear the route for sanctions, helped sink **agreements** tentatively negotiated with Iran to ship most of its uranium out of the country.

SS - V: As Secretary of State, Hillary Clinton was a hawk on the Iranian nuclear issue. The *violator* of the event *agreements* is **Clinton.** In 2009 - 2010, she led the opposition to any negotiated settlement and pushed for punishing sanctions. To clear the route for sanctions, **Clinton** helped sink **agreements** tentatively negotiated with Iran to ship most of its uranium out of the country.

CR - R: As Secretary of State, Hillary Clinton was a hawk on the Iranian nuclear issue. In 2009 - 2010, **She** led the opposition to any negotiated settlement and pushed for punishing sanctions. To clear the route for sanctions, **Clinton** helped sink **agreements** tentatively negotiated with Iran to ship most of its uranium out of the country.

CR - M: As Secretary of State, **Hillary Clinton** was a hawk on the Iranian nuclear issue. In 2009 - 2010, she led the opposition to any negotiated settlement and pushed for punishing sanctions. To clear the route for sanctions, **Clinton** helped sink **agreements** tentatively negotiated with Iran to ship most of its uranium out of the country.

LLM - P: Hillary Clinton was a strong supporter of the Iranian nuclear issue as Secretary of State. She led the opposition to any negotiated settlement and pushed for punishing sanctions after Iran indicated a willingness to compromise. **Clinton** helped sink **agreements** that would have allowed Iran to ship most of its uranium out of the country.

LLM - G: Secretary of State Hillary Clinton has taken a hard line against Iran's nuclear ambitions, recently thwarting **agreements** tentatively negotiated between the two countries. **Clinton** has pushed for punishing sanctions, as she argued that any negotiated settlement was not enough to ensure Iran would not pursue nuclear weapons.

Figure 2: Examples of the data augmentation strategies (gold event: *agreements*, highlighted in red; gold argument: *Clinton*, highlighted in green). Blue highlights indicate the arguments in the augmented samples. SS stands for Simple Swapping (P: Plain, V: Verbose), CR for Coreference Resolution (R: Random, M: Most Meaningful), and LLM for Large Language Model (P: Pegasus, G: GPT-3). In the gold sample, the event-argument is intra-sentential. Five of the six data augmentation strategies result in an inter-sentential argument.

4.2. Zero- and Few-Shot Question Answering

Large language models are credited with having emergent abilities (Wei et al., 2022). They are also capable of in-context learning (Brown et al., 2020), meaning that they can (presumably) solve problems with a small number of training examples when given instructions (Wang et al., 2022).

In order to test the aforementioned abilities when it comes to extracting event-argument structures across sentences, we experiment with GPT-3 and zero- and few-shot approaches. We provide prompt examples with details in Appendix A

Zero-Shot. We prompt GPT-3 with the input document (five sentences) and the questions generated with the template-based approach (Section 4.1.1). Note that the transformer-based approach to generate questions cannot be used as it requires the answers to the questions (i.e., the arguments we are prompting GPT-3 to find).

Few-Shot. Few-shot prompts is similar to zero-shot prompts except that they are preceded by two randomly selected data samples from the training split (using the same format than the expected answer). These examples also include questions without answers.

5. Quantitative Results and Analyses

We present results with the test split of RAMS (mean and standard deviation of five runs).[2] All our models are tuned with the train and validation splits of RAMS (and the same splits of the additional corpora with transfer learning). For data augmentation with coreference, we use the model by Clark and Manning (2016). All our models use

[2] Code including dataset transformed into questions and answers are available at https://github.com/nurakib/event-question-answering

	Base	Large
Supervised (RAMS) with Template-Based Questions	42.58±0.73	48.23±0.82
+ Merging Transformer-based Questions	45.39±0.13*	**50.69±1.52***
+ Blending Augmented Data		
Simple Swapping		
Plain (α = 0.4)	45.20±0.75*†	49.61±0.93*
Verbose (α = 0.6)	**46.86±0.30***†	49.97±0.41*
Leveraging Coreference Resolution		
Random mention (α = 0.4)	44.65±0.92*	48.38±0.52*
Meaningful mention (α = 0.4)	45.89±0.80*†	49.12±0.55*†
Leveraging Large Language Models		
Pegasus (α = 0.2)	46.72±1.11*†	47.72±0.21
GPT3 (α = 0.4)	45.89±1.55*†	48.45±0.37*

Table 2: Results (F1) obtained with the test split of RAMS (mean and standard deviation of five runs). Merging transformer-based questions is useful with both base and large models whereas blending augmented data yields improvements with base models. We indicate statistically significantly better results (McNemar test (McNemar, 1947), $p < 0.01$) with respect to *Supervised (RAMS) with Template-Based Questions* with an asterisk (*), and to *Merging Transformer-based Questions* with a dagger (†).

RoBERTa (Liu et al., 2019) for extractive question answering, similar to (Du and Cardie, 2020). We conducted experiments utilizing the base and large variants of RoBERTa to assess performance relative to the model size. We use Pytorch (Paszke et al., 2019) and HuggingFace transformers (Wolf et al., 2020). The only exception is GPT-3, which has its own API.

All models are evaluated using the official RAMS evaluation script (P, R, and F1; exact match with the ground truth). The only exception is GPT-3, with which we use a more lenient evaluation. Since the input document sometimes contains the text generated by GPT-3 more than once, there are multiple options to map the GPT-3 generated answer to token indices. We consider GPT-3 is correct if any of the mappings matches the ground truth.

Merging and Blending Using transformer-based questions, data augmentation, and transfer learning requires us to decide how to leverage these additional instances during the training process. Note that transformer-based questions are new question-answer pairs derived from the original RAMS instances, whereas data augmentation and transfer learning increase the number of training instances—and the question-answer pairs.

We explore two options: merging and blending. Merging is the simpler option: concatenate all the question-answer pairs and consider them equal during the training process. Blending (Shnarch et al., 2018) is more complicated and relies on the intuition that some instances (in our case, the question-answer pairs generated from RAMS) ought to be given more importance than the additional training instances. Specifically, blending starts the training process (first epoch) with the result of merging RAMS and additional instances. Then, in each epoch, the amount of additional instances is reduced by a hyperparameter α.

Results training with RAMS Table 2 presents the results with RoBERTa base and large models. We experiment with both merging and blending transformer-based questions and augmented data samples, resulting in four distinct combinations. Among these, we identified the most effective approach—merging transformer-based questions followed by blending augmented samples—which is presented in Table 2. Merging transformer-based questions brings a 2.81 F1 improvement (42.58 vs. 45.39) with the base model and a 2.46 F1 improvement (48.23 vs 50.69) with the large model. This is the case despite these automatically generated questions are noisy and are often worded with respect to other events than the event of interest (Section 4.1.1). Blending augmented data samples (*Simple Swapping*) is also beneficial with the base model (+1.47 F1; 45.39 vs. 46.86). Surprisingly, data augmentation strategies are not beneficial with the large model.

To summarize the results of merging and blending in Table 2, (a) merging transformer-based questions is more beneficial than blending, and (b) blending augmented data is more beneficial than merging. These results are intuitive, as transformer-based questions are rarely nonsensical but data augmentation is noisy—simple swapping results in non-grammatical texts, coreference resolution is noisy, and LLMs rephrasing is nondeterministic. In other words, blending outperforms merging when the additional data is less reliable.

Transfer learning brings statistically significantly better results (Table 3) with the base (F1: 48.53 vs. 46.86) and large model (52.89 vs. 50.69). QA-SRL, unlike ACE and WikiEvents, does not

	PLM	Base	Large
RAMS	RoBERTa	$46.86_{\pm0.30}$	$50.69_{\pm1.52}$
+ACE	RoBERTa	$\mathbf{48.53}_{\pm1.30}{}^{*}$	$\mathbf{52.89}_{\pm\mathbf{0.61}}{}^{*}$
+QA-SRL	RoBERTa	$44.94_{\pm0.79}{}^{*}$	$44.90_{\pm1.05}{}^{*}$
+WikiEvnt	RoBERTa	$46.58_{\pm0.68}$	$51.46_{\pm0.80}$
Previous Work (top 2)			
PAIE	BART	$49.50_{\pm0.65}$	$52.20_{\pm\text{n/a}}$
TSAR	BART	$48.06_{\pm\text{n/a}}$	$51.18_{\pm\text{n/a}}$

Table 3: Results (F1) obtained merging RAMS and related corpora (mean and standard deviation of five runs). We retrain the best model using only RAMS (boldfaced in Table 2, same as first row) and indicate statistically significantly better results (McNemar test (McNemar, 1947), $p < 0.01$) with an asterisk (*). We compare with PAIE (Ma et al., 2022) and TSAR (Xu et al., 2022).

annotate explicit event-argument structures. Instead, it encodes them in the wording of questions. We hypothesize that QA-SRL yields worse results because questions in QA-SRL are actual natural language written by crowdworkers rather than the result of instantiating templates or T5.

Comparison with Previous Work. Our best model outperforms the best published results with RAMS (Table 3): PAIE (Ma et al., 2022) obtains 52.2 F1 and we obtain 52.89 F1 (both large). Using the base model, however, we do not outperform previous work: PAIE obtains 49.5 F1 and we obtain 48.53 F1. We point out that PAIE requires role-specific parameters, meaning that unlike our approach, it cannot easily accommodate transfer learning and it is unable to make zero-shot predictions. Additionally, PAIE uses BART as the underlying pre-trained model, which has 15% more parameters than the one we use, RoBERTa.

Comparison with GPT-3 Zero-shot and few-shot prompting with GPT-3 obtains much worse results than our supervised question-answering approach.[3] The results in Table 4 show that GPT-3 obtains better results in a few-shot in-context learning setting, yet the performance lags behind the supervised models by a significant margin.

Are Inter-Sentential Arguments Harder? Table 5 details the results of our best models (boldfaced in Table 3) broken down by the number of sentences between the event trigger and argument. The improvements (%ΔF1) with respect to the simplest question-answering approach (only template-based questions, no data augmentation and no transfer learning; first supervised model in Table 2) are substantial regardless of distance between the event trigger and argument. For the base model, we observe 402% improvement when the argument appears two sentences after the event trigger. The improvements are substantial when arguments appear in the sentences before (36.5% F1, 30.5% F1) or the sentence after (45.6% F1). For the large model, arguments in the sentence before the event benefit the most (44.6% F1), followed by those two sentences before (38.5% F1). Arguments after the event also benefit (6.5% and 18.2% F1). In summary, our model is beneficial regardless of where the argument appears with respect to the event.

	P	R	F1
Ours, base	$54.48_{\pm0.86}$	$43.47_{\pm1.76}$	$48.53_{\pm1.30}$
Ours, large	$60.90_{\pm0.46}$	$46.74_{\pm0.78}$	$52.89_{\pm0.61}$
GPT-3			
Zero-shot	27.3	21.4	24.0
Few-shot	32.6	29.1	30.7

Table 4: Results obtained with our models (boldfaced in Table 3) and GPT-3 (*text-davinci-003*).

	Base		Large	
	F1	%ΔF1	F1	%ΔF1
2 before	29.17	+36.5	30.30	+38.5
1 before	31.23	+30.5	34.56	+44.6
same	54.43	+12.5	56.98	+6.7
1 after	22.08	+45.6	22.76	+6.5
2 after	27.91	+402.0	22.73	+18.2

Table 5: Results by our best models (boldfaced in Table 3) broken down by distance (# sentences) between arguments and events. %ΔF1 indicates the relative improvement with respect to training only with template-based questions and RAMS (first supervised model in Table 2). Our approach benefits all arguments, especially those that are not in the same sentence than the event.

Are Frequent Arguments Easier? It is a common belief that the more training data the easier it is to learn. Figure 3 provides empirical evidence showing that this is not the case when predicting event-argument structures in RAMS. We observe that per-argument F1 scores range from 0.34 to 0.70, but there is no pattern indicating that frequency correlates with F1 score. For example, infrequent events such as *employee* and *passenger* (2%) obtain results as high as those obtained with communicator (6%) and victim (5%).

Are Frequent Events Easier? No, they are not. Surprisingly, more training data for an event does

[3] We only report one run for zero- and few-shot with GPT-3 as we do not train it.

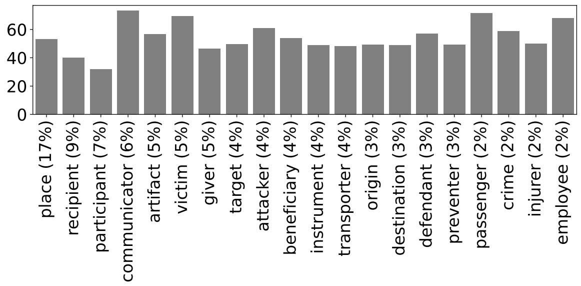


Figure 3: F1 per argument of our best model (boldfaced in Table 3, large). Frequency in training (between parenthesis) is only a weak indicator of F1, leading to the conclusion that some arguments are easier to learn. For example, *employee* is less frequent than *participant* yet the former obtains twice the F1 (0.70 vs. 0.33).

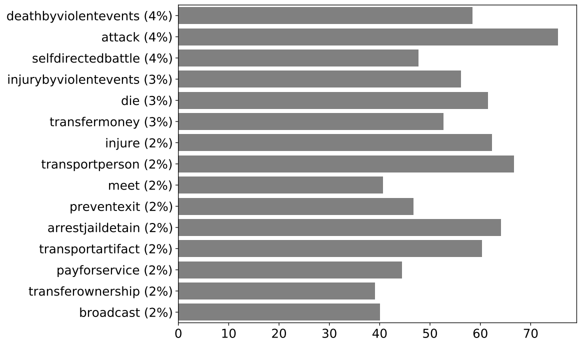
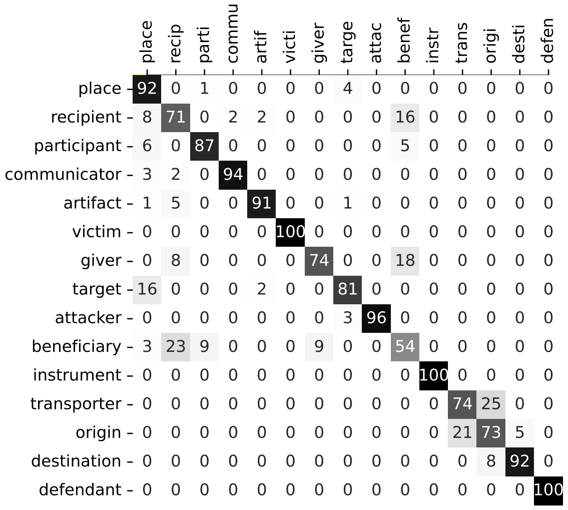


Figure 4: Average F1 per event (top 15 most frequent events) by our best model (boldfaced in Table 3, large). There is no clear relation between event frequency in training (between parenthesis) and F1, leading to the conclusion that arguments of some events are easier to learn (e.g., *selfdirectedbattle* vs. *transfortartifact*)

not always lead to better results. Figure 4 shows the average F1 for the top 15 most frequent events. The graph shows no clear pattern between event frequency in training and F1. Indeed, arguments of events with 2% frequency obtains F1-scores ranging from 0.33 and 0.60, a large range that overlaps with the F1-scores of more frequent events.

Which Arguments are Mislabeled? Our best model obtains 52.89 F1. This number is low, but the evaluation is strict: it expects predictions to match the exact text span and argument type. Figure 5 compares gold (rows) and predicted (columns) argument types when our best model (boldfaced in Table 3, large) predicts the correct argument spans. We observe two main trends. First, the model mislabels arguments with labels that are plausible at first sight but wrong given the input document. For example, *recipient*, *beneficiary*, and *giver* are often people but they have different semantics given an event trigger in context. Second, our model mislabels arguments that could be considered correct but do not follow the RAMS annotations. For example, the *transporter* of a *transporting* event (i.e., the person moving something) could be the *origin* or the event, but RAMS uses that argument type for the location where *transporting* started. We hypothesize that our model leverages the knowledge acquired about *transporter* and *origin* prior to our training with RAMS (and it never overcame this prior knowledge).



	place	recip	parti	commu	artif	victi	giver	targe	attac	benef	instr	trans	origi	desti	defen
place	92	0	1	0	0	0	0	4	0	0	0	0	0	0	0
recipient	8	71	0	2	2	0	0	0	0	16	0	0	0	0	0
participant	6	0	87	0	0	0	0	0	0	5	0	0	0	0	0
communicator	3	2	0	94	0	0	0	0	0	0	0	0	0	0	0
artifact	1	5	0	0	91	0	0	1	0	0	0	0	0	0	0
victim	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
giver	0	8	0	0	0	0	74	0	0	18	0	0	0	0	0
target	16	0	0	0	2	0	0	81	0	0	0	0	0	0	0
attacker	0	0	0	0	0	0	0	3	96	0	0	0	0	0	0
beneficiary	3	23	9	0	0	0	9	0	0	54	0	0	0	0	0
instrument	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
transporter	0	0	0	0	0	0	0	0	0	0	0	74	25	0	0
origin	0	0	0	0	0	0	0	0	0	0	0	21	73	5	0
destination	0	0	0	0	0	0	0	0	0	0	0	0	8	92	0
defendant	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure 5: Confusion matrix comparing gold (rows) and predicted (columns) argument types for correctly predicted argument spans (top 15 most frequent types). Most errors are plausible (at face value) but semantically wrong argument types (e.g., mislabeling the *beneficiary* as the *recipient*; note that both are usually people).

6. Error Analysis

We close the analyses examining the errors made by the best system (boldfaced in Table 3, large). We discuss linguistic commonalities in either the input documents or system predictions observed in a manual analysis of all the errors made in 100 documents (148 errors).

The majority of errors (38.51%, Table 6) are due to predicting *partial spans* (either shorter or longer than the gold). The differences include articles, conjunctions, numbers, and detailed descriptions complementing entities. Completely *wrong spans* are much less likely (6.76%). Despite we could not identify the underlying cause of all *wrong spans*, there are two common causes. First, the ground truth includes *one token span* per argument, but valid *alternatives* are sometimes present (13.51% of errors). In the example, our system predicts a coreferent mention that is counted as an error.

Error Type	Example
Partial spans (38.51%)	The Trump Wall, the past shows, does not promise a solution to the forces driving migration along [the [U.S.-Mexico border.]$^{\text{GOLD\_PLACE}}$]$^{\text{PREDICTED\_PLACE}}$ But it does offer the illusion of a solution. So if the Trump Wall is ever built, no one should be surprised when it is bypassed, breached or [bombarded]$^{\text{EVENT\_TRIGGER}}$, just like those that came before it.
Alternatives (13.51%)	[. . .] Then she gave an expansive denunciation of [Pakistan]$^{\text{PREDICTED\_JAILER}}$. Since its creation, [it]$^{\text{GOLD\_JAILER}}$ had [jailed]$^{\text{EVENT\_TRIGGER}}$ or exiled rival politicians. [. . .]
Distractors (4.05%)	[. . .] has published documents such as the probable-cause affidavit in a lieutenant's pain-pill addiction case, [purchase]$^{\text{EVENT\_TRIGGER}}$ orders showing that the [sheriff's office]$^{\text{GOLD\_GIVER}}$ spent more than \$60,000 [. . .] Now a technology consultant who regularly travels to Russia, [Dougan]$^{\text{PREDICTED\_GIVER}}$ says he made friends with hackers there and sold his website to them.
Wrong spans (6.76%)	[. . .] As a result, for the second time in four months the ratings agency S&P has downgraded Saudi Arabia's debt rating, which makes it more expensive for Saudi Arabia to borrow [money]$^{\text{PREDICTED\_RECIPIENT}}$. The [country]$^{\text{GOLD\_RECIPIENT}}$ is reportedly also asking banks for a [loan]$^{\text{EVENT\_TRIGGER}}$ of up to \$10 billion (£6.8 billion) [. . .]
Two or more arguments (2.70%)	On Wednesday's Breitbart News Daily, Sirius XM host [Alex Marlow]$^{\text{GOLD\_PARTICIPANT\_1}}$ $^{\text{predicted\_participant}}$ [discussed]$^{\text{EVENT\_TRIGGER}}$ leaked Hillary Clinton emails with [former Navy SEAL and Blackwater CEO Erik Prince.]$^{\text{GOLD\_PARTICIPANT\_2}}$

Table 6: Most common errors made by our best performing model (boldfaced in Table 3, large).

Second, *distractors* sometimes mislead the system. In the example, the system appears to confuse the event trigger (i.e., purchase) with a semantically similar but unrelated event: *sold his website*.

Our system is limited to predicting one span per argument type, thus it will always make errors with events that have two instances of the same argument type (2.70% of errors). A previously reported by Zhang et al. (2020b), we found that some errors (6.08%) appear to be due to annotation errors—no annotations are perfect, and RAMS is not an exception. For example, the test set includes the following: *he raised the [funds]*$^{\text{RECIPIENT}}$ *privately and [reimbursed]*$^{\text{EVENT\_TRIGGER}}$ *the city [. . .]*.

7. Conclusions

We have presented an approach to extract event-argument structures by automatically asking and answering questions. Our approach combines two complementary strategies to generate questions: template- and transformer-based. The latter not only generates noisy question-answer pairs, but also correct pairs involving events other than the event of interest. Yet, using transformer-based questions yields better results. Further, we explore several data augmentation strategies targeting inter-sentential arguments, as they are harder to identify. Transforming intra-sentential arguments into inter-sentential arguments by moving them to random sentence boundaries is the best strategy when experimenting with RoBERTa base. Indeed, it yields better results than leveraging coreference resolution or large language models (and compounding their errors).

Our transformer-based question generation combined with transfer learning outperforms previous work with RoBERTa large. Also, the data augmentation strategies helped the base model to achieve better or comparable results to (i.e., within a standard deviation) the top-2 best performing previous works. We use 11-14% less parameters, and, crucially, our model does not have any role-specific parameters. The lack of role-specific parameters has two advantages. It allows us to streamline transfer learning and make zero-shot predictions.

8. Limitations

The work presented in this paper has several limitations. Our model is limited to predict only one argument type per event trigger. Thus, any even trigger that has two arguments with the same argument type is guaranteed to yield an error (Section 6). Addressing this limitation requires further work, including multi-turn question answering.

Despite the empirical benefits of transformer-based questions, they are noisy and potentially nonsensical (Section 4.1.1). Our model learns from the noisy questions, but further work is needed to understand why and improve the generation process. We did a small-scale analysis but could not identify recurrent errors to improve the transformer-based question generation. While our best model shows improvement across all event-argument relations, the benefits are most noticeable in inter-sentential arguments. That said, results are still better with intra-sentential arguments. Further work is needed to extract event-argument structures from long documents.

9. Acknowledgments

This research was supported by a grant from the U.S. Office of Naval Research (N00014-22-1-2596). We would like to thank the anonymous reviewers for their insightful comments and suggestions.

10. Bibliographical References

Jacqueline Aguilar, Charley Beller, Paul McNamee, Benjamin Van Durme, Stephanie Strassel, Zhiyi Song, and Joe Ellis. 2014. A comparison of the events and relations across ACE, ERE, TAC-KBP, and FrameNet annotation standards. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 45–53, Baltimore, Maryland, USA. Association for Computational Linguistics.

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney, Australia. Association for Computational Linguistics.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Tom Brown, 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.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–419, Vancouver, Canada. Association for Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Yunmo Chen, Tongfei Chen, and Benjamin Van Durme. 2020. Joint modeling of arguments for event understanding. In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 96–101, Online. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas. Association for 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).

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, 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, Online. Association for Computational Linguistics.

Jun Gao, Changlong Yu, Wei Wang, Huan Zhao, and Ruifeng Xu. 2022. Mask-then-fill: A flexible and effective data augmentation framework for event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4537–4544, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Matthew Gerber and Joyce Chai. 2010. Beyond NomBank: A study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1583–1592, Uppsala, Sweden. Association for Computational Linguistics.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Yu Hong, 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, Portland, Oregon, USA. Association for Computational Linguistics.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. Voss, Jiawei Han, and Avirup Sil. 2016. Liberal event extraction and event schema induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 258–268, Berlin, Germany. Association for Computational Linguistics.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.

Yin Jou Huang, Jing Lu, Sadao Kurohashi, and Vincent Ng. 2019. Improving event coreference resolution by learning argument compatibility from unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 785–795, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 829–838, Online. Association for Computational Linguistics.

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.

Wei Li, Lei He, and Hai Zhuge. 2016. Abstractive news summarization based on event semantic link network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 236–246, Osaka, Japan. The COLING 2016 Organizing Committee.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797, Uppsala, Sweden. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online. Association for Computational Linguistics.

Jian Liu, Yufeng Chen, and Jinan Xu. 2021. Machine reading comprehension as data augmentation: A case study on implicit event argument extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2725, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yinhan Liu, 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.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332, Online. Association for Computational Linguistics.

Yubo Ma, 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. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774, Dublin, Ireland. Association for Computational Linguistics.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. Annotating noun argument structure for NomBank. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

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.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Alan Ramponi, Rob van der Goot, Rosario Lombardo, and Barbara Plank. 2020. Biomedical event extraction as sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5357–5367, Online. Association for Computational Linguistics.

Josef Ruppenhofer, 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, Uppsala, Sweden. Association for Computational Linguistics.

Oscar Sainz, Itziar Gonzalez-Dios, Oier Lopez de Lacalle, Bonan Min, and Eneko Agirre. 2022. Textual entailment for event argument extraction: Zero- and few-shot with multi-source learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2439–2455, Seattle, United States. Association for Computational Linguistics.

Eyal Shnarch, Carlos Alzate, Lena Dankin, Martin Gleize, Yufang Hou, Leshem Choshen, Ranit Aharonov, and Noam Slonim. 2018. Will it blend? blending weak and strong labeled data in a neural network for argumentation mining. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 599–605, Melbourne, Australia. Association for Computational Linguistics.

Liuyin Wang, Zihan Xu, Zibo Lin, Haitao Zheng, and Ying Shen. 2020. Answer-driven deep question generation based on reinforcement learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5159–5170, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu

Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Kaiwen Wei, 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, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Runxin Xu, Peiyi Wang, Tianyu Liu, Shuang Zeng, Baobao Chang, and Zhifang Sui. 2022. A two-stream AMR-enhanced model for document-level event argument extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5025–5036, Seattle, United States. Association for Computational Linguistics.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018. DCFEE: A document-level Chinese financial event extraction system based on automatically labeled training data. In *Proceedings of ACL 2018, System Demonstrations*, pages 50–55, Melbourne, Australia. Association for Computational Linguistics.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. GPT3Mix: Leveraging large-scale language models for text augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2225–2239, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Tongtao Zhang, Hongzhi Li, Heng Ji, and Shih-Fu Chang. 2015. Cross-document event coreference resolution based on cross-media features. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 201–206, Lisbon, Portugal. Association for Computational Linguistics.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020b. A two-step approach for implicit event argument detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7479–7485, Online. Association for Computational Linguistics.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. Transfer learning from semantic role labeling to event argument extraction with template-based slot querying. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2627–2647, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jianing Zhou and Suma Bhat. 2021. Paraphrase generation: A survey of the state of the art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5086, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

11. Language Resource References

Ebner, Seth and Xia, Patrick and Culkin, Ryan and Rawlins, Kyle and Van Durme, Benjamin. 2020. *Multi-Sentence Argument Linking*. Association for Computational Linguistics. PID https://nlp.jhu.edu/rams/.

He, Luheng and Lewis, Mike and Zettlemoyer, Luke. 2015. *Question-Answer Driven Semantic Role Labeling: Using Natural Language to Annotate Natural Language*. Association for Computational Linguistics. PID https://dada.cs.washington.edu/qasrl/.

Li, Sha and Ji, Heng and Han, Jiawei. 2021. *Document-Level Event Argument Extraction by Conditional Generation*. Association for Computational Linguistics. PID https://github.com/raspberryice/gen-arg.

Walker, Christopher and Strassel, Stephanie and Medero, Julie and Maeda, Kazuaki. 2006. *Ace 2005 multilingual training corpus*. ISLRN 458-031-085-383-4.

A. Zero- and Few-Shot Prompts

To conduct our experiments with GPT-3 in zero and few-shot settings, we utilized the OpenAI API.[4] Specifically, we use the ‘text-davinci-003’ model and the ‘Completion’ endpoint provided by the API.

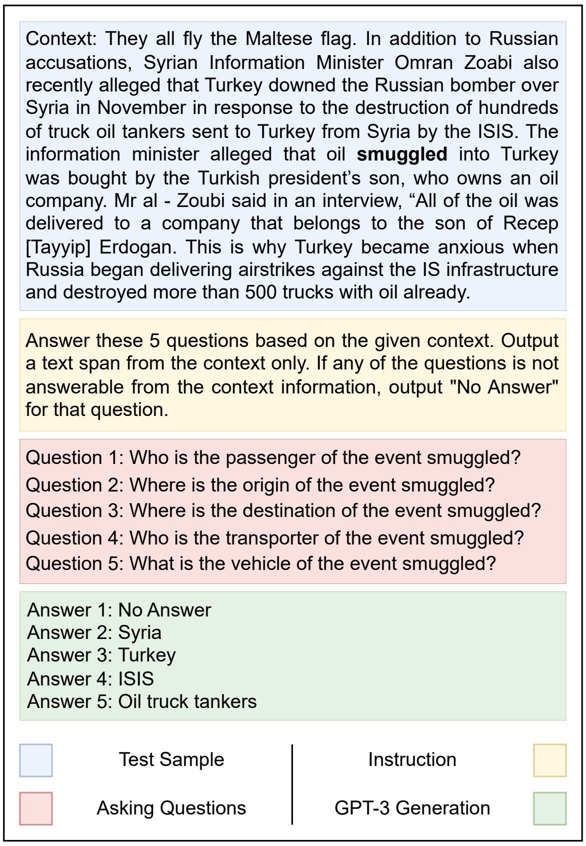
To ensure consistency with the inputs used in the supervised model, we designed the prompts for the GPT-3 model in a similar manner. However, there were slight differences in how the prompts were handled between the supervised settings and the zero and few-shot experiments. In the supervised settings, we posed one question per iteration, whereas, for the GPT-3 zero-shot and few-shot experiments, we included all the questions simultaneously. We conducted a small-scale study using a subset of samples from the RAMS dataset to validate the impact of asking all questions at once compared to asking one question per iteration. Our study did not reveal any difference between the two approaches. Hence, we proceeded with asking all questions at once for the zero and few-shot experiments. This streamlined the experimentation process and also helped to reduce the costs of querying the API.

A.1. Example of Zero-Shot Prompt

In the zero-shot setting, our objective is to extract event arguments without any training examples. To accomplish this, we construct prompts with template-based questions. The GPT-3 model generates answers to these questions, which are then mapped back to the provided document to extract matched event argument spans. Figure 6 presents a snapshot of a zero-shot prompt.

A.2. Example of Few-Shot Prompt

In the few-shot setting, we leverage a limited amount of training data. We randomly select two training samples from the RAMS dataset. These examples are formatted to match the inputs used during supervised training. By incorporating two training samples, we enhance the model’s ability to capture event arguments and generate accurate responses. Figure 7 presents a snapshot of a few-shot prompt.



Context: They all fly the Maltese flag. In addition to Russian accusations, Syrian Information Minister Omran Zoabi also recently alleged that Turkey downed the Russian bomber over Syria in November in response to the destruction of hundreds of truck oil tankers sent to Turkey from Syria by the ISIS. The information minister alleged that oil **smuggled** into Turkey was bought by the Turkish president’s son, who owns an oil company. Mr al - Zoubi said in an interview, “All of the oil was delivered to a company that belongs to the son of Recep [Tayyip] Erdogan. This is why Turkey became anxious when Russia began delivering airstrikes against the IS infrastructure and destroyed more than 500 trucks with oil already.

Answer these 5 questions based on the given context. Output a text span from the context only. If any of the questions is not answerable from the context information, output "No Answer" for that question.

Question 1: Who is the passenger of the event smuggled?
Question 2: Where is the origin of the event smuggled?
Question 3: Where is the destination of the event smuggled?
Question 4: Who is the transporter of the event smuggled?
Question 5: What is the vehicle of the event smuggled?

Answer 1: No Answer
Answer 2: Syria
Answer 3: Turkey
Answer 4: ISIS
Answer 5: Oil truck tankers

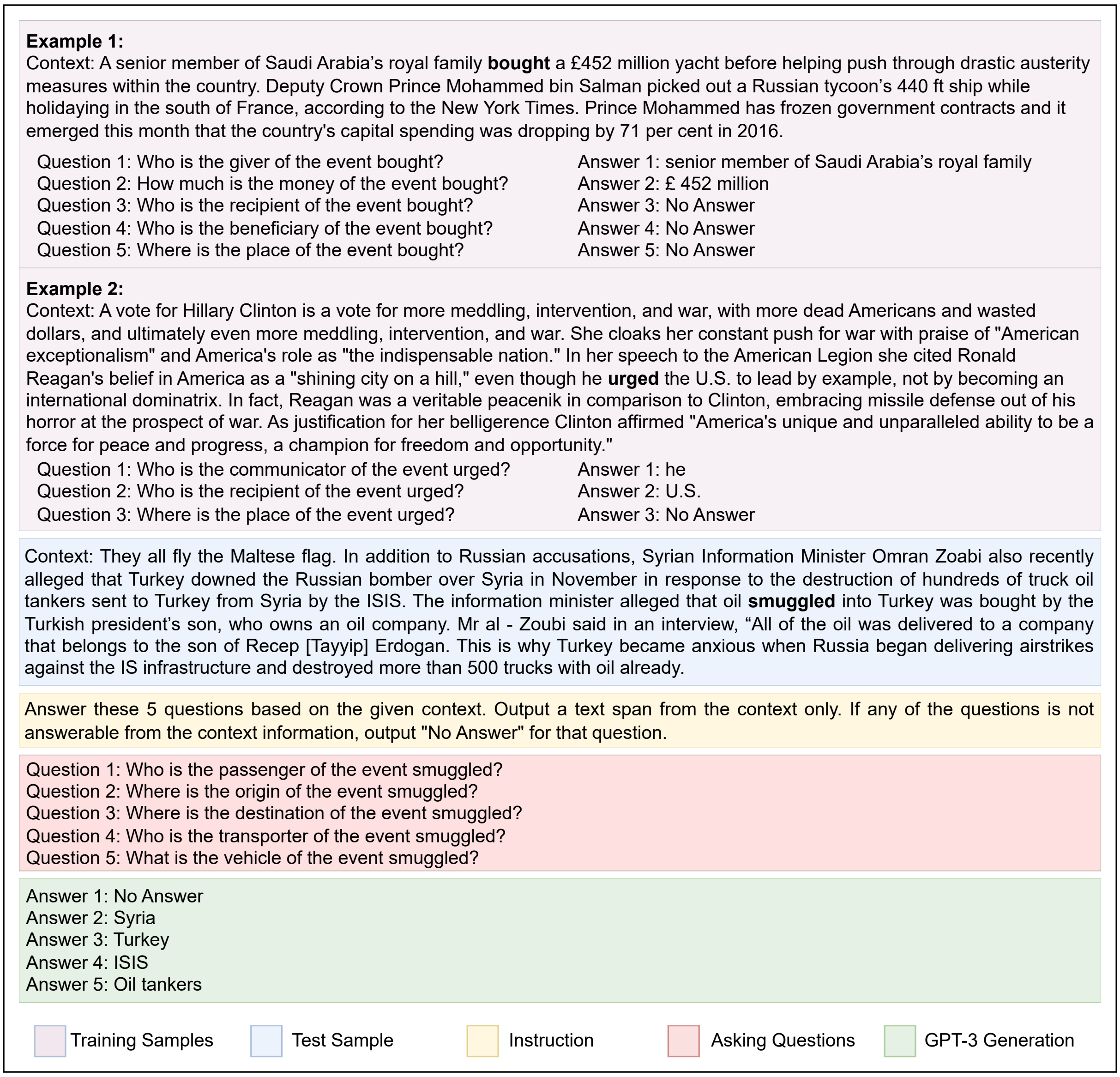
Figure 6: Example of a zero-shot GPT-3 prompt. *Test sample*, *Instruction* and *Asking Questions* are all together considered the prompt (*input*) and *GPT-3 Generation* is the output.

B. QA Model and Hyperparameters

In this section, we provide an overview of the model and hyperparameters used in the supervised approach for event argument extraction. We leveraged the RoBERTa-base and large models to generate contextual representations for the question and document pairs. By feeding the question and document as input to the RoBERTa model, we obtained the contextualized representation of the combined text. Then, we employ a task-specific layer that operates on top of these representations. This layer is responsible for predicting the start and end offsets of the argument span.

During training, we utilize annotated data samples where the ground truth start and end offsets of the argument span are provided. The output layer is trained using the cross-entropy loss function to minimize the discrepancy between the predicted offsets and the ground truth offsets. We conducted experiments using three different learning rates [2e-5, 3e-5, 5e-5] and dropout values [0.3, 0.4, 0.5] to optimize the performance of the models. In order to determine the optimal hyperparameters,

[4] OpenAI



Example 1:
Context: A senior member of Saudi Arabia’s royal family **bought** a £452 million yacht before helping push through drastic austerity measures within the country. Deputy Crown Prince Mohammed bin Salman picked out a Russian tycoon’s 440 ft ship while holidaying in the south of France, according to the New York Times. Prince Mohammed has frozen government contracts and it emerged this month that the country's capital spending was dropping by 71 per cent in 2016.

Question 1: Who is the giver of the event bought?	Answer 1: senior member of Saudi Arabia’s royal family
Question 2: How much is the money of the event bought?	Answer 2: £ 452 million
Question 3: Who is the recipient of the event bought?	Answer 3: No Answer
Question 4: Who is the beneficiary of the event bought?	Answer 4: No Answer
Question 5: Where is the place of the event bought?	Answer 5: No Answer

Example 2:
Context: A vote for Hillary Clinton is a vote for more meddling, intervention, and war, with more dead Americans and wasted dollars, and ultimately even more meddling, intervention, and war. She cloaks her constant push for war with praise of "American exceptionalism" and America's role as "the indispensable nation." In her speech to the American Legion she cited Ronald Reagan's belief in America as a "shining city on a hill," even though he **urged** the U.S. to lead by example, not by becoming an international dominatrix. In fact, Reagan was a veritable peacenik in comparison to Clinton, embracing missile defense out of his horror at the prospect of war. As justification for her belligerence Clinton affirmed "America's unique and unparalleled ability to be a force for peace and progress, a champion for freedom and opportunity."

Question 1: Who is the communicator of the event urged?	Answer 1: he
Question 2: Who is the recipient of the event urged?	Answer 2: U.S.
Question 3: Where is the place of the event urged?	Answer 3: No Answer

Context: They all fly the Maltese flag. In addition to Russian accusations, Syrian Information Minister Omran Zoabi also recently alleged that Turkey downed the Russian bomber over Syria in November in response to the destruction of hundreds of truck oil tankers sent to Turkey from Syria by the ISIS. The information minister alleged that oil **smuggled** into Turkey was bought by the Turkish president’s son, who owns an oil company. Mr al - Zoubi said in an interview, “All of the oil was delivered to a company that belongs to the son of Recep [Tayyip] Erdogan. This is why Turkey became anxious when Russia began delivering airstrikes against the IS infrastructure and destroyed more than 500 trucks with oil already.

Answer these 5 questions based on the given context. Output a text span from the context only. If any of the questions is not answerable from the context information, output "No Answer" for that question.

Question 1: Who is the passenger of the event smuggled?
Question 2: Where is the origin of the event smuggled?
Question 3: Where is the destination of the event smuggled?
Question 4: Who is the transporter of the event smuggled?
Question 5: What is the vehicle of the event smuggled?

Answer 1: No Answer
Answer 2: Syria
Answer 3: Turkey
Answer 4: ISIS
Answer 5: Oil tankers

🟪 Training Samples 🟦 Test Sample 🟨 Instruction 🟥 Asking Questions 🟩 GPT-3 Generation

Figure 7: Example of a few-shot GPT-3 prompt. *Training samples*, *Test sample*, *Instruction* and *Asking Questions* are all together considered as prompt (*input*) and *GPT-3 generation* is the output.

we evaluated their performance on the validation dataset. This evaluation allowed us to select the final hyperparameters that yielded the best results. We list all the hyperparameters in Table 7.

To mitigate the risk of overfitting and ensure efficient training, we incorporated the technique of early stopping. If the loss function fails to show improvement over 10 consecutive epochs, training is terminated before completing 200 epochs.

At inference time, given a new question and document pair, the trained model applies the learned weights and biases to predict the most likely start and end offsets of the argument span. These predicted offsets indicate the span within the document where the argument is expected to be found.

Name	Value
learning rate	2e-5, 3e-5, 5e-5
number of epochs	200
patience	10
dropout	0.3, 0.4, 0.5
training batch size	8
validation batch size	4
test batch size	4
max length	512
loss function	cross-entropy
optimizer	Adam
blending factor (α)	0.2, 0.4, 0.6

Table 7: Hyperparameters of the supervised models trained on RAMS.

C. Data Augmentation Examples

In this section, we demonstrate the three data augmentation strategies. The goal of these data augmentation strategies is to transform intra-sentential arguments into inter-sentential arguments while generating new training instances. For demonstration purposes, we select a data sample from RAMS, featuring the event triggering word *agreement* and two event arguments: *Clinton* (violator) and *Iran* (otherparticipant).

C.1. Simple Swapping

The simple swapping strategy involves shifting an intra-sentential argument to different positions (at the beginning or end position of each sentence), transforming it into an inter-sentential argument. The given example in Table 8 has 5 sentences, so that leaves 6 different places to shift the argument. We consider the end position of the ith sentence and the beginning position of the $(i+1)$th sentence as the same position. Then, one position is chosen randomly from these 5 positions to replace the corresponding gold annotation. The *verbose* version of the simple swapping approach follows the same procedure for determining the new position of the argument. However, we replace the argument with a simple sentence such as *"The violator of the event agreement is Clinton."* Also, we keep the original and the augmented argument in the document whereas we discard the original argument for the simple swapping (*Plain*). It is worth noting that both versions generate grammatically incorrect sentences, but we focused on generating argument spans that are inter-sentential.

C.2. Leveraging Coreference Resolution

In the coreference-based data augmentation strategy, the first step involves identifying the coreference chains related to the given argument. In Table 8, the sample exhibits two coreference chains corresponding to the arguments. These chains are extracted using the spaCy library.[5] For the argument *Clinton*, the coreference chain appears as *Hillary Clinton: [Hillary Clinton (sent 1), she (sent 2), Clinton (sent 3)]*. Similarly, for the argument *Iran*, the coreference chain is *[Iran (sent 2), Iran (sent 3), its (sent 3), the country (sent 3), Iran (sent 4)].* To generate augmented data using these coreference chains, we randomly select a mention from the coreference chain to replace the corresponding gold annotation. Alternatively, for the *most meaningful* mention, we prioritize the selection of the argument with the highest number of tokens and named entities, such as choosing *Hillary Clinton* instead of *Clinton* or *She*.

C.3. Leveraging LLMs for Paraphrasing

To leverage Large Language Models (LLMs) for paraphrasing the RAMS samples, we employed both sentence-level and document-level paraphrasing techniques. Upon examining the examples from Table 9, we observed that sentence-level paraphrasing did not facilitate moving the intra-sentential arguments to inter-sentential arguments. This is because we could only provide one sentence as an input to the paraphraser model. However, using the GPT-3 model with the prompt shown in Section 4.1.2 generated samples with more inter-sentential arguments. In Table 9, both intra-sentential arguments (*Clinton and Iran*) from the original sample successfully shifted to inter-sentential arguments. This transition of arguments across sentence boundaries demonstrates the potential of LLMs for enhancing the diversity of training data in natural language processing tasks.

[5] spaCy

Original from RAMS	As Secretary of State, Hillary Clinton was a hawk on the Iranian nuclear issue. In 2009 - 2010, when Iran first indicated a willingness to compromise, she led the opposition to any negotiated settlement and pushed for punishing sanctions. To clear the route for sanctions, [Clinton]$^{\text{VIOLATOR}}$ helped sink [agreements]$^{\text{EVENT}}$ tentatively negotiated with [Iran]$^{\text{OTHERPARTICIPANT}}$ to ship most of its low - enriched uranium out of the country. In 2009, Iran was refining uranium only to the level of about 3-4 percent, as needed for energy production. Its negotiators offered to swap much of that for nuclear isotopes for medical research.
Augmented instances	
Simple Swapping	
Plain	[Clinton]$^{\text{VIOLATOR}}$ As Secretary of State, Hillary Clinton was a hawk on the Iranian nuclear issue. In 2009 - 2010, when Iran first indicated a willingness to compromise, she led the opposition to any negotiated settlement and pushed for punishing sanctions. To clear the route for sanctions, helped sink [agreements]$^{\text{EVENT}}$ tentatively negotiated with [Iran]$^{\text{OTHERPARTICIPANT}}$ to ship most of its low - enriched uranium out of the country. In 2009, Iran was refining uranium only to the level of about 3-4 percent, as needed for energy production. Its negotiators offered to swap much of that for nuclear isotopes for medical research.
Verbose	The violator of the event agreements is [Clinton]$^{\text{VIOLATOR}}$. As Secretary of State, Hillary Clinton was a hawk on the Iranian nuclear issue. In 2009 - 2010, when Iran first indicated a willingness to compromise, she led the opposition to any negotiated settlement and pushed for punishing sanctions. To clear the route for sanctions, helped sink [agreements]$^{\text{EVENT}}$ tentatively negotiated with [Iran]$^{\text{OTHERPARTICIPANT}}$ to ship most of its low - enriched uranium out of the country. In 2009, Iran was refining uranium only to the level of about 3-4 percent, as needed for energy production. Its negotiators offered to swap much of that for nuclear isotopes for medical research.
Leveraging Coreference	
Random mention	As Secretary of State, Hillary Clinton was a hawk on the Iranian nuclear issue. In 2009 - 2010, when Iran first indicated a willingness to compromise, [she]$^{\text{VIOLATOR}}$ led the opposition to any negotiated settlement and pushed for punishing sanctions. To clear the route for sanctions, Clinton helped sink [agreements]$^{\text{EVENT}}$ tentatively negotiated with [Iran]$^{\text{OTHERPARTICIPANT}}$ to ship most of its low - enriched uranium out of the country. In 2009, Iran was refining uranium only to the level of about 3-4 percent, as needed for energy production. Its negotiators offered to swap much of that for nuclear isotopes for medical research.
Meaningful mention	As Secretary of State, [Hillary Clinton]$^{\text{VIOLATOR}}$ was a hawk on the Iranian nuclear issue. In 2009 - 2010, when Iran first indicated a willingness to compromise, she led the opposition to any negotiated settlement and pushed for punishing sanctions. To clear the route for sanctions, Clinton helped sink [agreements]$^{\text{EVENT}}$ tentatively negotiated with [Iran]$^{\text{OTHERPARTICIPANT}}$ to ship most of its low - enriched uranium out of the country. In 2009, Iran was refining uranium only to the level of about 3-4 percent, as needed for energy production. Its negotiators offered to swap much of that for nuclear isotopes for medical research.

Table 8: Original instance from RAMS and additional instances obtained with our data augmentation strategies (*Simple Swapping* and *Leveraging Coreference*). See descriptions of the strategies in Section 4.1.2). While *Simple Swapping* results in ungrammatical text as it shuffles intra-sentential arguments, it is the strategy that yields the best results (Table 2). Leveraging coreference introduces errors when the predicted coreference chains are wrong.

Original from RAMS	As Secretary of State, Hillary Clinton was a hawk on the Iranian nuclear issue. In 2009 - 2010, when Iran first indicated a willingness to compromise, she led the opposition to any negotiated settlement and pushed for punishing sanctions. To clear the route for sanctions, [Clinton]$^{\text{VIOLATOR}}$ helped sink [agreements]$^{\text{EVENT}}$ tentatively negotiated with [Iran]$^{\text{OTHERPARTICIPANT}}$ to ship most of its low - enriched uranium out of the country. In 2009, Iran was refining uranium only to the level of about 3-4 percent, as needed for energy production. Its negotiators offered to swap much of that for nuclear isotopes for medical research.
Augmented instances	
Leveraging LLMs	
Pegasus	Hillary Clinton was a strong supporter of the Iranian nuclear issue as Secretary of State. She led the opposition to any negotiated settlement and pushed for punishing sanctions after Iran indicated a willingness to compromise. [Clinton]$^{\text{VIOLATOR}}$ helped sink [agreements]$^{\text{EVENT}}$ that would have allowed [Iran]$^{\text{OTHERPARTICIPANT}}$ to ship most of its low - enriched uranium out of the country. Iran only refined about 3 - 4 percent of its nuclear material in 2009, as needed for energy production. Much of that was offered to be used for medical research.
GPT3	Secretary of State Hillary Clinton has taken a hard line against Iran's nuclear ambitions, recently thwarting [agreements]$^{\text{EVENT}}$ tentatively negotiated between the two countries. [Clinton]$^{\text{VIOLATOR}}$ has pushed for punishing sanctions, as she argued that any negotiated settlement was not enough to ensure [Iran]$^{\text{OTHERPARTICIPANT}}$ would not pursue nuclear weapons. Her opposition to the agreements was based on Iran's offer to swap its low-enriched uranium for nuclear isotopes, which she felt was not sufficient to prevent their potential proliferation. Iran initially sought to refine the uranium to the levels necessary for energy production, but Clinton insisted that the sanctions remain in place. As the situation between Iran and the United States continues to evolve, Clinton's hard-line stance on the Iranian nuclear issue remains.

Table 9: Original instance from RAMS and additional instances obtained with our data augmentation strategies (*Leveraging LLMs*). See descriptions of the strategies in (Section 4.1.2).