Thinking about how to extract: Energizing LLMs' emergence capabilities for document-level event argument extraction

Kai Shuang, Ji Zhou *, Qiwei Wang and Jinyu Guo
State Key Laboratory of Networking and Switch Technology,
Beijing University of Posts and Telecommunications
{shuangk, zhouji0121, wqw12468, guojinyu}@bupt.edu.cn

Abstract

There are two key challenges remaining for the document-level event argument extraction (D-EAE) tasks: key feature forgetting and cross-event argument confusion. The emergence capability of large language models (LLMs) holds promise for solving the above two challenges. In this paper, we propose a document-level event argument extraction method based on guided summarization and reasoning (EAESR), which leverages the emergence capabilities of LLMs to highlight key event information and to clarify the explicit and implicit association between multiple events. Specifically, we generate document summarization information that shorten the length of the event context while preserving the key event features. In addition, we generate inter-event reasoning information, which helps EAESR make sense of the correlations between events and reduces their dependence on the event context, especially to better cope with the few-shot D-EAE task. Then, we obtain named entity information to enable EAESR to learn argument boundary features to improve the sensitivity of its argument boundary recognition. Eventually, we fused the above features and sentence features to make EAESR have summarizing and reasoning capabilities simultaneously. Extensive experiments on WIKIEVENTS and RAMS have shown that EAESR achieves a new state-of-the-art that outperforms the baseline models by 1.3% F1 and 1.6% F1, respectively, and averages 11% F1 in few-shot settings.

1 Introduction

Event argument extraction (EAE), which is a critical task in Event extraction (EE), can be divided into sentence-level EAE (S-EAE) and document-level EAE (D-EAE). As shown in Figure 1, S1 describes an *Injury* event, S2 describes an *Identification* event, etc., yet there are no events for S4 and S5 in the document. Most existing D-EAE models (Zhang et al., 2020; Wei et al., 2021) encode the entire document, which results in event-independent information interfering with event feature modeling, making it difficult to accurately identify events. In order to increase the proportion of key event information in the encoded features, some D-EAE models (Ma et al., 2022; Zhang et al., 2023) set a fixed window to encode a portion of the document. Its lack of global features of the document leads to its missed extraction of arguments scattered outside the window. The above two problems can be summarized as **key feature forgetting** in the D-EAE task.

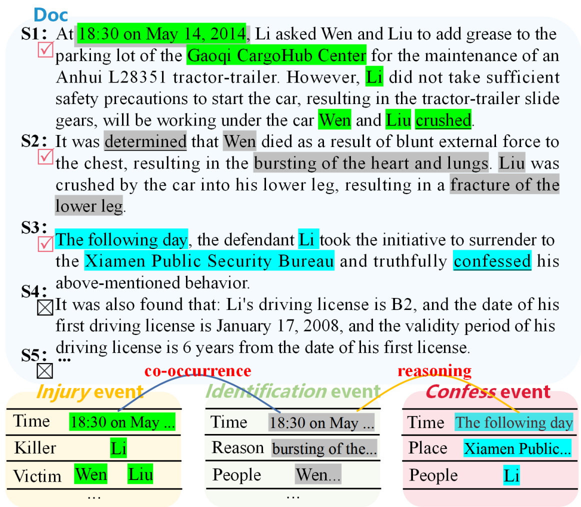

Figure 1: Example of D-EAE. S1-S3 have events, and S4-S5 have no events. There are explicit and implicit associations between the arguments of the different events.

On the other hand, as in Figure 1, "*18:30 on May 14, 2014*" is the *time* of both the *Injury* event and *Identification* event. Yet the *time* of the *Confess* event requires reasoning to learn that it is the day after the *Identification* event. Existing D-EAE models (Xu et al., 2021; Wen et al., 2021; Huang et al., 2020) construct the document as a heterogeneous graph so that they can obtain associations

*Correponding author.

Findings of the Association for Computational Linguistics: ACL 2024, pages 5520–5532
August 11-16, 2024

between tokens in the whole document. They will easily identify the explicit association of the co-occurrence of the argument "*18:30 on May 14, 2014*", while it is difficult to reason about the implicit association of "*The following day*", which may cause the D-EAE model to incorrectly extract the *time* of the *Confess* event as "*18:30 on May 14, 2014*". This phenomenon can be referred to as **cross-event argument confusion** in the D-EAE task.

Recently, large language models (LLMs), like InstructGPT (Ouyang et al., 2022), ChatGPT[1], and ChatGLM (Du et al., 2021), utilize the instructions to work well in various downstream tasks such as conversations, summarization generation, etc. Some works (Gao et al., 2023; Wei et al., 2023) have also tried to design reasonable prompts and utilize ChatGPT for event extraction with encouraging results. However, using LLMs directly for argument generation may cause serious precision problems due to the illusions of LLMs (Xu et al., 2024) that can result in generating arguments outside of context. Although it is not appropriate to apply LLMs directly for extracting arguments, we believe that the emergence capabilities of LLMs hold promise for D-EAE models to model complex implicit associations in events, especially if they are used for event association analysis rather than argument generation.

Inspired by this idea, we propose a document-level **E**vent **A**rgument **E**xtraction method based on guided **S**ummarization and **R**easoning (EAESR), which utilizes LLMs to generate document summarization information and guidance descriptions for argument extraction as external supplementary features to address the key feature forgetting and cross-event argument confusion challenges of the D-EAE task. Specifically, for the key feature forgetting challenge, we first design the prompt for document summarization information generation to streamline the content of the document. As the structure between triggers and arguments in the event has a strong similarity with the structure between nodes and edges in the graph. We establish an abstract meaning representation (AMR) graph for the document summarization information connecting the tokens in the summarization information. Next, we use graph convolutional networks (GCN) to learn the weights of the edges in the AMR graph, not only to obtain the global features of the document but also to establish co-occurrence associations of the event elements.

For the cross-event argument confusion challenge, we design the prompt for reasoning information generation to analyze the explicit and implicit associations of inter-event arguments. We use LLMs to do a step-by-step analysis of the document's event associations and obtain the event reasoning features. Through changing the learning objective from event features to event reasoning features, EAESR can accurately extract event arguments without relying on a large amount of training data. Since most arguments in an event are named entities, we design prompts to extract entities from sentences to build entity features in order to improve the sensitivity of EAESR for recognizing argument boundaries. Finally, we use the attention fusion layer combine the sentence features with the aforementioned features and obtain the event record. In summary, our proposed innovations and contributions are as follows:

(1) We propose a novel D-EAE model, called EAESR. It utilizes the emergent summarization and reasoning capabilities of the LLMs to efficiently address key feature forgetting and cross-event argument confusion for the D-EAE task.

(2) We change the learning objective of the D-EAE task from event features to event reasoning features that reduce EAESR's dependence on large labeled data and improve its performance on the few-shot D-EAE task.

(3) Extensive experiment results on WIKIEVENTS and RAMS show that EAESR achieves a new state-of-the-art that outperforms the baseline models by 1.3% F1 and 1.6% F1, respectively, and averages 11% F1 in few-shot settings.

2 The Proposed EAESR Method

As shown in Figure 2, EAESR contains three core components: *Summarizing feature extraction* extracts the event key information in the document through LLMs. This allows EAESR to obtain the global semantics of the document while shortening the encoding length. *Reasoning feature extraction* extracts the event argument extraction guidance description and entities through LLMs. This help EAESR to understand the explicit and implicit association of inter-event arguments and boundary features of the arguments. *Event argument extrac-*

[1]https://openai.com/blog/chatgpt

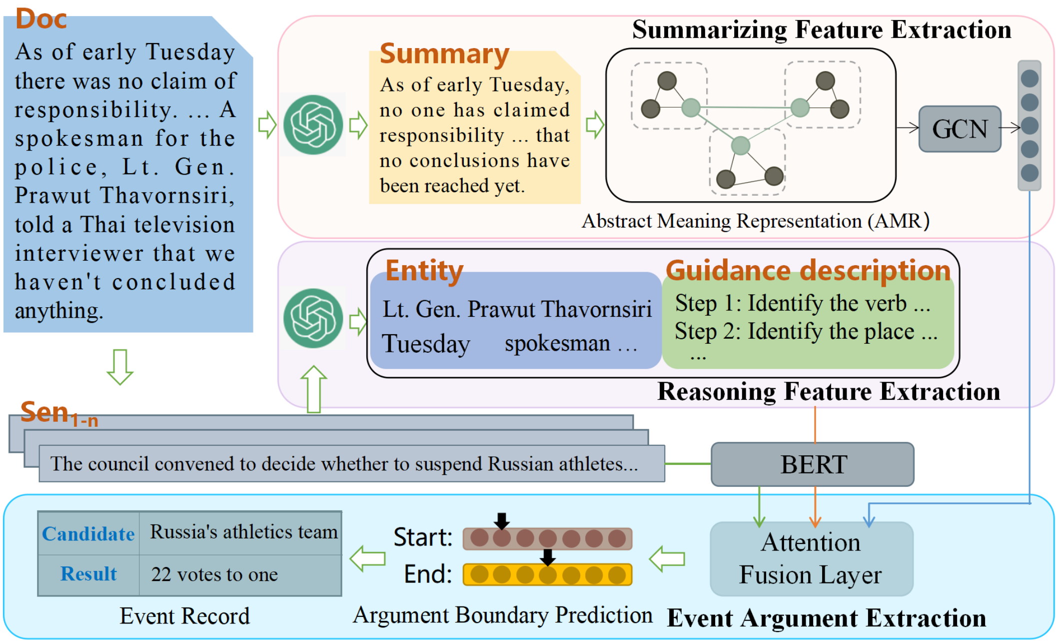

Figure 2: Overall architecture of EAESR. Given an input document, EAESR first generates document summarization information, event reasoning information and entity information using ChatGPT. Then EAESR constructs an AMR graph based on the document summarization information and uses GCN to obtain global features. Next, EAESR inputs sentence information, event reasoning information and sentence entity information into BERT to obtain sentence features, reasoning features and entity features. Finally EAESR uses the attention fusion layer to get the fusion features and the event record can be extracted from the fusion features.

tion fuses the sentence features with the above features through the attention fusion layer, so that EAESR can address both of key feature forgetting and cross-event argument confusion challenges.

2.1 Task formulation

We formulate the D-EAE task as a multiple-span boundary prediction task for the roles on dataset D. Given an instance $(C, t, e, A^{(e)}) \in D$, where $C = \{s_i\}$ denote the document, and s_i is the sentences in the document. t and e are the trigger and event type. $A^{(e)} = \{(r_i, span_i), \ldots\}$ denotes the set of event-specific role types, where r_i denotes the role, and $span_i$ is the offset of the argument. The inputs to the event argument extraction module are sentence features f_s, global features f_g, reasoning features f_r and entity features f_e. The target output is an event record $A^{(e)}_{Pred} = \{(r_i, span_i), \ldots\}$, which includes the roles and the model's predicted arguments for those roles.

2.2 Summarizing feature extraction

Extracting summarization information from a document can filter out context that is not relevant to the key event information, such as adjectives, conjunctions, and extra descriptive phrases. We first utilize the *Summarizing information generation module* to generate the document summarization information. Second, considering that AMR's ability to build richer semantic associations from the summarization information, we utilize the *AMR graph construction module* to build an AMR graph for this summarization information. Finally, we use the *global feature generation module* to convert the AMR graph into a heterogeneous graph based on edge types. GCN is used to learn the weights of edges to update the degree of association between nodes in the AMR graph, and ultimately to obtain global features.

Summarizing information generation module: given the original document, we design the prompt (SP) for summarizing information extraction, which consists of three parts: event type, maximum length of the summary L, and document. For example, "*summarize the text related to <e> from the following document, with a word limit of <L>: "<C>"*". Based on the text summarization capability of LLMs, we input SP into ChatGPT to get the document summarization information. Appendix B shows the details.

AMR graph construction module: we use the

AMR parser[2] to obtain the initial AMR graph of the summarizing information. The input of the AMR parser is the summarized content, and the output is a directed graph, where each node u denotes a semantic concept, e.g., *city*, *name*, etc., and each edge e describes the categorical semantic relationship between two concepts, e.g., *location*, *time*, etc. In the initial AMR graph, nodes with the same edge type have a higher probability of being relevant arguments. We follow previous works (Zhang and Ji, 2021; Xu et al., 2022) to further cluster the relevant edge types in the initial AMR graph into 8 major categories, including *spatial*, *temporal*, *means*, *modifiers*, *operators*, *prepositions*, *core roles*, and *others*, to get the final AMR graph G_s for extracting significant information from the document.

Global feature generation module: we define the AMR graph as a heterogeneous graph based on edge types, and the node embeddings in the heterogeneous graph are initialized using BERT. Then, we use L-layer GCN to learn the interaction weights between the nodes, calculated as shown in Eq (1):

$$h_u^{(l+1)} = \sigma(\sum_{k \in K} \sum_{v \in N_k^{(u)} \cup \{u\}} \frac{1}{c_{u,k}} W_k^{(l)} h_u^{(l)}), \quad (1)$$

where σ is the ReLU function, K is the number of edges of the heterogeneous graph, in this paper K = 8, $W_k^{(l)}$ is the trainable parameters, $N_k^{(u)}$ denotes the neighbor of node u connected in the k-th edge, $c_{u,k}$ is a normalization constant, $h_u^{(l)}$ is the embeddings of node u. We then use a linear layer l_n to further transform $H_u^{(L)}$ into the global feature f_g, as shown in Eq (2) below:

$$f_g = l_n[H_u^{(L)}], \quad (2)$$

where $H_u^{(L)} \in \mathbb{R}^{N \times H}$ denotes the set of all nodes in the graph. N is the number of node, H is the hidden state. L is the number of GCN layers, and it is set to 3 in this study. $f_g \in \mathbb{R}^{S \times H}$, and S is the sequence length.

2.3 Reasoning feature extraction

Designing prompts based on downstream task requirements will purposefully improve the performance of downstream tasks, such as code prompts (Wang et al., 2022) for code generation tasks. One effective approach is that supplementing the model with more intermediate processes, such as chain-of-thought (Wei et al., 2022), can improve the model's performance on downstream tasks compared to a prompt with only inputs and outputs. Based on this consideration, we supplement the D-EAE task with more reasoning features and entity features. For reasoning feature construction, we design prompts (RP) such as,""*<C>", in this document there are some <e> events, how do you find out the <r_i> in the <e_i> event? ... Please think step by step*". Based on the reasoning capability of the LLMs, we input the RP into the ChatGPT to get the reasoning information F_r, which consists the event argument extraction steps, the analysis of associations between arguments, and preliminary argument extraction conclusions. Appendix B shows the details. Through the representation of explicit and implicit associations between arguments as a statement, it is able to guide the model to extract the arguments of the given event type step-by-step.

For entity feature construction, we design prompts (EP) such as, "*You are an expert in the field of entity extraction, and now you are required to extract all the entities from the following sentence, "<s_i>""*". LLMs have a large amount of general knowledge that can effectively recognize common entities in text, such as *time*, *people*, *location*, etc. We input EP into ChatGPT to get entity information F_e. Then, we use BERT to encode F_r and F_e to get reasoning features $f_r \in \mathbb{R}^{S \times H}$ and entity features $f_e \in \mathbb{R}^{S \times H}$, respectively, as in Eq (3):

$$f_{r,e} = \text{BERT}(F_{r,e}). \quad (3)$$

2.4 Event argument extraction

We use BERT to encode s_i to get sentence features f_s. Then, we use the *attention fusion layer* to fuse the sentence features with the above features to get fusion features H. Next, we define a simple but effective *argument boundary prediction* method, which given the roles and fusion features, EAESR recognizes the offset of the argument span and obtains the event record.

Attention fusion layer: sentence features contain direct features of events; global features, reasoning features, and entity features are used as external features to supplement sentence features for event argument extraction. We first fuse sentence features and global features to get F_{cln}^g, which enables EAESR to learn the global semantics of documents, and then we fuse sentence features and reasoning features to get F_{cln}^r, which enables EAESR

[2] https://github.com/IBM/transition-amr-parser

to learn associations between event arguments as shown in Eqs (4)-(6):

$$gamma = l_n(f_s) + g, \quad (4)$$

$$beta = l_n(f_s) + b, \quad (5)$$

$$F_{cln}^{[r,g]} = \frac{gamma \times (f_{[r,g]} - \mathrm{mean}(f_{[r,g]}))}{\mathrm{std}(f_{[r,g]})} + beta, \quad (6)$$

where l_n is a linear layer, g and b are the trainable parameters, $\mathrm{mean}(\cdot)$ is the mean matrix of $f_{[r,g]}$ and $\mathrm{std}(\cdot)$ is the standard deviation matrix of $f_{[r,g]}$.

In order to make EAESR learn both global and reasoning features and to improve the its sensitivity in recognizing the argument boundaries. We fuse global, reasoning, and entity features using a multi-head attention mechanism as shown in Eqs (7)-(10) and Figure 3. Entity features have a greater impact on the boundary recognition of the argument span, so make it as V in the attention mechanism directly multiplied with the fusion result of Q and K. f_s, F_{cln}^r, F_{cln}^g, and f_e are fused to get the in-depth feature F_{sa}^{rg}. Inspired by the idea of residuals (He et al., 2016), we concatenate shallow features F_{cln}^g with in-depth features F_{sa}^{rg} and feed them into the $LayerNorm(\cdot)$ module and the GeLU function to obtain the final output H.

$$SA(Q,K,V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V, \quad (7)$$

$$F_{sa}^s = SA(F_{cln}^g, f_s, f_e), \quad (8)$$

$$F_{sa}^{rg} = SA(F_{sa}^s, F_{cln}^r, f_e), \quad (9)$$

$$H = \mathrm{GeLU}[LayerNorm(F_{sa}^{rg} + F_{cln}^g)]. \quad (10)$$

Argument boundary prediction: given the feature representation H of the event and the set of roles r_i, we use Eq (11) to compute the probability that each token in the sentence is selected as the start/end position of the argument span for each role. We then define the loss function as Eqs (12)-(13), and finally get the complete event record $A_{Pred}^{(e)}$.

$$p_k^{s,e} = \mathrm{Sigmoid}(H), \quad (11)$$

$$L^{s/e} = -\sum_{k=0}^{K}(1-\hat{p}_k^{s/e})\log(1-p_k^{s/e}) + \hat{p}_k^{s/e}\log(p_k^{s/e}), \quad (12)$$

$$L = L^s + L^e, \quad (13)$$

where p_k^s and p_k^e are the probabilities of the token in the sentence as the start position and end position of the argument, and $\hat{p}_k^s$ and $\hat{p}_k^e$ are the labels of the start position and end position of the argument, respectively.

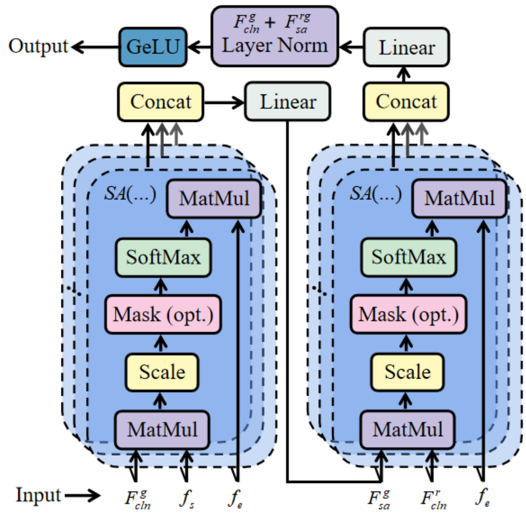

Figure 3: The architecture of the feature fusion layer.

3 Experiment

Datasets We evaluate EAESR on two popular D-EAE datasets: RAMS (Ebner et al., 2019) and WIKIEVENTS (Li et al., 2021). RAMS has 139 event types and 65 argument roles, and the average document length in the test set is 134 words. WIKIEVENTS has 50 event types and 59 argument roles, and the average document length in the test set is 789 words. Appendix A shows detailed statistics.

Baselines We compare our model with the following state-of-the-art baseline models: (1) D-EAE methods based on the span boundary prediction model: **FEAE** (Wei et al., 2021), **BERT-CRF**, **TSAR** (Xu et al., 2022). (2) D-EAE methods based on QA/MRC models: **EEQA** (Du and Cardie, 2020), **EEQA-BART**, **DocMRC** (Liu et al., 2021). (3) D-EAE methods based on generative models: **BART-Gen** (Li et al., 2021), **PAIE** (Ma et al., 2022), **UnifiedEAE** (Zhou et al., 2022), **Memory-DocIE** (Du et al., 2022), **APE** (Zhang et al., 2023), **RA-DocEAE** (Ren et al., 2023). Appendix C describes the above baseline model in detail.

Evaluation metric Following Ma et al. (2022), we adopt two evaluation metrics. (1) Argument Identification F1 score (Arg-I): an argument span is correctly identified when the predicted offset fits the ground truth span. (2) Argument Classification F1 score (Arg-C): both the span and the argument role type are matched with the ground truth.

Method	RAMS		WIKIEVENTS	
	Arg-I	Arg-C	Arg-I	Arg-C
FEAE (Wei et al., 2021)	53.5	47.4	-	-
BERT-CRF	-	39.3	<u>72.2</u>	56.7
TSAR (Xu et al., 2022)	-	48.1	**73.2**	<u>66.3</u>
EEQA (Du and Cardie, 2020)	46.4	44.0	54.3	53.2
EEQA-BART	49.45	46.3	60.3	57.1
DocMRC (Liu et al., 2021)	-	45.7	-	43.3
UnifiedEAE (Zhou et al., 2022)	55.5	49.9	69.8	64.0
PAIE (Ma et al., 2022)	53.0	49.8	68.2	63.4
BART-Gen (Li et al., 2021)	50.9	44.9	47.5	41.7
Memory-DocIE (Du et al., 2022)	55.0	47.3	63.5	58.0
RA-DocEAE (Ren et al., 2023)	53.3	46.3	61.4	46.1
APE (Zhang et al., 2023)	<u>56.1</u>	<u>51.6</u>	70.7	66.0
EAESR	**60.2**	**53.2**	71.3	**67.6**

Table 1: Overall performance. We highlight the best result and underline the second best of the D-EAE methods. The Pre-trained Language Models (PLMs) all use the base models.

Method	RAMS		WIKIEVENTS	
	Arg-I	Arg-C	Arg-I	Arg-C
GPT3.5	46.2	40.4	42.4	40.6
GLM2-6B	50.9	45.8	60.3	58.6
EAESR	**60.2**	**53.2**	**71.3**	**67.6**

Table 2: Performance of LLMs on RAMS and WIKIEVENTS.

Method	RAMS		WIKIEVENTS	
	Arg-I	Arg-C	Arg-I	Arg-C
+events	**60.2**	**53.2**	**71.3**	**67.6**
-events	59.3	52.6	69.6	65.6

Table 3: Impact of prompt on RAMS and WIKIEVENTS.

4 Main results

4.1 Overall performance

Tabel 1 compares EAESR with the baseline models. We observe that EAESR performs best on RAMS and WIKIVENTS, which obtained +1.6% and +1.3% gains in F1 (Arg-C), respectively, that can prove EAESR is effective in both long-document D-EAE tasks (WIKIEVENTS) and short-document D-EAE tasks (RAMS). We also realized that TSAR achieved the second-best results on WIKIEVENTS yet not on RAMS, suggesting that AMR is more capable of modeling long content features. APE achieved the second-best results in RAMS and competitive results in WIKIEVENTS, indicating that overlapping knowledge between datasets plays an important role in improving the generalization performance of the D-EAE task.

4.2 Detailed analysis

4.2.1 Performance of LLMs on D-EAE tasks

In this section, we compare the performance of directly using LLMs (ChatGPT3.5 and ChatGLM2-6B) and EAESR for event argument extraction in RAMS and WIKIEVENTS. For LLMs, we design the event argument extraction prompt as "*You are an expert in the field of event extraction. Now you are required to extract the argument: <r_i> from following sentence: <s_i>. Only output in the following format as <r_1> is 'a word from the sentence', ... without outputting other words or analysis*". And as the parameters of ChatGLM2-6B are much less than those of ChatGPT3.5, we converted the EAE task to a dialog task and make instruction-tuning for ChatGLM2-6B. Table 2 shows the comparison results of LLMs and EAESR, in which it can be shown that EAESR is 7.4% F1 (Arg-C) and 9% F1 (Arg-C) higher than ChatGLM2-6B on RAMS and WIKIEVENTS, respectively, which suggests that there are serious precision issues caused by the direct use of LLMs to generate arguments that outside of context. ChatGLM2-6B outperforms ChatGPT3.5 by 5.4% F1 (Arg-C) and 18% F1 (Arg-C) on RAMS and WIKIEVENTS, respectively, suggesting that injecting task-specific knowledge into LLMs is more beneficial for the D-EAE task than the general knowledge owned by the LLMs themselves.

4.2.2 Impact of prompt on D-EAE tasks

In this section, we compare the impact of providing and not providing event-related information for prompt in the process of generating external supplementary information by using ChatGPT. For SP, the prompt (SP- with-events) that provides event-related information is the prompt used in this paper. The prompt that does not provide event-related information (SP-without-events) is as follows: "*summarize the following document, with a word limit of* <L>*: "*<C>*"*.". For RP, the prompt (RP-with-events) that provides event-related information is the prompt used in this paper. The prompt that does not provide event-related information (RP-without-events) is as follows: "*Analyze the main content of the document. Please think step by step.*". Table 3 shows the results for different prompt settings, and we find that Prompt-with-events contributes more to EAESR than Prompt-without-events, suggesting that it is essential to provide prompts with appropriate event information for both generating document summarization information and event reasoning information.

4.2.3 Performance in few-shot settings

EAESR learns how to extract event arguments by learning event reasoning features. Its application of transfer learning to obtain event-overlapping knowledge further expands the knowledge accumulation. Thus, EAESR can be trained with only a few samples to achieve competitive results of some baseline models trained using all samples. In this section, we utilize RAMS to validate the performance of EAESR in a few-shot scenario. As shown in Figure 4, we find that when there are only 10, 50, 100, and 200 random event records in the training set, EAESR has an improvement of 17.8%F1, 11.8%F1, 8%F1, and 6.4%F1 compared to APE, respectively. Furthermore, EAESR requires only 10 random event records to exceed the training effect that APE requires 200 random event records to achieve, indicating that it can greatly reduce the labor cost associated with labeling training data.

4.3 Ablation study

In this section, we investigate the effectiveness of EAESR by removing each external supplementary feature in turn. (1) **Global features**: we replace F^g_{cln} with f_s. (2) **Reasoning features**: we replace F^s_{cln} with f_s. (3) **Entity features**: we replace f_e in the SA$(\cdot)$ with f_s.

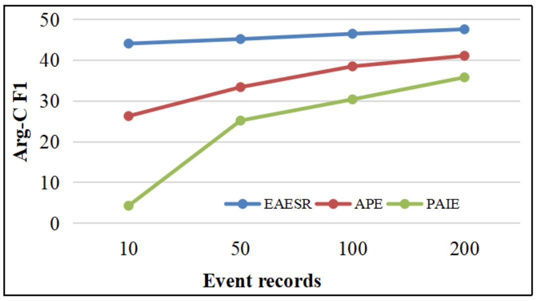

Figure 4: Comparison of models in RAMS with few-shot settings.

Method	RAMS		WIKIEVENTS	
	Arg-I	Arg-C	Arg-I	Arg-C
EAESR	**60.2**	**53.2**	**71.3**	**67.6**
-f_g	59.7	52.9	70.4	66.1
-f_r	58.7	51.9	71.0	66.3
-f_e	59.5	52.3	70.5	67.1

Table 4: Ablation study on RAMS and WIKIEVENTS.

We summarize the results of ablation studies in Table 4 as follows: 1) After removing the global features, EAESR decreased by 0.3% F1 (Arg-C) and 1.5% F1 (Arg-C) on the RAMS and WIKIEVENTS, respectively. This suggests that providing EAESR with a global feature can be effective in complementing the current sentence semantic features with semantic information from other sentences, and that the effect is more pronounced as the document get longer. 2) After removing the reasoning features, EAESR decreased by 1.3% F1 (Arg-C) and 1.3% F1 (Arg-C) on the RAMS and WIKIEVENTS, respectively. This suggests that the reasoning information of events enhances the ability of EAESR to learn associations between events and that it is not sensitive to document length. 3) After removing the entity features, EAESR decreased by 0.9% F1 (Arg-C) and 0.5% F1 (Arg-C) on the RAMS and WIKIEVENTS, respectively. This suggests that entity features can serve to constrain EAESR in recognizing the boundaries of the arguments.

4.4 Case study

To visually demonstrate the benefits of using external supplementary information in our approach, we show two examples comparing the output of ChatGLM2-6B, APE and EAESR as shown in Figure 5 in Appendix. **Example A** describes the *Conflict.Attack.DetonateExplode* event, and we find

Doc Conflict.Attack.DetonateExplode Event

... **Two gunmen**(Attacker) entered the **campus** (Place) right after the explosion and **shot** at many **Afghan soldiers** (Target) before being gunned down during the clashes ...

Output

Method	Attacker	Place	Target
ChatGLM2-6B	Two gunmen	(school) campus	Afghan soldiers
APE	(Two) gunmen	campus	(Afghan) soldiers
EAESR	Two gunmen	campus	Afghan soldiers

Example A

Doc Life.Die.Unspecified Event

... A **young woman** (Victim) was found **dead** in **her home** (Place) and police found a bloodstained knife. ...
... After further investigations, the police confirmed that the deceased was a **woman** (Victim) who had committed **suicide** (Reason)....

Output

Method	Victim	Place	Reason
ChatGLM2-6B	young woman	her home	-
APE	young woman	(her) home	-
EAESR	young woman	her home	suicide

Example B

Figure 5: Case study for EAESR.

that ChatGLM2-6B tends to generate semantically rich arguments, which leads to over-extraction problems and generating arguments outside the original context. For instance, the word "*school*" is not included in the original context, but ChatGLM2-6B fails to adhere to the instructions and generates words outside the original context. We statistically analyze the extraction results of ChatGLM2-6B and EAESR in WIKIEVENTS, where the precision (Arg-C) of EAESR is 72.90%, while the precision (Arg-C) of ChatGLM2-6B is 62.87%. In ChatGLM2-6B's extraction results, the loss of precision due to parsing errors was 1.70%, and the loss of precision due to hallucinations was 15.23%, which shows that the hallucinations of LLM have a huge impact on the EAE task. APE tends to extract the core words of an argument, leading to its missed extraction, e.g., *Target*'s argument is missing "*Afghan*". EAESR mitigates the problem of over- or under-extraction due to its use of entity features to constrain the boundaries of recognized arguments, and avoids extract arguments outside of the original context.

Example B describes the *Life.Die.Unspecified* event, in which the *Reason*'s argument is not in the same sentence as the other arguments. And it is necessary to use "*woman*" to relate the events in the two sentences in order to infer that "*suicide*" in the second sentence is the *Reason* of the *Life.Die.Unspecified* event. Because EAESR uses event reasoning features and document global features to provide implicit associations between events and global semantics of documents, it extracts "*suicide*", while the other two models do not.

5 Related works

Document-level Event Argument Extraction The goal of the D-EAE task is to extract event arguments from the given triggers and roles. The methods of the D-EAE task can be divided into D-EAE based on span boundary prediction, QA-based models, and generative models. The D-EAE method based on span boundary prediction (Zhang et al., 2020; Dai et al., 2022; Yang et al., 2023; He et al., 2023) is to consider the D-EAE task as a classification task. In order to make the D-EAE model more focused on the associations between events in a document, some work (Liu et al., 2020; Chen et al., 2020; Liu et al., 2021) uses Question Answering (QA)/Machine Reading Comprehension (MRC) to understand the document semantics before extracting the event arguments. The D-EAE method based on generative models (Lu et al., 2021; Li et al., 2021; Hsu et al., 2022; Ren et al., 2023; Lin et al., 2023) to designing diverse prompts makes the D-EAE task more relevant to the text generation task. This method allows the event's arguments to be directly generated in the sequence-to-structure manner, however, some restriction of the generation (Lu et al., 2021) need to be taken to avoid the model generating words that are not within the event's definition.

LLMs in Event Extraction LLMs have excellent emergence capabilities, and they can achieve impressive performance on a wide range of downstream tasks, such as ChatLaw (Cui et al., 2023), MuseChat (Dong et al., 2023), ChatReviewing (Berrezueta-Guzman et al., 2023), and so on. Some work has also attempted to apply LLMs to event extraction tasks. Wei et al. (2023) attempt to convert a zero-shot information extraction task into a multi-round QA task using ChatGPT. It achieves promising performance on the zero-shot information extraction task and even outperforms the full-shot model on some datasets (e.g., NYT11-HRL (Takanobu et al., 2019)). Gao et al. (2023) tests the performance of ChatGPT on ACE2005 (Doddington et al., 2004) and show that due to ChatGPT's lack of event specific knowledge, it is only 51.04% as effective as task-specific models (e.g., EEQA (Du and Cardie, 2020)) in long-tailed and complex scenarios. In summary, LLMs are far better at text comprehension than they are at event extraction, and how to exploit the potential of LLMs for event extraction tasks still needs to be thoroughly investigated.

6 Conclusion and future works

In this work, we propose a novel model of EAESR that can resolve key feature forgetting and cross-event argument confusion simultaneously. We utilize LLMs to generate external supplementary features related to events, including document global features, event reasoning features, and entity features. The document global features can provide EAESR with a global perspective and help it solve key feature forgetting. The event reasoning features and entity features can provide EAESR with explicit and implicit associations between events and help it solve cross-event argument confusion. Extensive experiments on RAMS and WIKIEVENTS demonstrate the effectiveness of our proposed model in the D-EAE task. In future work, we will explore the use of LLMs to generate supplemental features for other information extraction tasks, such as event reasoning features for event relation extraction tasks.

Limitations

Our goal is to utilize the emergence capabilities of LLMs to improve the performance of the D-EAE task, due to their large number of parameters, leading to the fact that inference on LLMs will be time-consuming. As an example, generating document summarization information, event reasoning information, and entity information for WIKIEVENTS will take 30h on the NVIDIA A40 48GB GPU. This limitation is expected to be alleviated by the adoption of a lightweight generative model. In addition, EAESR is based on AMR graphs generated by a pre-trained AMR parser. The AMR graph of the generated document summarization inevitably has a certain possibility of imperfection, which leads to error propagation. In future work, applying LLMs to construct a joint extraction model for document global information will likely avoid error propagation.

Ethics Statement

Our work complies with the ACL Ethics Policy. The document level event argument extraction (D-EAE) task is a well-defined classical task in the field of event extraction (EE). In this work, our use of existing datasets is licensed and consistent with their intended use. We see no other ethical issues.

Acknowledgements

This work was supported by Beijing Natural Science Foundation (Grant No.4222032) and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No.61921003).

References

Jonnathan Berrezueta-Guzman, Laura Malache-Silva, and Stephan Krusche. 2023. Chatgpt-4 as a tool for reviewing academic books in spanish. In *Latin American Conference on Learning Technologies*, pages 384–397. Springer.

Yunmo Chen, Tongfei Chen, Seth Ebner, Aaron Steven White, and Benjamin Van Durme. 2020. Reading the manual: Event extraction as definition comprehension. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pages 74–83, Online. Association for Computational Linguistics.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv:2306.16092*.

Lu Dai, Bang Wang, Wei Xiang, and Yijun Mo. 2022. Bi-directional iterative prompt-tuning for event argument extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6251–6263, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Zhikang Dong, Bin Chen, Xiulong Liu, Pawel Polak, and Peng Zhang. 2023. Musechat: A conversational music recommendation system for videos. *arXiv preprint arXiv:2310.06282*.

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, 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.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv:2103.10360*.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2019. Multi-sentence argument linking. *arXiv preprint arXiv:1911.03766*.

Jun Gao, Huan Zhao, Changlong Yu, and Ruifeng Xu. 2023. Exploring the feasibility of chatgpt for event extraction. *arXiv preprint arXiv:2303.03836*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Yuxin He, Jingyue Hu, and Buzhou Tang. 2023. Revisiting event argument extraction: Can EAE models learn better when being aware of event cooccurrences? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12542–12556, Toronto, Canada. Association for Computational Linguistics.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908, Seattle, United States. Association for Computational Linguistics.

Kung-Hsiang Huang, Mu Yang, and Nanyun Peng. 2020. Biomedical event extraction with hierarchical knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1277–1285, Online. Association for Computational Linguistics.

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.

Zizheng Lin, Hongming Zhang, and Yangqiu Song. 2023. Global constraints with prompting for zero-shot event argument classification. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2527–2538, Dubrovnik, Croatia. 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.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event 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 2795–2806, 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.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Yubing Ren, Yanan Cao, Ping Guo, Fang Fang, Wei Ma, and Zheng Lin. 2023. Retrieve-and-sample: Document-level event argument extraction via hybrid retrieval augmentation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–306, Toronto, Canada. Association for Computational Linguistics.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7072–7079.

Chaozheng Wang, Yuanhang Yang, Cuiyun Gao, Yun Peng, Hongyu Zhang, and Michael R Lyu. 2022. No more fine-tuning? an experimental evaluation of prompt tuning in code intelligence. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 382–394.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

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.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Haoyang Wen, Yanru Qu, Heng Ji, Qiang Ning, Jiawei Han, Avi Sil, Hanghang Tong, and Dan Roth. 2021. Event time extraction and propagation via graph attention networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 62–73, Online. Association for Computational Linguistics.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. 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 3533–3546, 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. *arXiv preprint arXiv:2205.00241*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

Xianjun Yang, Yujie Lu, and Linda Petzold. 2023. Few-shot document-level event argument extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8029–8046, Toronto, Canada. Association for Computational Linguistics.

Kaihang Zhang, Kai Shuang, Xinyue Yang, Xuyang Yao, and Jinyu Guo. 2023. What is overlap knowledge in event argument extraction? APE: A cross-datasets transfer learning model for EAE. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 393–409, Toronto, Canada. Association for Computational Linguistics.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020. 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.

Zixuan Zhang and Heng Ji. 2021. Abstract meaning representation guided graph encoding and decoding for joint information extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–49.

Jie Zhou, Qi Zhang, Qin Chen, Qi Zhang, Liang He, and Xuanjing Huang. 2022. A multi-format transfer learning model for event argument extraction via variational information bottleneck. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1990–2000, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Appendix

A Datasets

We present detailed dataset statistics in Table 5.

B Summarization and reasoning information

We present detailed summarization and reasoning information in Table 6.

C Baselines

This section supplements the baseline models used in this paper.

D-EAE methods based on the span boundary prediction model: **FEAE** (Wei et al., 2021) trained a teacher model for implicit EAE by introducing a course knowledge extraction strategy. **BERT-CRF** defines the D-EAE task as a sequence labeling task. **TSAR** (Xu et al., 2022) is the first and sole work utilizing AMR for D-EAE.

D-EAE methods based on QA/MRC models: **EEQA** (Du and Cardie, 2020) defines the EAE task as an end-to-end question-answering (QA) task. **EEQA-BART** replaces BERT with BART for event extraction. **DocMRC** (Liu et al., 2021) regards the EAE task as a document reading comprehension (MRC) task.

D-EAE methods based on generative models: **BART-Gen** (Li et al., 2021) proposes a conditional generation approach to complete the D-EAE task. **PAIE** (Ma et al., 2022) utilizes multi-role prompts under extractive settings to capture argument interactions. **UnifiedEAE** (Zhou et al., 2022) explores shared knowledge in different event extraction datasets using transfer learning. **Memory-DocIE** (Du et al., 2022) constructs a document memory store to record contextual event information. **APE** (Zhang et al., 2023) defines overlapping knowledge between EAE datasets and combines specific knowledge for event argument extraction. **RA-DocEAE** (Ren et al., 2023) validates the effectiveness of the data retrieval augmentation approach on the D-EAE task.

D Implementation details

We train the D-EAE task of RAMS by loading the pre-trained BERT parameters of WIKIEVENTS and train the D-EAE task of WIKIEVENTS by loading the pre-trained BERT parameters of RAMS. We use the base version of the pre-trained model for all models, like BERT-base, BART-base, and LLMs, which use ChatGPT3.5 and ChatGLM2-6B. We train models on NVIDIA-A40 by AdamW with a 0.1 warmup ratio and 0.01 weight decay. We set the initial learning rate to 1e-5, the batch size for training to 2, and the number of training epochs to 30.

Dataset	Train		Dev		Test	
	#Sents	#Args	#Sents	#Args	#Sents	#Args
RAMS	7329	17026	924	2188	871	2023
WIKIEVENTS	5262	4552	378	428	492	566

Table 5: Statistics of datasets. #Sents denotes the number of sentences of the dataset, #Args denotes the number of arguments of the dataset.

Information Type	Generated Information
Summarization	Indonesian police have received an anonymous letter warning that Bali will be the next target for a terrorist assault after the bombings in Jakarta last week by Islamist militants. The authorities are increasing security at shopping malls and other locations that draw crowds in Bali following the bomb threats. Jemaah Islamiyah, an Indonesia-based terrorist group with links to al-Qaida, targeted Bali in 2002 killing 202 people, mostly foreigners. The Bali bombing severely hurt Indonesia's tourism industry. Indonesia successfully combated the JI related terror threat through police action, intelligence operations and high profile criminal prosecutions. However, after last week's attack there are concerns of more deadly attacks carried out by groups inspired by ISIS.
Reasoning	To identify the Recipient and Communicator in the "ThreatenCoerce Correspondence" event mentioned in the sentence, we can follow these steps: 1. Identify the Threat: In this case, it is a letter that contains a threat or coercion. 2. Identify the Target of the Threat Coercion: The sentence mentions that the letter was sent to Buleleng district, which implies that Buleleng district is the target of the threat coercion. 3. Identify who Sent Communicated the Threat Coercion: The sentence also mentions that an anonymous individual sent the letter, but their identity is unknown. 4. Identify Law Enforcement Response: The Bali Police Chief states that an investigation is underway to find out who sent the letter and urges people not to be afraid but stay alert. Therefore, in this "ThreatenCoerce Correspondence" event, Buleleng district is identified as the recipient or target of threat coercion while an anonymous individual is identified as a communicator sender of this threat coercion through their written correspondence (letter).

Table 6: Example of summarization and reasoning information.