Document-Level Event Argument Extraction With A Chain Reasoning Paradigm

Jian Liu[1]*, Chen Liang[1]*, Jinan Xu[1], Haoyan Liu[2] and Zhe Zhao[2]
[1] Beijing Jiaotong University [2] Tencent AI Lab
{jianliu, 21120367, jaxu}@bjtu.edu.cn
{haoyanliu, nlpzhezhao}@tencent.com

Abstract

Document-level event argument extraction aims to identify event arguments beyond sentence level, where a significant challenge is to model long-range dependencies. Focusing on this challenge, we present a new chain reasoning paradigm for the task, which can generate decomposable first-order logic rules for reasoning. This paradigm naturally captures long-range interdependence due to the chains' compositional nature, which also improves interpretability by explicitly modeling the reasoning process. We introduce T-norm fuzzy logic for optimization, which permits end-to-end learning and shows promise for integrating the expressiveness of logical reasoning with the generalization of neural networks. In experiments, we show that our approach outperforms previous methods by a significant margin on two standard benchmarks (over 6 points in F1). Moreover, it is data-efficient in low-resource scenarios and robust enough to defend against adversarial attacks.

1 Introduction

Identifying event arguments (i.e., participants of an event) is a crucial task for document-level event understanding (Ebner et al., 2020; Li et al., 2021). In this task, the major challenge is to model long-range dependencies between event triggers and arguments, as an event expression can span multiple sentences (Ebner et al., 2020; Liu et al., 2021; Li et al., 2021). Consider the event expressed by a trigger *detonated* (type=`Attack`) in Figure 1. To locate its argument *Tartus* (semantic role=`Place`), a model should capture a large context window of three sentences and 178 words to support the reasoning process.

Currently, it still remains an open problem for effectively capturing such dependencies (Liu et al., 2021, 2022c). Prior research has proposed to model beyond-sentence clues by incorporating hierarchical encoding mechanisms (Du and Cardie, 2020a), generative paradigms (Li et al., 2021; Ma et al., 2022; Du et al., 2022), and document-level inductive bias (Wei et al., 2021; Pouran Ben Veyseh et al., 2022; Liu et al., 2022b). Nevertheless, such methods do not explicitly characterize the reasoning patterns underlying the document context, which potentially suffers sub-optimal performance. In addition, most previous methods are not interpretable because they rely on black-box neural networks.

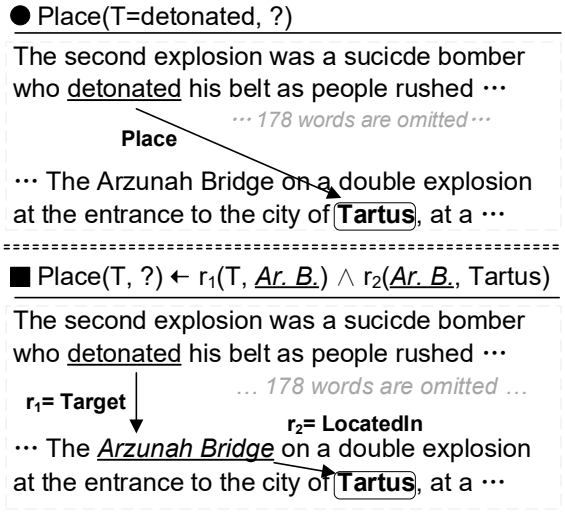


Figure 1: Illustration of the document-level EAE task (●) and our chain-of-reasoning paradigm (■).

In this paper, we propose a new chain-of-reasoning paradigm to address document-level event argument extraction (EAE). As indicated at the bottom of Figure 1, our method seeks to describe the global argument-finding process via a chain of local inference steps. For example, we may use the following chain to locate *Tartus*: *detonated* $\xrightarrow{\texttt{Target}}$ *Arzunah Bridge* $\xrightarrow{\texttt{LocatedIn}}$ *Tartus*. This chain-of-reasoning paradigm has three clear benefits over previous approaches: First, it naturally captures long-distance dependencies owing to the compositional structure of the reasoning chain.

*Equal contribution.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 9570–9583

Second, it involves only local reasoning, which is conceptually easier than performing global reasoning directly. Third, it improves interpretability as the reasoning processes are visible.

Our approach formalizes the reasoning chain as first-order logic (FOL) rules (Cresswell and Hughes, 1996). Concretely, let $\texttt{RL}(T, ?)$ be the query for an event argument fulfilling the semantic role $\texttt{RL}$ (e.g., $\texttt{Place}$) regarding an event trigger T. We formalize the query as the following FOL rule:

$$\texttt{RL}(T, ?) \leftarrow r_1(T, B_1) \wedge ... \wedge r_n(B_{n-1}, ?)$$

where the body of the rule (on the right) consists of conjunctive propositions with low-level predicates $\{r_i\}_1^n$ and intermediary clue entities $\{B_i\}_1^{n-1}$. We build a model to automatically generate the rule based on the document context, and then transform the rule into a reasoning chain to locate the event argument. Nevertheless, it is generally challenging to optimize with FOL rules owing to their discrete nature (Qu et al., 2021a). Inspired by work that augments neural networks with FOLs (Li and Srikumar, 2019; Ahmed et al., 2022), we present T-Norm fuzzy logic for relaxation (Hajek, 1998), which leads to an end-to-end training regime.

We verify the effectiveness of our method on two benchmarks (Ebner et al., 2020; Li et al., 2021). According to the results, our approach delivers promising results with this chain reasoning paradigm, such as yielding a 6-point improvement in F1 over models trained using large-scale external resources (§ 6.1). Interestingly, in addition to the performance boost, our approach demonstrates decent robustness, particularly in low-resource scenarios and defending against adversarial noises (§ 7.2). Lastly, we evaluate the interpretability of our methodology using a thorough case study (§ 7.3).

In conclusion, our contributions are three-fold:

- We introduce a new chain-of-reasoning paradigm for document-level EAE, demonstrating clear advantages in capturing long-range dependencies and enhancing interpretability. As a seminal study, our work may motivate more studies in this research line.
- We introduce T-Norm fuzzy logic, which relaxes discrete FOL rules for document-level EAE into differentiable forms; it also demonstrates the prospect of combining the expressiveness of logical reasoning with the generalization capabilities of neural networks.
- We report state-of-the-art performance on two benchmarks, and we have made our code available[1] for future exploration.

2 Related Work

Document-Level EAE. Extracting event arguments in a document context is a vital step in document-level event extraction (Grishman, 2019; Ebner et al., 2020). Earlier efforts on this problem explore the MUC-4 benchmark (Chinchor, 1991; Huang and Riloff, 2012), also known as "template filling" because the entire document is about one event. Recent research has focused on events with lexical triggers, intending to extract all arguments for a trigger-indicated event (Ebner et al., 2020; Li and Srikumar, 2019). For capturing the document context effectively, prior studies have explored hierarchical encoding mechanisms, generative perspectives (Li et al., 2021; Du et al., 2022; Ma et al., 2022), document-level inductive biases (Wei et al., 2021; Pouran Ben Veyseh et al., 2022), and external resources (Du and Cardie, 2020b; Liu et al., 2020; Xu et al., 2022; Liu et al., 2022a). Nonetheless, such methods do not explicitly model the underlying reasoning process for capturing long-range dependencies, which therefore risks achieving suboptimal performance. In addition, these methods are not interpretable because they employ neural networks with black-box architectures. In contrast to the previous study, we investigate employing a chain-of-reasoning paradigm to explain the reasoning process, which can effectively model long-range context while retaining interpretability.

Reasoning with FOL Rules. First-order logic (FOL) rules can encode declarative knowledge and play a crucial role in symbolic reasoning (Cresswell and Hughes, 1996). In the era of deep learning, several studies have examined the integration of FOL rules with neural networks for reasoning (termed neural-symbolic approaches), with applications in knowledge base inference (Qu et al., 2021b), text entailment (Li and Srikumar, 2019), question answering (Wang and Pan, 2022), and others (Medina et al., 2021; Ahmed et al., 2022). Our approach is inspired by the work on knowledge base inference, which, to the best of our knowledge, is the first attempt to incorporate FOL rules for reasoning in the context of document-level EAE. Compared to other methods, we investigate the prospect of

[1] https://github.com/jianliu-ml/logicEAE

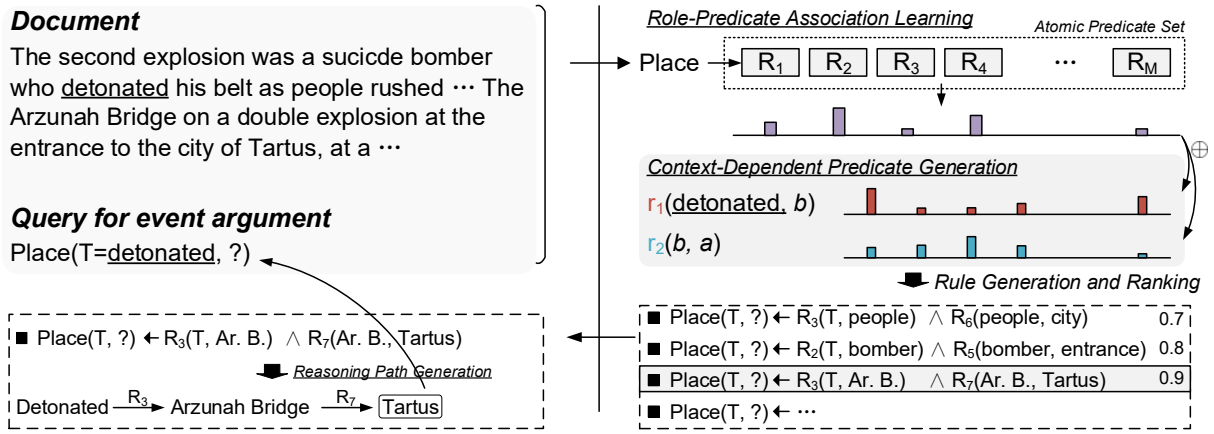


Figure 2: The overview of our approach, with a running example for extracting the argument of a `Place` role for the event trigger detonated. b and a indicate a particular clue entity (in $\mathcal{B}$) and candidate argument (in $\mathcal{A}$).

generating rules using neural networks automatically instead of employing expert-written rules as in (Li and Srikumar, 2019; Wang and Pan, 2022). Additionally, unlike those based on reinforcement learning (Qu et al., 2021b), we use T-norms for rule relaxation, resulting in an end-to-end training paradigm with a more stable learning procedure.

3 Approach

Figure 2 presents the overview of our approach, with an example for extracting the argument of a `Place` role for the event *detonated*. Let $D = \{w_1, \cdots, T, \cdots, w_N\}$ be a document with N words and an event trigger T, and let $\texttt{RL}(T, ?)$ be a query for the event argument of a semantic role `RL`. Instead of directly performing the reasoning that may involve high-level processes, our approach represents the query as a FOL rule with conjunctive propositions and low-level predicates $\{r_i\}_1^n$:

$$\texttt{RL}(T, ?) \leftarrow r_1(T, B_1) \wedge ... \wedge r_n(B_{n-1}, ?)$$

In this way, the body of the rule suggests a reasoning chain: $T \xrightarrow{r_1} B_1 \xrightarrow{r_2} \cdots B_{n-1} \xrightarrow{r_n} ?$. We utilize a two-predicate formulation, specifically $\texttt{RL}(T, ?) \leftarrow r_1(T, B) \wedge r_2(B, ?)$, to explain our method, and we describe general cases in § 4.

3.1 Clue Entity Set Generation

In the first step of our method, we create a set of entities from which one may be chosen as an intermediary clue entity to form the reasoning chain (regarding our two-predicate structure). We broaden the notion of "entity" to include any single word in the document for incorporating verb-based cues. To limit the size of the set, we give each word a score derived from BERT representations (Devlin et al., 2019). For example, the score for w_i is:

$$s_{w_i} = \frac{\exp(\boldsymbol{w}_s^T \boldsymbol{h}_{w_i} + b_s)}{\sum_j \exp(\boldsymbol{w}_s^T \boldsymbol{h}_{w_j} + b_s)} \quad (1)$$

where $\boldsymbol{h}_{w_i}$ is the representation of w_i, and $\boldsymbol{w}_s$ and b_s are model parameters. We rank all words based on the sores and select K with the highest sores to form the set, denoted by $\mathcal{B} = \{b_i\}_{i=1}^K$.

To facilitate training and testing, we additionally generate an argument candidate set. In this case, we do not utilize the broad definition of entity because an event argument is defined to be a noun entity (Walker and Consortium, 2005; Ahn, 2006). When ground-truth entities are available (such as in WikiEvents (Li et al., 2021)), we consider the candidate set to be the ground-truth entity set; otherwise, we use external toolkits[2] to recognize entities. We denote the argument candidate set by $\mathcal{A} = \{a_i\}_{i=1}^L$.

3.2 FOL Rule Generation

Given the entity candidate set $\mathcal{B}$ and the argument candidate set $\mathcal{A}$, the next step is to generate two predicates and select related candidates in the sets to form the rule. Here we explain our method for generating predicates regarding a particular entity-argument pair ($B \in \mathcal{B}$, $A \in \mathcal{A}$), and we show metrics for ranking the rules generated by different candidate pairs in § 4.

Predicate Representations. In our approach, we assume that there are M atomic predicates with

[2]We use spacy (https://github.com/explosion/spaCy) with default settings as entity recognizer.

indecomposable semantics, represented by a predicate set $\mathcal{R} = \{R_i\}_{i=1}^{M}$. We give each predicate a d-dimensional vectorized representation and derive a matrix representation $\boldsymbol{U} \in \mathbb{R}^{M\times d}$ for $\mathcal{R}$. For the semantic role $\texttt{RL}$, we also give it a d-dimensional representation, indicated by $\boldsymbol{r}_{\texttt{RL}} \in \mathbb{R}^d$.

Learning Role-Predicate Associations. Given the representations, we first learn a role-to-predicate association that indicates which predicates are likely to be generated based on the role solely and disregarding context. We employ auto-regressive learning and generate a probability vector $\boldsymbol{a}_{\texttt{RL}}^{(1)} \in \mathbb{R}^M$ indicating the distribution of the first predicate r_1 over the predicate set $\mathcal{R}$:

$$\boldsymbol{a}_{\texttt{RL}}^{(1)} = \text{softmax}(\boldsymbol{U}\boldsymbol{W}_s^{(1)}\boldsymbol{r}_{\texttt{RL}}) \tag{2}$$

where $\boldsymbol{W}_s^{(1)} \in \mathbb{R}^{d\times d}$ is a parameter. To learn the distribution of the second predicate r_2, we first update the role's representation by integrating the impact of the first predicate:

$$\boldsymbol{r}_{\texttt{RL}}^{(1)} = \boldsymbol{r}_{\texttt{RL}} + \boldsymbol{a}_{\texttt{RL}}^{(1)}\boldsymbol{W}_a^{(1)}\boldsymbol{U} \tag{3}$$

and then compute a probability vector $\boldsymbol{a}_{\texttt{RL}}^{(2)} \in \mathbb{R}^M$:

$$\boldsymbol{a}_{\texttt{RL}}^{(2)} = \text{softmax}(\boldsymbol{U}\boldsymbol{W}_s^{(2)}\boldsymbol{r}_{\texttt{RL}}^{(1)}) \tag{4}$$

where $\boldsymbol{W}_a^{(1)} \in \mathbb{R}^{M\times d}$ and $\boldsymbol{W}_s^{(2)} \in \mathbb{R}^{d\times d}$ are parameters to learn. We can set r_1 and r_2 as predicates with the highest probability in $\boldsymbol{a}_{\texttt{RL}}^{(1)}$ and $\boldsymbol{a}_{\texttt{RL}}^{(2)}$, respectively. However, such an approach always generates the same predicates for a semantic role and has a pretty poor performance (7.1). As a solution, we introduce a mechanism for re-ranking the predicates based on contexts.

Context-Dependent Predicate Generation. Let X and Y be two entities. We first compute a probability vector $\boldsymbol{v}_{(X,Y)} \in \mathbb{R}^M$ denoting the compatibility of (X,Y) with each predicate $R \in \mathcal{R}$ to form a proposition $R(X,Y)$:

$$\boldsymbol{v}_{(X,Y)} = \text{softmax}(\boldsymbol{W}(\boldsymbol{h}_X \oplus \boldsymbol{h}_Y)) \tag{5}$$

where $\boldsymbol{h}_X$ and $\boldsymbol{h}_Y$ are representations of X and Y, $\oplus$ is a concatenation operator, and $\boldsymbol{W} \in \mathbb{R}^{m\times 2d}$ is a model parameter. We combine integrate the compatibility probabilities with the role-predicate association probabilities for final predicate generation. Specifically, for an event trigger T, a certain entity $B \in \mathcal{B}$ and argument candidate $A \in \mathcal{A}$, we generate the following two predicates:

$$r_1 = \arg\max_{R\in\mathcal{R}}(\boldsymbol{a}_{\texttt{RL}}^{(1)} \odot \boldsymbol{v}_{(T,B)} \cdot s_T \cdot s_B) \tag{6}$$

$$r_2 = \arg\max_{R\in\mathcal{R}}(\boldsymbol{a}_{\texttt{RL}}^{(2)} \odot \boldsymbol{v}_{(B,A)} \cdot s_B \cdot s_A) \tag{7}$$

where $\odot$ is an element-wise multiplication operator and s_X indicates the score[3] of an entity X selected to be in the candidate clue entity set $\mathcal{B}$ (Eq. (1)). In this way, the generated FOL rule is $\texttt{RL}(T,A) \leftarrow r_1(T,B) \wedge r_2(B,A)$, suggesting a reasoning path to reach the event argument A: $T \xrightarrow{r_1} B \xrightarrow{r_2} A$.

4 Optimization and Generalization

Optimization with FOL rules is typically challenging due to their discrete nature (Qu et al., 2021a). Here we present T-Norm fuzzy logic for relaxation, which yields an end-to-end learning process.

T-Norm Fuzzy Logic for Relaxation. T-Norm fuzzy logic generalizes classical two-valued logic by admitting intermediary truth values between 1 (truth) and 0 (falsity). For our generated FOL rule $\texttt{RL}(T,A) \leftarrow r_1(T,B) \wedge r_2(B,A)$, we set the truth values of $r_1(T,B)$ and $r_2(B,A)$ to be the corresponding scores in Equation (6) and (7), denoted by p_1 and p_2 respectively. Then, following the Łukasiewicz T-Norm logic, the conjunction of two propositions corresponds to:

$$p(r_1(T,B) \wedge r_2(B,A)) = \min(\mathrm{p}_1, \mathrm{p}_2) \tag{8}$$

where we re-write it as a metric[4]: $M(T,B,A) = p(r_1(T,B) \wedge r_2(B,A))$ and use it for rule ranking and optimization. Particularly, we enumerate each entity-argument pair $(B, A) \in \mathcal{B} \times \mathcal{A}$, and denote the one with the highest score by $(\hat{B}, \hat{A})$. We then derive the following loss for optimization:

$$\mathcal{J}(\Theta) = -\log \frac{\exp(M(T,\hat{B},\hat{A}))}{\sum_{B\in\mathcal{B},A\in\mathcal{A}} \exp(M(T,B,A))} \tag{9}$$

where Θ indicates the overall parameter set (In the training time, the ground-truth argument is known, and we can directly set the optimal argument to the ground-truth). Even though our method considers each candidate entity and argument, we show with parallel tensor operations, our method runs rivalry as effectively as prior methods (see Appendix A.1).

[3] We set the scores of s_T and s_A as 1 as the trigger T and the argument A has no relation with the clue entity set.

[4] Since r_1 and r_2 are completely dependent on T, B and A, we do not consider them as arguments of the metric.

Generalization to General Cases. We explain our method using a structure two-predicate structure, but it is easy to adapt it for general cases with any number of predicates. Now assume a n-predicate structure. We first learn a sequence of role-predict association vectors $\boldsymbol{a}_{\text{RL}}^{(1)}, \boldsymbol{a}_{\text{RL}}^{(2)}, \cdots, \boldsymbol{a}_{\text{RL}}^{(n)}$ using an auto-regressive regime similar to E.q. (3) and (4). Then, we re-rank and generate predicates $r_1, r_2, \cdots, r_n$ to form the logic rule. For optimization, we drive the following metric, $p(r_1 \wedge r_2 \wedge \cdots \wedge r_n) = \min(\mathrm{p}_1, \mathrm{p}_2, \cdots, \mathrm{p}_n)$, which is similar to E.q. (8) to perform rule ranking and model training.

5 Experimental Setups

Benchmarks and Evaluations. We conduct experiments using two document-level EAE benchmarks: RAMS (Ebner et al., 2020) and WikiEvents (Li et al., 2021). The RAMS benchmark defines 139 event types and 59 semantic roles and gives 7,329 annotated documents; The WikiEvents benchmark defines 50 event types and 59 semantic roles and provides 246 annotated documents. The detailed data statistics are shown in Table 1. Following (Ebner et al., 2020; Liu et al., 2021), we adopt the type constrained decoding (TCD) setup for evaluation, which assumes the events triggers and their types are known. We employ Span-F1 on RAMS and Head-F1 and Coref-F1 on WikiEvents as evaluation metrics, where Head-F1 only examines the head word in an argument and Coref-F1 also takes co-reference linkages between arguments into account (Du and Cardie, 2020a; Li et al., 2021; Wei et al., 2021; Ma et al., 2022).

Implementations. In our approach, we use $\text{BERT}_{\text{base}}$ to learn the contextualized word representations (Devlin et al., 2019). The hyper-parameters are tuned using the development set. Finally, the size of the entity candidate set K is set to 40, selected from the range [20, 30, 40, 50], whereas the size of the argument candidate set is determined automatically by the external entity recognizer. The number of predicates M is set to 20 out of [10, 15, 20, 25] options. For optimization, we use the Adam optimizer (Kingma and Ba, 2015) with a batch size of 10 from [5, 10, 15, 20] and a learning rate of 1e-4 from [1e-3, 1e-4, 1e-5].

Baselines. For comparison, we consider the following four categories of methods: 1) Traditional approaches, such as BIOLabel (Shi and Lin, 2019), which views the task as a sequential labeling problem. 2) Global encoding methods, such as QAEE (Du and Cardie, 2020b) and DocMRC (Liu et al., 2021), which form the task as a document-based question-answering problem, and MemNet (Du et al., 2022), which uses a memory to store global event information. 3) Generative methods, such as BART-Gen (Li et al., 2021), which proposes a sequence-to-sequence paradigm for argument extraction, and PAIE (Ma et al., 2022), which employs a set generation formulation. 4) Methods using extra supervisions, for example, FEAE (Wei et al., 2021), which adopts frame-related knowledge, and TSAR (Xu et al., 2022), which utilizes abstract meaning representation (AMR) resources.

Dataset	Split	# Trigger	# Arg.	# Entity
RAMS	Train	7,329	17,026	123,127
	Dev.	924	2,188	13,305
	Test	871	2,023	30,345
WikiEv.	Train	3,241	4,552	64,171
	Dev.	345	428	5,968
	Test	365	566	7,044

Table 1: Data statistics of RAMS and WikiEvents.

6 Experimental Results

In this section, we present the key results, separated by the overall performance and results of capturing long-range dependencies.

6.1 Overall Performance

Table 2 and 3 display the performance of different models on RAMS and WikiEvents, respectively. By adopting the chain-of-reasoning paradigm, our approach outperforms previous methods by significant margins and achieves state-of-the-art performance — 56.1% in F1 on RAMS and 72.3% in Head-F1 and Coref-F1 on WikiEvents. Notably, our model uses no external resources for training, yet it outperforms previous models trained with extensive external resources by over 6% in F1 on RAMS and 4% in Head-F1 (7% in Coref-F1) on WikiEvents. In addition, we discover that the main improvement derives from improved recall, suggesting that learning the reasoning logic rule facilitates locating arguments that were difficult for previous global reasoning methods.

Model	P	R	F1
BIOlabel (Shi and Lin, 2019)	39.9	40.7	40.3
QAEE (Du and Cardie, 2020b)	42.4	44.9	43.6
DocMRC (Liu et al., 2021)	43.4	48.3	45.7
MemNet (Du et al., 2022)	46.2	47.0	46.6
BART-Gen (Li et al., 2021)	42.1	47.3	44.5
PAIE (Ma et al., 2022)	-	-	49.5
FEAE (Wei et al., 2021)	53.1	42.7	47.4
TSAR (Xu et al., 2022)	-	-	48.1
Our Method	**54.8**	**57.5**	**56.1***

Table 2: Results on RAMS. $*$ indicates a significance test at the level of p = 0.05.

Model	P_{Head}	R_{Head}	$F1_{Head}$	$F1_C$
BIOLabel (2019)	55.2	52.3	53.7	56.7
QAEE (2020b)	54.3	53.2	56.9	59.3
DocMRC (2021)	56.9	51.6	54.1	56.3
MemNet (2022)	57.2	51.8	54.4	58.8
BART-Gen (2021)	54.0	51.2	52.6	65.1
PAIE (2022)	-	-	66.5	-
TSAR (2022)	-	-	68.1	66.3
Our Method	**73.5**	**71.2**	**72.3***	**72.3***
w/o GT Entity	68.8	70.4	69.6	65.6

Table 3: Results on WikiEvents. "w/o GT Entity" denotes the use of predicted entities rather than ground-truth (GT) entities as argument candidates. $*$ indicates a significance test at the level of p = 0.05.

6.2 Addressing Long-Range Dependencies

We then assess the ability of different models to handle long-range dependencies, which is crucial for the document-level task. Table 4 and 5 show results on different argument-trigger distance d — accordingly, our model achieves remarkable performance for addressing long-range dependencies, for example, yielding 10.9%, 15.7%, and 6.7% absolute improvement in F1 for d=-1, d=1, and d=2 on RAMS, respectively. The insight behind the effectiveness is that by adopting the chain-of-reasoning paradigm, our method can utilize clue entities to reduce the distance between triggers and arguments, which therefore facilitates learning with long context. Nevertheless, we also note that our method yields relatively poor performance when the argument is two sentences prior to the trigger (d=-2). One possible reason is that our reasoning chain always starts with the trigger and we do not define

	Argument-Trig. Distance				
Model	$-2_{4\%}$	$-1_{8\%}$	$0_{83\%}$	$1_{4\%}$	$2_{2\%}$
BIOLabel (2019)	14.0	14.0	41.2	15.7	4.2
DocMRC (2021)	21.0	20.3	46.6	17.2	12.2
BART-Gen (2021)	17.7	16.8	44.8	16.6	9.0
PAIE (2022)	21.7	27.3	54.7	29.4	25.4
FEAE (2021)	23.7	19.3	49.2	25.0	5.4
TSAR (2022)	**24.3**	21.9	49.6	24.6	11.9
Our Method	15.0	**38.2**	**59.8**	**45.1**	**32.1**

Table 4: Performance (F1-score) of different models for capturing long-range dependencies on RAMS.

	Argument-Trig. Distance		
Model	$<=-1_{6\%}$	$0_{88\%}$	$>=1_{2\%}$
BIOLabel (2019)	34.4	54.6	31.5
DocMRC (2021)	31.5	56.2	40.0
BART-Gen (2021)	64.5	67.5	39.4
PAIE (2022)	68.8	69.5	41.3
Our Method	**70.5**	**75.0**	**44.1**

Table 5: Performance (F1-score) of different models for capturing long-range dependencies on WikiEvents.

reverse predicates[5], which may limit its flexibility. We leave addressing these issues for further work.

7 Discussion

We conduct a series of detailed studies to further verify the effectiveness of our model. To ease discussion, we use the RAMS benchmark as a case.

7.1 Ablation Study

We perform an ablation study to analyze the influence of different components.

Impact of Predicate Generation. Table 6 contrasts our method with methods employing various predicate generation strategies: 1) "w/o Predicate Generation", which generates the reasoning path directly without predicate generation (in other words, it only cares if there is a relationship between two variables or not, but not the specific relationship). 2) "w/o Role Association", which removes the role-predicate association learning process in which a predicate is determined purely by the two variables. 3) "w/o CTX Re-Rank", which

[5]For example, we may define r^{-1} as the reverse predicate of r if $r(T, B) \iff r^{-1}(B, T)$

Model	P	R	F1	Δ_{F1}
Full Approach	54.8	57.5	56.1	-
w/o Predicate Gen.	42.7	25.7	32.2	23.9↓
w/o Role Asso.	39.7	41.1	40.4	15.7↓
w/o CTX Re-Rank	54.2	50.4	52.2	3.9↓

Table 6: Ablations on predicate generation mechanisms.

Rule's Length	P	R	F1
One (Strict)	12.0	36.3	18.1
Two (Strict)	37.0	38.5	37.8
Three (Strict)	38.0	35.4	36.7
Two (Adaptive)	**54.8**	57.5	**56.1**
Three (Adaptive)	52.9	**58.6**	55.6
Two (Ensemble)	53.4	55.7	54.5
Three (Ensemble)	52.6	57.0	54.7

Table 7: The impact of a LOC rule's length. N (Strict) indicates that we adopt a rule with N predicates precisely, N (Adaptive) indicates that we adopt a rule with N predicates at most, N (Ensemble) indicates that we ensemble the results by marginalization.

omits the context-dependent predicate re-ranking process in which the predicates are completely generated by the role. According to the results, predicate generation is essential for reasoning; without it, performance drops significantly (23.9% in F1). In addition, the semantic of the role is essential for predicate generation; without it, performance falls by 15.7% in F1. Lastly, learning context-dependent predicate re-ranking is advantageous, resulting in a 3.9% absolute improvement in F1.

Ablation on the Rule's Length. Table 7 examines the effect of predicate count in a LOC rule, where N (Strict) denotes that we adopt a rule with N predicates precisely, N (Adaptive) denotes that we adopt a rule with N predicates at most and consider the prediction with the greatest score adaptively, N (Ensemble) indicates that we ensemble the results by summing the final score of an argument. The results demonstrate that mandating a fixed number of predicates leads to poor performance, whereas providing the option to choose varying numbers of predicates results in excellent performance. This also implies that the argument-finding process does involve different reasoning patterns. In addition, we do not notice an advantage of N (Adaptive) over N (Ensemble), indicating that FOL rules may not facilitate ensemble.

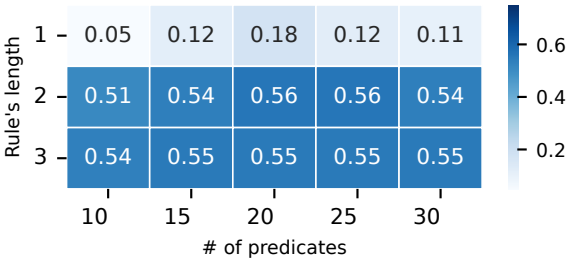


Figure 3: The impact of the amount of predicates.

Ablation on the Amount of Predicates. Figure 3 examines the effect of the number of predicates on the final performance based on the RAMS development set, as well as their joint effect with the length of the rule (we use the Adaptive setting). According to the results, our method is insensitive to the number of predicates and consistently achieves high performance when the number of predicates is more than 15. In addition, we demonstrate that the number of predicates can be lowered when the rule length is increased (e.g., from two to three). This makes sense, as a longer rule implies a longer reasoning chain, which already has a high degree of intrinsic expressivity. In contrast, for a 1-length FOL rule, the performance is always unsatisfactory even if we increase the number of predicates to increase their diversity.

7.2 Robustness Evaluation

Given that our approach uses FOL rules to capture the essential reasoning pattern, it might be more robust than previous methods to perform reasoning. We validate this assumption by analyzing its performance in low-resource scenarios and for defending against adversarial attacks (Jia and Liang, 2017).

Performance in Low-Resource Scenarios. Figure 4 compares different models in low-resource conditions, which show models training on only partial training data (we report 5-run average to against randomness). Clearly, our approach consistently outperforms other methods, and remarkably, in extremely low resource settings (less than 5% training data), it outperforms PAIE based on prompting with large pre-trained language models and TSAR based on external resources, indicating its effectiveness and generalizability in learning FOL rules for reasoning. The performance improves as more training data becomes available.

Example	Semantic Role	LOC Rule
1) Feaver added, noting that all three countries are waging brutal assaults on Sunni groups in **[Syria]** that are likely to fuel ...	`Place`	$\texttt{Place}(T, ?) \leftarrow r_7(T, \underline{S})$
2) ... intends to retake **[Aleppo]** — the rebel stronghold. The UN's envoy to **<Syria Staffan>** warned that the battle could be ..	`Place`	$\texttt{Place}(T, ?) \leftarrow r_2(T, \underline{S}) \wedge r_4(\underline{S}, \underline{A})$
3) ... suicide bombings at **[Bataclan concert hall]** ... and **<ISIS>** claimed responsibility for that massacre, which left ...	`Place`	$\texttt{Place}(T, ?) \leftarrow r_2(T, \underline{I}) \wedge r_4(\underline{I}, \underline{B})$
4) ... report said that the party of former **<Ukraine>** president ... set aside the payments for **[Manafort]** as part of an illegal ...	`Recipient`	$\texttt{Rec.}(T, ?) \leftarrow r_2(T, \underline{U}) \wedge r_6(\underline{U}, \underline{M})$
5) ... government surveillance via **<weblink>** ... *50 words are omitted* ... Whistleblower Edward Snowden said: "**[People]** ...	`ObservedEntity`	$\texttt{Obs.}(T, ?) \leftarrow r_3(T, \underline{w}) \wedge r_5(\underline{w}, \underline{P})$

Table 8: Case study. The trigger is underlined, the argument is in bracketed **[]**, and the clue entity is in **<>**. In the generated FOL rule, we use T to denote the trigger and use the capital letter to indicate an entity/argument.

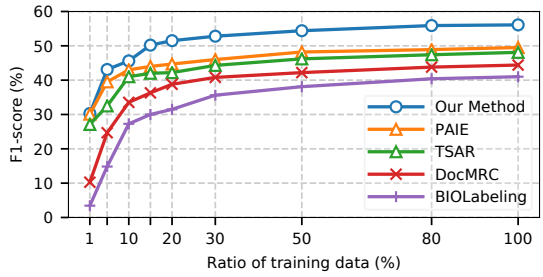
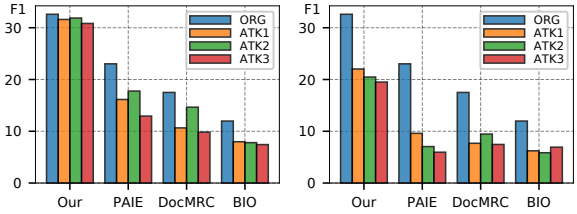


Figure 4: Results in low-resource scenarios, where the performance is based on a 5-run average.



(a) Attack (Random) (b) Attack (Argument)

Figure 5: Results in defending against adversarial attacks. ORG indicates the original performance. ATK1, ATK2, and ATK3 are three types of noises.

Defending Against Adversarial Attacks. Figure 5 shows results in defending adversarial attacks by injecting three forms of noises in a testing example. ATK1: we randomly replace a word in the sentence that contains the trigger with the slot symbol [BLANK]; ATK2: we put the corrupted sentence "The answer is [BLANK]" after the sentence that contains the trigger. ATK3: we insert a sentence "The argument of the [ROLE] is [BLANK]" after the sentence that contains the trigger, where [ROLE] is replaced by the semantic role on which we focus. Two settings are considered: Attack (Random), where the slot is filled with an argument that fulfills the same role in other instances. Attack (Gold), where the slot is filled with the ground-truth argument, but we consider it an error if the model predicts the argument in the slot to be the answer since the injected sentence is unrelated to the context. The results show that our approach is excelled at defending against adversarial attacks, especially with the Attack (Random) setting (see Figure 5(a)). One reason is that our method forces predicting arguments that have semantic relations with other entities in the document context, so it is less affected by the isolated injected arguments. Defending the attacks with ground-truth arguments is more challenging (Figure 5(b)), but our method still achieves the best overall performance.

7.3 Interpretability and Case Study

Table 8 examines the interpretability of our method using a case study. By analyzing cases 1), 2), and 3), we suggest that our method can generate specific and context-dependent reasoning rules for the same semantic role. In addition, the reasoning patterns for cases 2) and 3) are similar, where r_2 may be interpreted as an `Attacker` predicate and r_4 as a `LocatedIn` predicate. Case 4) generates the same predicate r_2 as cases 2) and 3), which may be interpreted as a `Committer` predicate for the payment event; it shares a similar semantic as `Attacker` to an `Attack` event in cases 2) and 3). Case 5) indicates that our method can capture extremely distant dependencies.

8 Conclusion

In conclusion, we present a new chain reasoning paradigm for document-level EAE, demonstrating clear benefits in capturing long-range dependen-

cies and improving interpretability. Our method constructs a first-order logic rule to represent an argument query, with T-Norm fuzzy logic for end-to-end learning. With this mechanism, our approach achieves state-of-the-art performance on two benchmarks and demonstrates decent robustness for addressing low-resource scenarios and defending against adversarial attacks. In future work, we seek to extend our methodology to other tasks requiring modeling of long-range dependencies, such as document-level relation extraction.

9 Limitations

One limitation of our method is that when there are rules of different lengths, the final result is decided by ensemble, not by building a model to generate a single rule with the best length. The second way is more natural and important because figuring out the length of the rule is also a key part of symbolic reasoning. However, it requires more parameterization (for example, the length of the rule could be a parameter) and a more advanced way to optimize. The investigation of the above method is left for future works.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.62106016), the Open Projects Program of the State Key Laboratory of Multimodal Artificial Intelligence Systems, and the Tencent Open Fund.

References

Kareem Ahmed, Tao Li, Thy Ton, Quan Guo, Kai-Wei Chang, Parisa Kordjamshidi, Vivek Srikumar, Guy Van den Broeck, and Sameer Singh. 2022. Pylon: A pytorch framework for learning with constraints. In *Proceedings of the NeurIPS 2021 Competitions and Demonstrations Track*, volume 176 of *Proceedings of Machine Learning Research*, pages 319–324. PMLR.

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.

Nancy Chinchor. 1991. MUC-3 evaluation metrics. In *Third Message Uunderstanding Conference (MUC-3): Proceedings of a Conference Held in San Diego, California, May 21-23, 1991*.

M. J. Cresswell and G. E. Hughes. 1996. *A New Introduction to Modal Logic*. Routledge.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020a. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8010–8020, Online. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020b. 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, Sha Li, and Heng Ji. 2022. Dynamic global memory for document-level argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5264–5275, Dublin, Ireland. Association for Computational Linguistics.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics.

Ralph Grishman. 2019. Twenty-five years of information extraction. *Natural Language Engineering*, 25(6):677–692.

P. Hajek. 1998. *The Metamathematics of Fuzzy Logic*. Kluwer.

Ruihong Huang and Ellen Riloff. 2012. Bootstrapped training of event extraction classifiers. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 286–295, Avignon, France. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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.

Tao Li and Vivek Srikumar. 2019. Augmenting neural networks with first-order logic. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 292–302, Florence, Italy. 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.

Jian Liu, Yufeng Chen, and Jinan Xu. 2022a. Document-level event argument linking as machine reading comprehension. *Neurocomputing*, 488:414–423.

Jian Liu, Yufeng Chen, and Jinan Xu. 2022b. Mrcaug: Data augmentation via machine reading comprehension for document-level event argument extraction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:3160–3172.

Jian Liu, Chen Liang, and Jinan Xu. 2022c. Document-level event argument extraction with self-augmentation and a cross-domain joint training mechanism. *Knowledge-Based Systems*, 257:109904.

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.

Mattia Medina, Grespan, Ashim Gupta, and Vivek Srikumar. 2021. Evaluating relaxations of logic for neural networks: A comprehensive study. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2812–2818. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, Bonan Min, and Thien Nguyen. 2022. Document-level event argument extraction via optimal transport. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1648–1658, Dublin, Ireland. Association for Computational Linguistics.

Meng Qu, Junkun Chen, Louis-Pascal A. C. Xhonneux, Yoshua Bengio, and Jian Tang. 2021a. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Meng Qu, Junkun Chen, Louis-Pascal A. C. Xhonneux, Yoshua Bengio, and Jian Tang. 2021b. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *ArXiv preprint*, abs/1904.05255.

C. Walker and Linguistic Data Consortium. 2005. *ACE 2005 Multilingual Training Corpus*. LDC corpora. Linguistic Data Consortium.

Wenya Wang and Sinno Pan. 2022. Deep inductive logic reasoning for multi-hop reading comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4999–5009, Dublin, Ireland. Association for Computational Linguistics.

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.

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.

A Appendix

A.1 Parallelization and Training Time

We show methods for parallelizing our approach to identify the optimal entity-argument pair and compare our approach to others in real training time. Given an event trigger T with a query $\texttt{RL}(T, ?)$, a

Model	Time (minutes)
BIOlabel (Shi and Lin, 2019)	7.5
QAEE (Du and Cardie, 2020b)	28.0
DocMRC (Liu et al., 2021)	55.0
FEAE (Wei et al., 2021)	56.3
BART-Gen (Li et al., 2021)	14.0
PAIE (Ma et al., 2022)	11.1
Our Method	33.5

Table 9: The training time per epoch of different models on the RAMS dataset.

candidate entity set $\mathcal{B} = \{b_i\}_{i=1}^{K}$ of size K, and an argument candidate set $\mathcal{A} = \{b_i\}_{i=1}^{L}$ of size L, we first compute the predicate comparability for the event trigger T with each candidate entity in $\mathcal{B}$ as follows:

$$\boldsymbol{V}_{(T,\mathcal{B})} = \text{softmax}(\boldsymbol{W}(\boldsymbol{h}_T \oplus \boldsymbol{H}_{\mathcal{B}})) \quad (10)$$

where the concatenation operator of the vector $\boldsymbol{h}_T \in \mathbb{R}^d$ and the matrix $\boldsymbol{H}_{\mathcal{B}} \in \mathbb{R}^{M\times d}$ is performed by first broadcasting the vector to the same dimension as the matrix, followed by an element-wise concatenation operation. This results in a M by K matrix: $\boldsymbol{V}_{(T,\mathcal{B})} \in \mathbb{R}^{M\times K}$. To unify illustration, we add an extra dimension to $\boldsymbol{V}_{(T,\mathcal{B})}$ to represent the event trigger, which thus makes it a high-order tensor $\boldsymbol{V}_{(T,\mathcal{B})} \in \mathbb{R}^{M\times 1\times K}$. In a similar fashion, we compute the predicate comparability for each entity-argument pair in $\mathcal{B}$ and $\mathcal{A}$ and obtain a high-order tensor $\boldsymbol{V}_{(\mathcal{B},\mathcal{A})} \in \mathbb{R}^{M\times K\times L}$:

$$\boldsymbol{V}_{(\mathcal{B},\mathcal{A})} = \text{softmax}(\boldsymbol{W}(\boldsymbol{H}_{\mathcal{B}} \oplus \boldsymbol{H}_{\mathcal{A}})) \quad (11)$$

where the concatenation operator of two matrices is implemented by first broadcasting each matrix into a dimension of K by L by d and then concatenating each element individually.

Given $\boldsymbol{V}_{(T,\mathcal{B})}$ and $\boldsymbol{V}_{(\mathcal{B},\mathcal{A})}$, we can apply a softmax operator[6] on their first dimension to identify the best-fitting predicates for each trigger-entity and entity-argument pair and only keep the maximum values as their scores. Suppose the results are two matrices $\boldsymbol{O}_1 \in \mathbb{R}^{1\times K}$ and $\boldsymbol{O}_2 \in \mathbb{R}^{K\times L}$ for $\boldsymbol{V}_{(T,\mathcal{B})}$ and $\boldsymbol{V}_{(\mathcal{B},\mathcal{A})}$ respectively. We then apply the T-Norm relaxation for the conjunction operator as follows:

$$\boldsymbol{O} = \underline{\min}(\hat{\boldsymbol{O}}_1, \hat{\boldsymbol{O}}_2) \quad (12)$$

[6]We omit the computation produces in E.q. (6) and (7) because they have no effect on parallelization.

Noise Type	Document with Noise
ATK1 (Rand.)	[S1][S2][S3]: ... their homes and **Paris** have been damaged, burned or destroyed ... [S4][S5]
ATK2 (Rand.)	[S1][S2][S3] **The answer is Paris.** [S4][S5]
ATK3 (Rand.)	[S1][S2][S3] **The argument of the place is Paris.** [S4][S5]
ATK1 (Arg.)	[S1][S2][S3]: ... their homes and **Aleppo** have been damaged ... [S4][S5]
ATK2 (Arg.)	[S1][S2][S3] **The answer is Aleppo.** [S4][S5]
ATK3 (Arg.)	[S1][S2][S3] **The argument of the place is Aleppo.** [S4][S5]

Table 10: Cases of adversarial examples.

where $\hat{\boldsymbol{O}}_1 \in \mathbb{R}^{1\times K\times L}$ and $\hat{\boldsymbol{O}}_2 \in \mathbb{R}^{1\times K\times L}$, which have the same dimension, are the tensor broadcasting results of $\boldsymbol{O}_1$ and $\boldsymbol{O}_2$ respectively, and $\underline{\min}$ indicates an element-wise minimization operator. Finally, by examining the element with the highest value in $\boldsymbol{O}$, the optimal entity and argument pair can be determined. For example, if $\boldsymbol{O}_{1,4,2}$ is the element with the highest values, then (B_4, A_2) corresponds to the optimal entity-argument pair.

In Table 9, we compare the real training time for each model on the RAMS dataset. All experiments are conducted on a 16G-memory NVIDIA Tesla P100-SXM2 Card to ensure a fair comparison. From the results, we can see that our model maintains a comparable time to earlier methods such as QAEE and is faster than many models such as FEAE and DocMRC, where FEAE has two base models for knowledge distillation and DocMRC uses external data to pretrain the model.

A.2 Cases of Adversarial Examples

In this section, we provide a specific adversarial example to enhance comprehension. Our original document with annotations for the event trigger (which is underlined) and an argument (which is in **bold**) fulfilling a semantic role of `Place` is:

"[S1] People we meet who are displaced took shelter in schools, in unfinished buildings and other facilities, some of which are simply skeleton infrastructure. [S2] Most people with whom I spoke

have been displaced for at least two to three years. [S3] Many of them see no prospect of returning home any time soon, either because fighting is still going on, or because for many of them, their homes and land have been damaged, burned or destroyed. [S4] Every single family is affected, and most communities in **Aleppo**, and beyond, have reached the limit of their endurance. [S5] Aid workers have said there is just enough fuel to keep generators, bakeries, and hospitals running for a month."

We show the generated noisy examples in Table 10, and note that if the model identifies the noisy arguments presented in the table as the result, it should be counted as an error.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 9

☑ A2. Did you discuss any potential risks of your work?
Section 9

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section 6 and 7

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
No response.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 6 and 7

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 5

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.