# TISE: A Tripartite In-context Selection Method for Event Argument Extraction

**Yanhe Fu**[1,2], **Yanan Cao**[1,2,*], **Qingyue Wang**[1,2] and **Yi Liu**[1,2]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{fuyanhe,caoyanan,wangqingyue,liuyi1999}@iie.ac.cn

## Abstract

In-context learning enhances the reasoning capabilities of LLMs by providing several examples. A direct yet effective approach to obtain in-context example is to select the top-k examples based on their semantic similarity to the test input. However, when applied to event argument extraction (EAE), this approach exhibits two shortcomings: 1) It may select almost identical examples, thus failing to provide additional event information, and 2) It overlooks event attributes, leading to the selected examples being unrelated to the test event type. In this paper, we introduce three necessary requirements when selecting an in-context example for EAE task: semantic similarity, example diversity and event correlation. And we further propose TISE, which scores examples from these three perspectives and integrates them using Determinantal Point Processes to directly select a set of examples as context. Experimental results on the ACE05 dataset demonstrate the effectiveness of TISE and the necessity of three requirements. Furthermore, we surprisingly observe that TISE can achieve superior performance with fewer examples and can even exceed some supervised methods.

## 1 Introduction

Event Argument Extraction (EAE) aims to identify and classify event arguments within textual data based on the given event type. Label for this task is a structured table in which each argument requires annotation. Resulting in a scarcity of training data, which limits the generalizability of traditional methods to unknown samples (Gao et al., 2023a,b).

Recently, Large Language Models (LLMs) exemplified by GPT-3 (Brown et al., 2020) have shown remarkable capabilities in low-resource scenarios. As shown in Figure 1, given the test input and three labeled examples, LLM can finish inference, which is called in-context learning (ICL; Brown et al., 2020). However, it has been found that the performance of LLMs is highly dependent on the selection of these examples (Rubin et al., 2022), highlighting the importance of choosing appropriate in-context example. To address this issue, some researchers calculate the semantic similarity between examples and test input, and subsequently selecting the top-$k$ examples as context (Rubin et al., 2022; Liu et al., 2022), yielding appreciable performances.

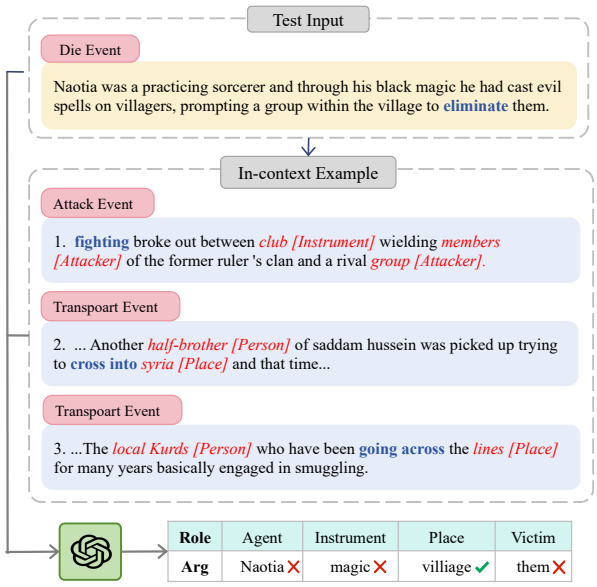


Figure 1: An in-context learning example on EAE task, where examples are obtained by the semantic selection. The upper left of each sentence is its event type, argument [role] is colored *red* and trigger is **bolded**.

While this semantic selection method has shown empirical success, it exhibits two shortcomings when applied to EAE. **Firstly**, it may select examples with overlapping semantics, thus failing to provide additional event information. In Figure 1, the second and third examples con-

*Corresponding Author.



*Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1801–1818

tain the same event ("cross-border") and identical roles ("Person" and "Place"). As a result, the in-context example provides limited event information and primarily contributes to predicting the role "Place". **Secondly**, it does not consider the correlation of event attributes between the test input and examples, which is also crucial for LLMs to make reasonable predictions. As shown in Figure 1, there is no example related to the test event type ("Die"), thus LLM is unable to comprehend the test input and predict the relevant roles, e.g., "Victim" and "Agent".

In this paper, we propose TISE (A **T**ripartite **I**n-context **S**election method for **E**vent argument extraction) to select an optimal in-context example for EAE task. TISE scores examples based on three requirements, related to the different factors mentioned above. Firstly, it calculates the textual similarity between examples and test input as **Semantic Similarity**, which serves as a fundamental guarantee. Secondly, TISE calculates the **Example Diversity** by evaluating the inter-example similarity. This requirement avoids the semantic overlap between examples, thus obtaining the diverse examples with more event information. Lastly, TISE designs natural language descriptions for event types and event roles and calculates the **Event Correlation** by measuring the similarity between these descriptions. This requirement ensures the selected examples can effectively convey event information to LLM. TISE utilizes the kernel matrix of the Determinantal Point Process (DPP) (Kulesza et al., 2012) to combine these scores, and the example set with the highest probability is selected as the in-context example. Finally, we employ the code imitation prompt (Wang et al., 2023) to conduct EAE task on LLM and evaluate the proposed method on the ACE05 dataset. The experimental results show the effectiveness and robustness of our method. We outline our contributions as follows:

- To the best of our knowledge, we are the first work to explore an optimal in-context example for EAE task and present three requirements for the example selection.
- We further propose an effective selection method, which scores the examples from the above perspectives to obtain the optimal set.
- The proposed model achieves better performance with fewer examples and exceeds the supervised method on the public datasets.

## 2 Related Work

### 2.1 Event Argument Extraction

EAE is a core subtask of Event Extraction (EE), earlier works directly perform EE task, which includes the EAE task (Chen et al., 2015; Nguyen et al., 2016; Yang and Mitchell, 2016; Lin et al., 2020). Recently, EAE has been studied as a standalone task and can be categorized into 3 main paradigms: *Span-based paradigm* treats EAE as a span classification problem, Ebner et al., 2020a; Zhang et al., 2020; Xu et al., 2022 identify all candidate spans and classify them into the corresponding roles. *Reading comprehension paradigm* designs questions for each event role and converts EAE to question-answering problem to extract arguments (Du and Cardie, 2020; Liu et al., 2021). *Text generation paradigm* sequentially generates arguments using auto-regressive LMs with the help of prompts (Li et al., 2021; Du et al., 2021; Lu et al., 2021; Lin et al., 2022). With the development of LLMs, the *generation paradigm* has gained prominence. Researchers directly use LLMs to extract arguments, leading to significant breakthroughs (Wei et al., 2023; Wang et al., 2023; Gao et al., 2023b).

This paper also employs LLMs for EAE, but TISE is orthogonal to these works. We focus on how to select the optimal in-context example and our method is adaptable to various prompts.

### 2.2 In-context Example Selection

Constrained by the difficulties of fine-tuning LLMs, in-context learning (ICL) is proposed to emulate few-shot learning by providing several labeled examples in the prompt (Brown et al., 2020). However, LLMs are sensitive to the quality of in-context example (Liu et al., 2022). To obtain a high-quality demonstration, selecting the top-$k$ similar examples becomes the most intuitive, simple, yet effective method (Rubin et al., 2022; Liu et al., 2022). More thoughtfully, researchers have found considering entropy (Lu et al., 2022; Wu et al., 2022) and diversity (Ye et al., 2022; Su et al., 2022; Levy et al., 2022; Ye et al., 2023) among examples is also useful. We conduct the first study to explore this problem in EAE and propose three requirements.

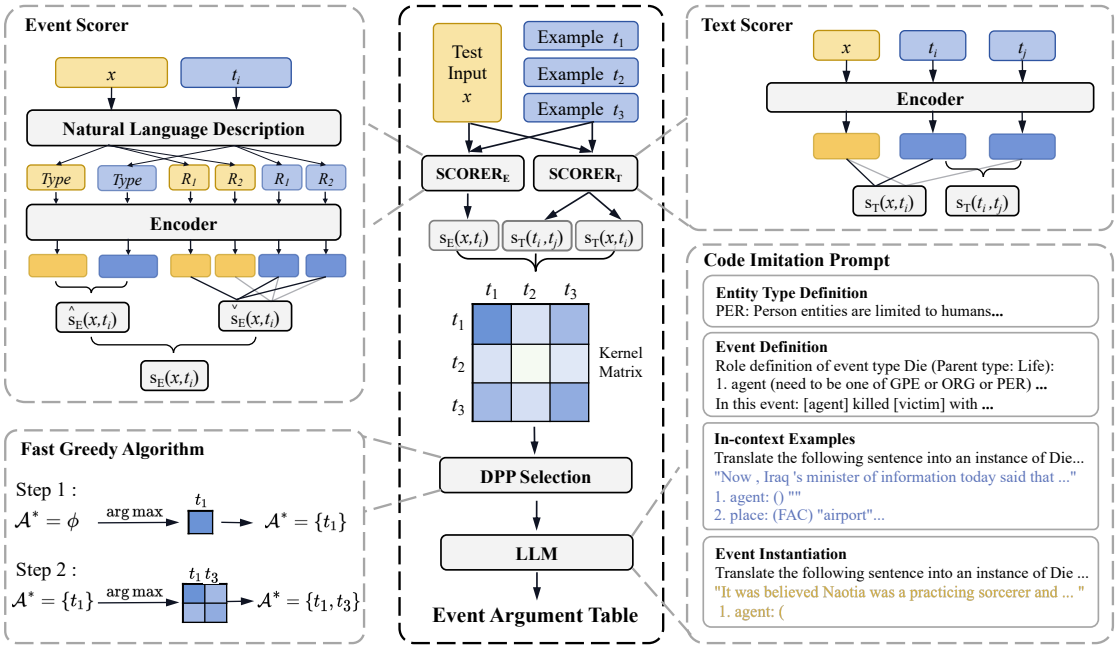


Figure 2: The overall structure of TISE when $|T| = 3$ and $k$=2. The center is the main process, surrounded by the sub-module processes. The main process uses an event scorer and a test scorer to calculate the score for each requirement. These scores are then combined into a kernel matrix. The fast greedy algorithm is employed to obtain the in-context example subset. Finally, the code imitation prompt is referenced to conduct LLM-based EAE.

## 3 Preliminary

### 3.1 Problem Definition

The target of in-context example selection is to select a few examples from the training dataset to form a context for ICL. Formally, given the test input $x$ and the training dataset $T = \{t_1, ..., t_{|T|}\}$, we need to choose $k$ examples from $T$ to form a subset $A \subseteq T$ as in-context example and $|A| = k$.

### 3.2 Determinantal Point Process

Unlike individually selecting the top-$k$ similar examples, DPP considers the co-occurrence of examples and directly obtains a subset with $k$ examples. Specifically, given the condition $x$, DPP is defined as a distribution $P$ to measure the selected probability of each subset, we use a kernel matrix $K \in \mathbb{R}^{|T|\times|T|}$ to represent the co-occurrence between examples, where item $K_{ij} = \mathrm{k}(t_i, t_j|x)$ evaluates the possibility that $t_i$ and $t_j$ appear in the same subset. Thus the selected probability of $A$ is:

$$P(A|x) = \frac{\det(K_A)}{\det(K + \mathbf{I})} \quad (1)$$

where $K_A \equiv [K_{ij}]_{i,j\in A}$ is a sub-matrix obtained by indexing, $\det(\cdot)$ is the determinant and $\mathbf{I}$ is the identity matrix. During the inference, we select the subset with the highest probability to form in-context example $A^* = \arg\max_{A\subseteq T} P(A|x)$.

Obviously, this is an NP-hard submodular maximization problem (Ko et al., 1995), we employ the fast greedy algorithm (Chen et al., 2018) for optimization. As shown in the lower left of Figure 2, we perform a $k$-step search. At each step, the selected example is chosen to maximize the increase in the determinants of the old and new matrices:

$$t^* = \arg\max_{t_i\in T\setminus A^*} \det(K_{A^*\cup\{t_i\}}) - \det(K_{A^*}) \quad (2)$$

and we update the subset using $A^* = A^* \cup \{t^*\}$.

## 4 Method

The overall structure of TISE is shown in Figure 2, it measures Semantic Similarity and Example Diversity using the text scorer ($\mathrm{SCORER_T}$), and assesses Event Correlation using the event scorer ($\mathrm{SCORER_E}$). Three scores are integrated into the kernel matrix $K$, and DPP is used to select an in-context example based on this. Lastly, it employs code imitation prompt to conduct LLM-based EAE.

### 4.1 Scorer and Kernel Matrix

We present sub-module corresponding to each requirement separately, including the scoring

| Event Type | Description |
|---|---|
| Justice:Convict | Involves a justice trial, recording a person has been convicted. |
| Justice:Sentence | Involves a justice trial, recording a person has been sentenced. |
| **Event Role** | |
| Life:Injure.Instrument | What device was used to inflict the harm? |
| Life:Die.Instrument | What device was used to kill? |

Table 1: Examples of natural language descriptions of Event Types and Event Roles.

methodology and its subsequent fusion into the kernel function. Moreover, we introduce hyperparameters to balance each requirement and prove the impact of the kernel function.

#### 4.1.1 Semantic Similarity

Ensure the semantic similarity between $x$ and $t_i$ is fundamental yet effective. As shown in the upper right of Figure 2, we design a text scorer for measurement, it employs a Pretrained Language Model as encoder and calculates the embedding similarity as text score $\mathrm{s_T}$:

$$\mathrm{s_T}(x,t) = \mathrm{sim}(\mathrm{E}(x), \mathrm{E}(t)) \tag{3}$$

where $\mathrm{E}(\cdot)$ denotes the encoder[1] and $\mathrm{sim}\,(u,v) = u^T v / \|u\| \cdot \|v\|$ is the cosine similarity. For $K_{ij}$, we simultaneously consider the semantic score of $t_i$ and $t_i$: $\mathrm{k_1}(t_i, t_j|x) = \mathrm{s_T}(x, t_i) \cdot \mathrm{s_T}(x, t_j)$.

#### 4.1.2 Example Diversity

To guarantee the selected subset contains diverse examples, TISE assigns lower scores to similar examples. It still uses text scorer to measure the similarity between examples. For training example $t_i$ and $t_j$, we calculate the score $\mathrm{s_T}(t_i, t_j)$, and modify $K_{ij}$ as $\mathrm{k_2}(t_i, t_j|x) = \mathrm{s_T}(x, t_i) \cdot \mathrm{s_T}(t_i, t_j) \cdot \mathrm{s_T}(x, t_j)$. The reason why this product form kernel function can simultaneously decrease $\mathrm{s_T}(t_i, t_j)$ yet increase $\mathrm{s_T}(x, t_i)$ will be explained in 4.1.4.

#### 4.1.3 Event Correlation

We propose an event scorer to measure the event correlation. As shown in the upper left of Figure 2, we partition event attributes into event type and event role. Subsequently, we design natural language descriptions for each of them and calculate event scores based on these descriptions.

**Event Type** The event type description comprises two sentences that are related to its parent event type and itself. See Table 1 as an example[2], this form of description has two advantages: *1.Event types belonging to the same parent will be closer.* and *2.Event types with related meanings will feature similar descriptions.*

After obtaining the descriptions, we calculate description similarity to score event type. Suppose $x$ and $t_i$ has event type description $\hat{d}(x)$ and $\hat{d}(t_i)$, we can obtain the event type score $\hat{\mathrm{s}}_\mathrm{E}(x, t_i)$ as:

$$\hat{\mathrm{s}}_\mathrm{E}(x, t_i) = \mathrm{sim}(\mathrm{E}(\hat{d}(x)), \mathrm{E}(\hat{d}(t_i))) \tag{4}$$

**Event Role** For each event role, the description is used to introduce its meaning within the event. Referring to Lu et al., 2023, we design the description as a single sentence. See Table 1 [2], it also offers two advantages: *1.Roles under the same event type exhibit closer relationships.* and *2.The same roles in different event type are similar but not identical.*

Let $R^{t_i}$ denote the set of *not-none* roles for $t_i$, and $R^x$ denote the set of roles to be predicted. For each $r_i^{t_i} \in R^{t_i}$, we obtain its description $\check{d}(r_i^{t_i})$, and sequentially compare it to $r_i^x \in R^x$ with description $\check{d}(r_i^x)$, the role score is calculated as:

$$\check{\mathrm{s}}_\mathrm{E}(r_i^x, r_i^{t_i}) = \mathrm{sim}(\mathrm{E}(\check{d}(r_i^x)), \mathrm{E}(\check{d}(r_i^{t_i}))) \tag{5}$$

We pick the most relevant $r_i^x$ using max-pool: $\check{\mathrm{s}}_\mathrm{E}(x, r_i^{t_i}) = \max \check{\mathrm{s}}_\mathrm{E}(r_i^x, r_i^{t_i})$, and the final event role score for $t_i$ is:

$$\check{\mathrm{s}}_\mathrm{E}(x, t_i) = \frac{\sum_{r_i^{t_i} \in R^{t_i}} \check{\mathrm{s}}_\mathrm{E}(x, r_i^{t_i})}{|R^{t_i}|} + \alpha \big|R^{t_i}\big| \tag{6}$$

where $\alpha = 0.1$ is a hyperparameter to reward those examples containing more useful roles. We combine event type score and event role score so that the final event score can reflect the event correlation from two perspectives:

$$\mathrm{s_E}(x, t_i) = \frac{1}{2}(\hat{\mathrm{s}}_\mathrm{E}(x, t_i) + \check{\mathrm{s}}_\mathrm{E}(x, t_i)) \tag{7}$$

and the final kernel function is $\mathrm{k}(t_i, t_j|x) = \mathrm{s_E}(x, t_i) \cdot \mathrm{k_2}(t_i, t_j|x) \cdot \mathrm{s_E}(x, t_j)$.

#### 4.1.4 Balance and Proof

To balance each requirement, we introduce the hyperparameter $\lambda$ into each score: $\mathrm{s_E}'(x, t_i) = \exp(\frac{\mathrm{s_E}(x,t_i)}{2\lambda_1})$ and $\mathrm{s_T}'(x, t_i) = \exp(\frac{\mathrm{s_T}(x,t_i)}{2\lambda_2})$, thus the kernel matrix $K$ can be represented as:

$$K = \mathrm{Diag}(S'_E) \cdot \mathrm{Diag}(S'_T) \cdot \bar{K} \cdot \mathrm{Diag}(S'_T) \cdot \mathrm{Diag}(S'_E) \tag{8}$$

[1] We use the embedding of [CLS] token as result.

[2] The complete descriptions of event type and event role can be found in Appendix A.1 and A.2, respectively.

| Methods | REQ | $k$=3 | | $k$=5 | | $k$=10 | | $k$=15 | |
|---|---|---|---|---|---|---|---|---|---|
| | | Arg-I | Arg-C | Arg-I | Arg-C | Arg-I | Arg-C | Arg-I | Arg-C |
| RANDOM | ✗ | 53.86 | 41.74 | 54.20 | 42.94 | 55.41 | 44.92 | 58.00 | 47.34 |
| *Semantic Retriever* | | | | | | | | | |
| BM25 | R.1 | 55.48 | 44.09 | 56.81 | 46.01 | 57.27 | 47.14 | 57.94 | 47.88 |
| BERT-TOPK | R.1 | 56.20 | 45.38 | 56.67 | 46.14 | 56.93 | 46.98 | 58.34 | 48.16 |
| DPR-TOPK | R.1 | 56.05 | 45.10 | 57.03 | 46.88 | 57.69 | 47.62 | 58.40 | 47.97 |
| *DPP-based Method* | | | | | | | | | |
| DPP-DIVERSITY | R.1,2 | 56.22 | 45.53 | 58.19 | 47.73 | 58.47 | 47.91 | 59.46 | 49.38 |
| TISE | R.1,2,3 | **56.66** | **46.25** | **58.95** | **48.72** | **60.57** | **50.78** | **60.99** | **51.43** |
| Δ SEMANTIC | – | +0.46 | +0.87 | +1.92 | +1.84 | +2.88 | +3.16 | +2.59 | +3.27 |
| Δ DIVERSITY | – | +0.44 | +0.72 | +0.76 | +0.99 | +2.10 | +2.87 | +1.53 | +2.05 |

Table 2: Results on ACE05. REQ indicates which requirements are satisfied by this method. The gap between TISE and *Semantic Retriever* is Δ SEMANTIC, and the gap between TISE and DPP-DIVERSITY is Δ DIVERSITY.

where $\bar{K}$ is a symmetric matrix. $\bar{K}_{ij} = \mathrm{s_T}(t_i, t_j)$, $S'_{E_i} = \mathrm{s_E}'(x, t_i)$ and $S'_{T_i} = \mathrm{s_T}'(x, t_i)$. Note the $\det(K + \mathbf{I})$ in Eq.(1) is a normalization term, we only consider the effect of $\det(K_A)$:

$$\mathrm{logdet}(K_A) = \sum_{t_i \in A} (\frac{S_{E_i}}{\lambda_1} + \frac{S_{T_i}}{\lambda_2}) + \mathrm{logdet}(\bar{K}_A) \tag{9}$$

So far, $P(A|x)$ has been associated with two parts via the kernel matrix. The first part requires the selected examples exhibit high event score and semantic score, while the second part necessitates low inter-example similarity due to the determinant's characteristics[3].

### 4.2 Code Imitation Prompt

TISE employs code imitation prompt (Wang et al., 2023) as template to instruct LLM to extract arguments based on the test input and in-context example. As shown in the lower right of Figure 2, it consists of 4 components[4]. 1) *Entity Type Definition* categorizes all argument entities into seven types and begins with a description of all entity types involved in test input. 2) *Event Definition* includes the hierarchy of event types, the role definition of current type and how these roles are presented in this event. 3) *In-context Examples* demonstrates selected examples individually. Each example contains extraction instruction, an input sentence and a label. 4) *Event Instantiation* directly feeds the extracted instruction and test input to LLM.

[3]See Appendix A.3 for detailed proof.

[4]A specific prompt example is shown in Appendix A.4.

## 5 Experiments

### 5.1 Dataset and Evaluation Metrics

We evaluate TISE on the Automatic Content Extraction 2005 (ACE05; Doddington et al., 2004) dataset, which is a sentence-level corpus. For fair comparison, we preprocess the dataset into 8 parent event types and 33 child types following Wadden et al., 2019.

We measure the performance with two metrics following previous work (Li et al., 2021; Ma et al., 2022): Argument Identified (Arg-I) denotes the head word of an argument matches the human annotation, and Argument Classified (Arg-C) means an argument is correctly classified into its annotated role as well. We report them using F1 score.

### 5.2 Implementation Details

For scorers and the kernel matrix, we employ `bert-base-uncased` (Devlin et al., 2018) as encoder and set $\lambda_1 = 0.5, \lambda_2 = 0.05$[5]. For LLM, due to the Codex models have been deprecated since March 2023[6], we mainly use `text-davinci-002` (Ouyang et al., 2022) as LLM. This model is instruction-tuned from `code-davinci-002` (Chen et al., 2021) and supports 4k input tokens. We access LLMs through OpenAI API[7]. For prompt, because the code prompt (Wang et al., 2023) easily exceeds the length limitation, we leverage the code imitation prompt as prompt template.

[5]The impact of $\lambda$ is shown in Appendix A.5.

[6]https://platform.openai.com/docs/guides/code

[7]https://openai.com/product

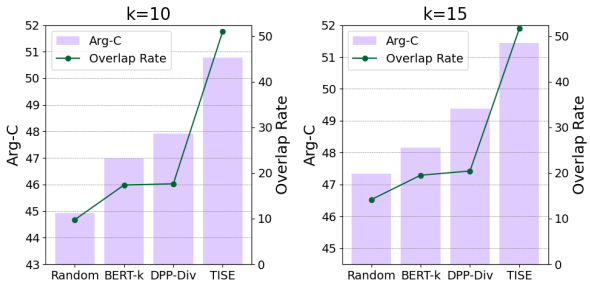


Figure 3: Relationship between EAE performance and the overlap rate on $k$=10 and $k$=15.

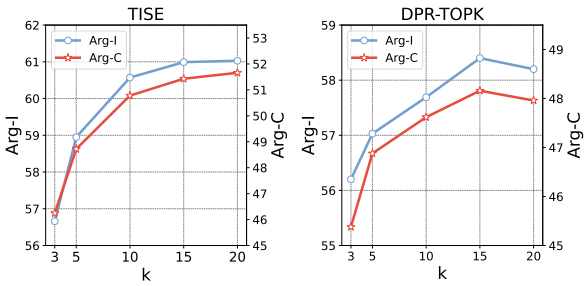


Figure 4: Performance of TISE and DPR-TOPK on different $k$. When $k$=20, the semantic retriever may exceed the input length limitation of LLM, resulting in a performance decline.

### 5.3 Baselines

We compare TISE with several baselines: RANDOM randomly selects examples from the training set without repetition. BM25 (Robertson et al., 2009) is an effective sparse retriever relies on TF-IDF. BERT-TOPK and DPR-TOPK employ BERT and the twin-tower dense retriever DPR (Karpukhin et al., 2020) as encoders, respectively, and select the top-$k$ examples. DPP-DIVERSITY satisfies Example Diversity with the help of DPP.

### 5.4 Main Results

We show the experimental results with different numbers of examples ($k$) in Table 2. As we expect, TISE achieves the state-of-the-art performance across all values of $k$, which verifies the effectiveness and robustness of our method. Furthermore, TISE achieves superior performance with fewer examples. For instance, TISE achieves 58.95%/48.72% on $k$=5, while BERT-TOPK achieves 58.34%/48.16% on $k$=15. This indicates that in scenarios with low resources or constrained input length, TISE is more powerful. For each requirement, we find that both the sparse (BM25) and dense (BERT-TOPK/DPR-TOPK) retrievers outperform RANDOM, which proves that semantic-based selection is useful for EAE. But it is uncertain which retriever is better, indicating that greater similarity does not necessarily equate to better performance. In addition, TISE outperforms DPP-DIVERSITY, which in turn outperforms semantic retrievers, confirming the effectiveness of Example Diversity and Event Correlation.

### 5.5 Principle Exploration

We further explore *how the examples selected by these requirements can help LLM extract arguments.* The supervised signals provided by the example are actually the meaning of each role in the current event type. Based on this, we hypothesize that an in-context example with a larger role overlap with the test input will yield better performance. To verify this hypothesis, we define a metric called "role overlap rate" for in-context example and conduct an experiment. Denotes each selected example is $t^* \in A^*$, the set of *not-none* roles of $t^*$ is $R^{t^*}$ and the role set of $x$ is $R^x$, we define the role overlap rate of $A^*$ as:

$$s_{A^*} = \frac{\sum_{t^* \in A^*} \sum_{r^{t^*} \in R^{t^*}, r^x \in R^x} \mathbb{I}(r^{t^*} = r^x)}{\sum_{t^* \in A^*} |R^{t^*}|} \tag{10}$$

where $\mathbb{I}(\cdot)$ is the indicator function. Notice that we only consider *not-none* roles and we count the same role of different event types because it can convey useful information (e.g., "Place" in "Life.Marry" and "Life.Be-Born" ).

The results in Figure 3 demonstrate an obvious positive correlation between EAE performance and role overlap rate, which proves that more overlapped roles can indeed help LLM extract arguments. Besides, we also find that each requirement has a distinct impact: 1) **Event Correlation** notably elevates the overlap of roles as it tends to select examples within the same event types. 2) **Semantic Similarity** also proves beneficial, as similar sentences tend to describe related events with overlapping roles. 3) **Example Diversity** contributes minimally to the overlap rate, it primarily ensures the entire subset covers different roles.

## 6 Discussion

### 6.1 Ablation Study

**Effect of Different $k$** Many works have pointed that the performance of LLM is sensitive to the

| Methods | Arg-I F1 | Arg-C F1 |
|---|---|---|
| TISE | 60.6 | 50.8 |
| w/o Event Correlation | 58.5 (2.1↓) | 47.9 (2.9↓) |
| w/o Event Role | 60.0 (0.6↓) | 50.3 (0.5↓) |
| w/o Event Type | 58.8 (1.8↓) | 49.1 (1.7↓) |
| w/o Example Diversity | 59.1 (1.5↓) | 48.7 (2.1↓) |
| w/o Correlation & Diversity | 57.3 (3.3↓) | 46.9 (3.9↓) |

Table 3: Ablation results for requirements on $k$=10.

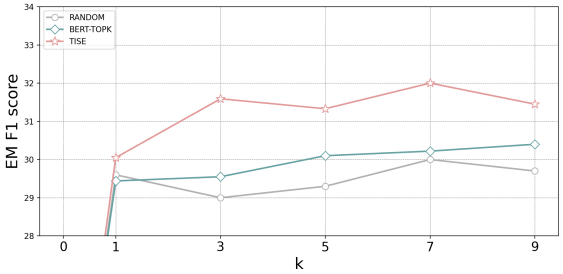


Figure 5: Performance on RAMS. Examples in document-level dataset are long, and $k$=9 will risk exceeding the input length limitation of LLM.

number of examples (Ye et al., 2023; Wang et al., 2023). Here, we evaluate the proposed model and baselines on different $k$ and report the results in Table 2. We have two conclusions: 1) As $k$ increases, the gap between the different methods also widens. The reason is that the in-context example with a small $k$ can not provide sufficient event information. However, as quantitative changes lead to qualitative changes, the quality of the examples becomes crucial and significantly impacts the performance of LLM. 2) Increasing the value of $k$ can enhance the extracted performance, but this benefit is diminishing. We provide the performance of TISE and DPR-TOPK on different $k$ in Figure 4. Both methods exhibit a monotonic increase, but eventually reach a plateau. This phenomenon is consistent with Wang et al., 2023, and we verify it is unrelated to the selection of in-context example. We speculate this phenomenon may be attributed to an upper bound on the effectiveness of ICL or a limitations imposed by the prompt template.

**Effect of Different Requirements** To verify the effectiveness of different requirements, we conduct experiments by removing Event Correlation, Example Diversity, and both of them simultaneously. Our results, detailed in Table 3, reveal that Event Correlation holds greater significance than Example Diversity, which is consistent with the conclusion drawn in principle exploration (5.5). Moreover, the joint removal of both requirements leads to a smaller decline (3.3% < 2.1% + 1.5% / 3.9% < 2.9% + 2.1%), indicating mutual reinforcement between these two requirements. For Semantic Similarity, its effectiveness and robustness have been verified in Table 2. Moreover, we also verify the event attributes. Experimental results show that the impact of Event Role (-0.6%/-0.5%) is less than Event Type (-1.8%/-1.7%), because Event Type directly filters the examples with similar event types, whereas Event Role is more likely to select irrelevant examples.

## 6.2 Analysis

**Document Level Extraction** TISE is a generalized method that can be also applied to the document-level event argument extraction. We conduct experiments on the RAMS (Ebner et al., 2020b) to verify its generality and robustness. We use F1 score as evaluation metrics based on the Exact Match (EM) criterion (Gusfield, 1997), and use a plain extraction instruction as prompt, such as *"Given a document that describes an event, you need to identify all event arguments from this document, and classify role of each argument."* Results are shown in Figure 5, TISE filters a better example set than baslines on different $k$, which in turn helps LLM to extract arguments. In addition to this, we have two other observations: 1. LLM is not sensitive to the number of displayed examples ($k$) when conducting document-level extraction, and 2. different selection methods do not improve the extraction results significantly, despite they can select examples that are more relevant to the test input. We speculate that this may be because the understanding of documents is much harder than sentences, and a better prompt is needed to help LLM understand how to extract arguments within examples.

**Adaptability** TISE can be adapted to various LLMs and prompts. To verify its adaptability, we choose `code prompt` and `code imitation prompt`[8] as prompt and `text-davinci-002` and `text-davinci-003` as LLMs. The experimental results on different $k$ are shown in Figure 6, TISE surpasses BERT-TOPK across all combinations, demonstrating the robustness of TISE in different LLMs and prompt templates. Further-

[8]Specific examples can be found in Appendix A.4.

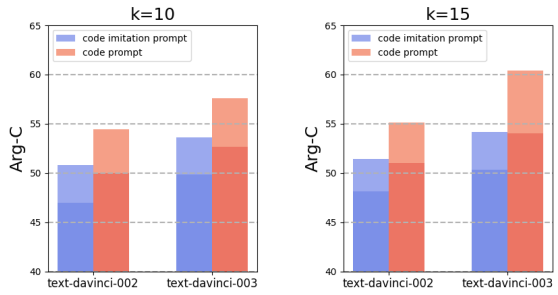


Figure 6: Performance on different LLMs and prompts. The darker part indicates the performance of BERT-TOPK, and the ligter part is the performance of TISE.

| Methods | $k$=10 | | $k$=15 | |
|---|---|---|---|---|
| | Arg-I | Arg-C | Arg-I | Arg-C |
| RANDOM | 55.71 | 45.27 | 57.64 | 46.44 |
| BERT-TOPK | 57.09 | 47.47 | 59.08 | 48.09 |
| TISE | 59.69 | 50.54 | 60.05 | 50.93 |

Table 4: Performance on the same event type scenario, note that some test inputs can not find $k$ training examples with the same event type.

more, we observe that `text-davinci-003` consistently outperforms `text-davinci-002`, and the `code prompt` yields superior results compared to the `code imitation prompt`. When the length limitation is not exceeded, the optimal combination should be TISE + `code prompt` + `text-davinci-003` on higher $k$.

**Same Event Type Scenario** If we have enough labeled data for each event type, selecting examples with the same event type would be more efficient. To verify the stability of TISE in this scenario, we conduct an experiment by restricting the selected examples to those with the same event type as the test input. The results are shown in Table 4, we find that examples with high semantic similarity remain effective even when all of them have the same event type (+1.38%/+2.20% on $k$=10 and +1.44%/+1.65% on $k$=15). Furthermore, TISE still outperforms baselines on different $k$(+2.60%/+3.07% on $k$=10 and +0.97%/+2.84% on $k$=15), we attribute this improvement to the Example Diversity and Event Role modules, which ensure TISE can select the optimal subset across different data distributions.

**Comparison with Supervised Methods** Supervised methods have achieved respectable results using limited training data through specialized architectural designs. We first compare TISE with the competitive method (DEGREE; Hsu et al., 2021) in the few shot scenario. We conduct experiment from two aspects: 1. Scaling down the training data, and TISE can only select examples from a limited amount of data. 2. TISE can select from the whole dataset and we calculate how many different training data are actually used. Results are reported in Table 5, TISE exceeds DEGREE when the total amount of available data is limited, and, despite the training data is adequate, TISE can still select a fraction of those that are truly useful to help LLM reasoning. These demonstrate that TISE can filter efficient examples based on test input so that shows capabilities in low resource scenarios. In the full shot scenario, we compare TISE with two baselines: DEGREE and DyGIE++ (Wadden et al., 2019). We are surprised to observe that TISE$_{k=15}$, coupled with `text-davinci-003` and `code prompt`, can outperform the supervised method (DyGIE++). However, there is still exists a performance gap between TISE and the DEGREE (12.6%). Hence, designing a better example selection method remains a promising direction for future research.

| | %Training Data | Arg-C |
|---|---|---|
| *few shot - aspect 1* | | |
| DEGREE | 5% | 35.5 |
| TISE$_{k=5}$ | 5% | 38.8 |
| DEGREE | 10% | 41.6 |
| TISE$_{k=15}$ | 10% | 42.0 |
| *few shot - aspect 2* | | |
| DEGREE | 20% | 46.2 |
| DEGREE | 30% | 48.7 |
| TISE$_{k=5}$ | 19% | 48.7 |
| TISE$_{k=10}$ | 28% | **50.8** |
| *full shot* | | |
| DyGIE++ | - | 60.7 |
| TISE$^{\dagger}_{k=15}$ | - | 60.9 |
| DEGREE | - | 73.5 |

Table 5: Comparison between TISE and supervised methods, the results are derived from Hsu et al., 2021. % Training data means the proportion of traning data. TISE$^{\dagger}$ uses `text-davinci-003` as LLM and `code prompt` as prompt template.

**Case Study** We present a case study in Figure 7 to directly visualize the advantages of each requirement. For Example Diversity, both examples 1 and 2 in BERT-TOPK describe a clash/war with event role "Place". As a result, LLM correctly extracts the argument for "Place", while the

| | Test Input | In-context Examples | EAE Result | Golden Label |
|---|---|---|---|---|
| BERT-TOPK | (Die) Australian commandos, who have been operating deep in Iraq, destroyed a command and control post and killed a number of soldiers , according to the country 's defense chief , Gen. Peter Cosgrove . | (Attack) ... Marines had broken up violent clashes on Wednesday in **Tikrit [Place]**, Saddam 's hometown<br>(Die) ... minimize **civilian [Victim]** casualties in the current **Iraq [Place]** war ...<br>(Transport) ... refused to say whether the paratroopers deployed directly from Italy into Iraq ... | Agent:None<br>Instrumen:None<br>Place:Iraq<br>Victim:soldiers | Agent:commandos<br>Instrument:None<br>Place:Iraq<br>Victim:number |
| TISE | | (Die) ... at dusk **they [Agent]** fired what observers say were seven artillery **rounds [Instrument]** back at their former stronghold ...<br>(Die) ... of coalition **soldiers [Victim]** had been killed, their graves were now at the **airport [Place]**.<br>(Die) ... iraqi **missile [Instrument]** hit operation **center [Place]** for 2nd brigade 3rd infantry zrigs south baghdad , at least **four [Victim]** dead , two **soldiers [Victim]** and two **journalists [Victim]**. | Agent:commandos<br>Instrument:None<br>Place:Iraq<br>Victim:number of soldiers | |

Figure 7: A case study with BERT-TOPK and TISE. The event type of each sentence is marked at the beginning using (bracket), the event information is bolded as **Argument [Role]**, red text indicates incorrect extraction and green ones indicates correct extraction. "Victim" of TISE is considered correct because the head token is matched.

other roles perform incorrect results (notice that example 2 mentions a "Victim", so the argument of "Victim" is not "None", although it is not entirely accurate). On the other hand, the examples of TISE encompass all event roles involved in the label, enabling LLM to extract arguments comprehensively, verifying Example Diversity can ensure the selected examples contain different role information. For event correlation, BERT-TOPK selects a "Transport" example that does not contain any useful event roles. In contrast, all the examples of TISE have the event type "Die", ensuring that each of them can convey useful information. Overall, both Example Diversity and Event Correlation can help LLM extract arguments better.

## 7 Conclusion

In this paper, we present three necessary requirements for the in-context example selection when using LLM to conduct EAE task. We propose TISE, which scores the examples from three perspectives and leverages DPP to fuse these scores so that the optimal in-context example is directly selected. Experiments on ACE05 show TISE can select a more efficient in-context example than baselines, is robust to the number of examples, and even can achieve better performance with fewer examples. TISE can also be adapted to different prompts and LLMs, and outperforms some fully fine-tuned supervised methods. Moreover, we explore the principle of the effectiveness of three requirements.

## Acknowledgements

This work is supported by the National Key Research and Development Program of China (NO.2022YFB3102200).

## Limitations

1) The encoder of TISE is a vanilla BERT model. Except for replacing it with a dense retriever, some methods use ranking labels obtained by the language model to supervisedly learn a retriever as an encoder that is adapted to the current dataset (Rubin et al., 2022; Ye et al., 2023). However, there is a gap between the retrieval task and the EAE task, so it is not reasonable to directly use the ranking label as a supervised signal. Intuitively, it makes more sense to use reinforcement learning to optimize a retriever using the performance of the EAE as a reward. 2) The time consumption of TISE is high, although we store the descriptions in a dictionary, which reduces the time complexity of the event scorer to O(1). However, each role in the example requires a separate query of the test roles, which increases the overhead.

## Ethics Statement

This work does not present any direct ethical issues. We focus on selecting the optimal in-context example for LLMs to conduct EAE task. All experiments are conducted on open datasets and the findings and conclusions of this paper are reported accurately and objectively.

## References


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

Laming Chen, Guoxin Zhang, and Eric Zhou. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. *Advances in Neural Information Processing Systems*, 31.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

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.

Xinya Du, Alexander M Rush, and Claire Cardie. 2021. Template filling with generative transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 909–914.

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

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

Jun Gao, Changlong Yu, Wei Wang, Huan Zhao, and Ruifeng Xu. 2023a. Mask-then-fill: A flexible and effective data augmentation framework for event extraction. *arXiv preprint arXiv:2301.02427*.

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

Dan Gusfield. 1997. Algorithms on stings, trees, and sequences: Computer science and computational biology. *Acm Sigact News*, 28(4):41–60.

I Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, et al. 2021. Degree: A data-efficient generation-based event extraction model. *arXiv preprint arXiv:2108.12724*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Chun-Wa Ko, Jon Lee, and Maurice Queyranne. 1995. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691.

Alex Kulesza, Ben Taskar, et al. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286.

Itay Levy, Ben Bogin, and Jonathan Berant. 2022. Diverse demonstrations improve in-context compositional generalization. *arXiv preprint arXiv:2212.06800*.

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.

Jiaju Lin, Qin Chen, Jie Zhou, Jian Jin, and Liang He. 2022. Cup: Curriculum learning based prompt tuning for implicit event argument extraction. *arXiv preprint arXiv:2205.00498*.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7999–8009.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114.

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.

Di Lu, Shihao Ran, Joel Tetreault, and Alejandro Jaimes. 2023. Event extraction as question generation and answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1666–1688.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098.

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.

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.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 300–309.

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.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. *ArXiv*, abs/1909.03546.

Xingyao Wang, Sha Li, and Heng Ji. 2023. Code4struct: Code generation for few-shot event structure prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3640–3663.

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*.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2022. Self-adaptive in-context learning. *arXiv preprint arXiv:2212.10375*.

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.

Bishan Yang and Tom Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. *arXiv preprint arXiv:2302.05698*.

Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Ves Stoyanov, Greg Durrett, and Ramakanth Pasunuru. 2022. Complementary explanations for effective in-context learning. *arXiv preprint arXiv:2211.13892*.

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.

# A Appendix

## A.1 Event Type Description

The complete event type description is shown in Table 7, we display the event types belonging to the same parent together to highlight the advantages.

## A.2 Event Role Description

The complete event role description is shown in Table 8. Similarly, we display roles of the same event type together.

## A.3 Theoretical Proof

Since $K_{ij} = \mathrm{k}(t_i, t_j|x) = \mathrm{s_E}'(x, t_i) \cdot \mathrm{s_T}'(x, t_i) \cdot \mathrm{s_T}(t_i, t_j) \cdot \mathrm{s_T}'(x, t_j) \cdot \mathrm{s_E}'(x, t_j)$ , the kernel matrix $K$ can be represented as:

$$K = \mathrm{Diag}(S'_E) \cdot \mathrm{Diag}(S'_T) \cdot \bar{K} \cdot \mathrm{Diag}(S'_T) \cdot \mathrm{Diag}(S'_E)$$

We only consider the effect of $\det(K_A)$:

$$\det(K_A) = \prod_{t_i \in A} {S'_{E_i}}^2 * \prod_{t_i \in A} {S_{T_i}}'^2 * \det(\bar{K}_A)$$

After logarization:

$$\begin{aligned} \mathrm{logdet}(K_A) &= \sum_{t_i \in A} \log({S'_{E_i}}^2 * {S'_{T_i}}^2) + \mathrm{logdet}(\bar{K}_A) \\ &= \sum_{t_i \in A} (\frac{S_{E_i}}{\lambda_1} + \frac{S_{T_i}}{\lambda_2}) + \mathrm{logdet}(\bar{K}_A) \end{aligned}$$

Simplifying, we assume that $A = \{t_i, t_j\}$, thus the second part can be decomposed into:

$$\det(\bar{K}_A) = \begin{vmatrix} \bar{K}_{ii} & \bar{K}_{ij} \\ \bar{K}_{ji} & \bar{K}_{jj} \end{vmatrix} = -\bar{K}_{ij}^2 + \bar{K}_{ii}\bar{K}_{jj} \quad (11)$$

It can be found that $\det(\bar{K}_A)$ and $\bar{K}_{ij} = \mathrm{s_T}(t_i, t_j)$ are negatively correlated, meaning a higher $P(A|x)$ requires lower $\bar{K}_{ij}$, in other words, $t_i$ and $t_j$ have low similarity.

## A.4 Prompt Examples

We display examples of code imitation prompt and code prompt in Figures 8 and 9, respectively. TISE only needs to fuse the selected examples into the *In-context Examples* according to the prompt format, so that can be adapted to any prompt template.

| | | Arg-C$_{k=10}$ | Arg-C$_{k=15}$ |
|---|---|---|---|
| $\lambda_1 = 0.5$ | $\lambda_2 = 0.05$ | 50.78 | 51.43 |
| | $\lambda_2 = 0.1$ | 50.01 | 51.01 |
| | $\lambda_2 = 0.5$ | 48.49 | 49.23 |
| $\lambda_2 = 0.05$ | $\lambda_1 = 0.05$ | 50.13 | 50.87 |
| | $\lambda_1 = 0.1$ | 50.54 | 50.99 |
| | $\lambda_1 = 0.5$ | 50.78 | 51.43 |

Table 6: Comparison with different $\lambda_1$ and $\lambda_2$ on $k$=10 and $k$=15.

## A.5 Impact of $\lambda$

To obtain the optimal hyperparameters, we conduct experiments with various combinations of $\lambda_1$ and $\lambda_2$ on different $k$. The experimental results are shown in Table 6, we find that $\lambda_1 = 0.5$ and $\lambda_2 = 0.05$ archives the best performance across different values of $k$. In addition, TISE is smooth on $\lambda_1$, while sensitive to $\lambda_2$. We guess this is because Semantic Similarity has a greater impact than Event Correlation.

**Entity Type Definition**

Description of base entity types:
FAC: A functional, primarily man-made structure. Facilities are artifacts falling under the domains of architecture and civil engineering, including more temporary human constructs, such as police lines and checkpoints.
… (other entity types)

**Event Definition**

Role definition of event type Attack (Parent type: Conflict):
1. attacker (need to be one of GPE or ORG or PER)
2. instrument (need to be one of VEH or WEA)
3. place (need to be one of FAC or GPE or LOC)
4. target (need to be one of FAC or LOC or ORG or PER or VEH or WEA)
5. victim (need to be one of PER)
Multiple entities can be extracted for the same role, each entity is a double-quote enclosed string.
Each extracted entity should look like: (Base Entity Type) "content of extracted string"
If entity is not present in the text, write: () ""
Different entities are delimited by a comma.
In this event: [attacker] attacked [target] hurting [victim] victims using [instrument] instrument at [place] place.

**In-context Examples**

Translate the following sentence into an instance of Attack event. The trigger word(s) of the event is marked with **trigger word**.
"reporter : experts say saddam hussein 's forces will likely try to hold out in baghdad for as long as possible without **using** the weapons his government insists it does not have , hoping to build international pressure on the u.s. and britain to back down."
1. attacker: (PER) "forces"
2. instrument: (WEA) "weapons"
3. place: (GPE) "baghdad"
4. target: () ""
5. victim: () ""
… (other examples)

**Event Instantiation**

Translate the following sentence into an instance of Attack event. The trigger word(s) of the event is marked with **trigger word**.
"Most analysts linked Russia 's opposition to a **war** in Iraq to fears that it will lose oil contracts that were sealed with the now - toppled regime of Saddam Hussein ."
1. attacker: (

Figure 8: An example of code imitation prompt.To save space, we only show one part of the *Entity Type Definition* and *In-context Examples*, the other items have the same formats. In-context example and its label are colored blue, and the test input is colored orange.

**Entity Type Definition**

```
class Entity:
    def __init__(self, name: str):
        self.name = name
class Event:
    def __init__(self, name: str):
        self.name = name
class FAC(Entity):
    """A functional, primarily man-made structure. Facilities are artifacts falling under the domains  of architecture and civil engineering, including more temporary human constructs, such as police lines and checkpoints."""
    def __init__(self, name: str):
        super().__init__(name=name)
```

**Event Definition**

```
class Conflict(Event):
    def __init__(
        self,
        attacker: List[GPE | ORG | PER] = [],
        instrument: List[VEH | WEA] = [],
        place: List[FAC | GPE | LOC] = [],
        target: List[FAC | LOC | ORG | PER | VEH | WEA] = [],
        victim: List[PER] = []):
        self.attacker = attacker
        self.instrument = instrument
        self.place = place
        self.target = target
        self.victim = victim
class Attack(Conflict):
    """self.attacker attacked self.target hurting self.victim victims using self.instrument instrument at self.place place."""
    def __init__(
        self,
        attacker: List[GPE | ORG | PER] = [],
        instrument: List[VEH | WEA] = [],
        place: List[FAC | GPE | LOC] = [],
        target: List[FAC | LOC | ORG | PER | VEH | WEA] = [],
        victim: List[PER] = []):
        super().__init__(
            attacker=attacker,
            instrument=instrument,
            place=place,
            target=target,
            victim=victim)
```

**In-context Examples**

```
"""Translate the following sentence into an instance of Attack. The trigger word(s) of the event is marked with **trigger word**.
"reporter : experts say saddam hussein ‘s forces will likely try to hold out in baghdad for as long as possible without **using** the weapons his government insists it does not have , hoping to build international pressure on the u.s. and britain to back down . " """
attack_event = Attack(attacker=[PER(“forces”)], place=[GPE(“baghdad”)], instrument=[WEA("weapons")])
```

**Event Instantiation**

```
"""Translate the following sentence into an instance of Attack. The trigger word(s) of the event is marked with **trigger word**.
"Most analysts linked Russia 's opposition to a **war** in Iraq to fears that it will lose oil contracts that were sealed with the now - toppled regime of Saddam Hussein ." """
attack_event = Attack(
```

Figure 9: An example of code prompt.To save space, we only show one part of the *Entity Type Definition* and *In-context Examples*, the other items have the same formats. In-context example and its label are colored blue, and the test input is colored orange.

| Parent Event Type | Event Type | Description |
|---|---|---|
| Movement | Transport | Involves an object movement, recording that an object has been transport to somewhere. |
| Personnel | Elect | Involves a personnel change, recording that an individual was elected to a position. |
| | Nominate | Involves a personnel change, recording that a person has been nominated for a position. |
| | Start-Position | Involves a change in personnel, recording someone starting work at a certain company. |
| | End-Position | Involves a change in personnel, recording someone stopping work at a certain company. |
| Conflict | Attack | Involves a conflict event, recording an attack or aggression. |
| | Demonstrate | Involves a conflict event, recording a military demonstration. |
| Contact | Meet | Involves a connection, recording a meeting between two people. |
| | Phone-Write | Involves a connection, recording a phone call between two people. |
| Life | Marry | Involves an event related to life, recording a marriage of two people. |
| | Injure | Involves an event related to life, recording that a person injuring another. |
| | Die | Involves an event related to life, recording the death of a person. |
| | Be-Born | Involves an event related to life, recording the birth of a newborn baby. |
| | Divorce | Involves an event related to life, recording a divorce of two people. |
| Transaction | Transfer-Money | Involves a transaction, recording a monetary transfer. |
| | Transfer-Ownership | Involves a transaction, recording a transfer of ownership of an item. |
| Business | End-Org | Involves an organizational business, recording an organization shut down. |
| | Start-Org | Involves an organizational business, recording the formation of an organization. |
| | Declare-Bankruptcy | Involves an organizational business, recording an organization declaring itself bankrupt. |
| | Merge-Org | Involves an organizational business, recording the merger of two organizations. |
| Justice | Sue | Involves a justice trial, recording a person has been sued before a court of law. |
| | Arrest-Jail | Involves a justice trial, recording a person has being arrested and jailed. |
| | Execute | Involves a justice trial, recording a person has been executed. |
| | Trial-Hearing | Involves a justice trial, recording a trial hearing. |
| | Charge-Indict | Involves a justice trial, recording a person has been charged or indicted. |

| | | |
|---|---|---|
| | Convict | Involves a justice trial, recording a person has been convicted. |
| | Sentence | Involves a justice trial, recording a person has been sentenced. |
| | Release-Parole | Involves a justice trial, recording a person has been released or paroled. |
| | Fine | Involves a justice trial, recording a person has been fined. |
| | Pardon | Involves a justice trial, recording a person has been pardoned. |
| | Appeal | Involves a justice trial, recording a person has been acquitted. |
| | Extradite | Involves a justice trial, recording a person has been extradited. |
| | Acquit | Involves a justice trial, recording a person has been acquitted. |

Table 7: The complete description of event type.

| Event Type | Role | Description |
|---|---|---|
| Life:Be-Born | Person | Who was born? |
| | Place | Where did the birth take place? |
| Life:Marry | Person | Who was married? |
| | Place | Where did the marriage take place? |
| Life:Divorce | Person | Who was divorced? |
| | Place | Where did the divorce take place? |
| Life:Injure | Agent | Who enacted the harm? |
| | Victim | Who was harmed? |
| | Instrument | What device was used to inflict the harm? |
| | Place | Where did the injuring take place? |
| Life:Die | Agent | Who was the killer? |
| | Victim | Who was killed? |
| | Instrument | What device was used to kill? |
| | Place | Where did the death take place? |
| Movement:Transport | Agent | Who is responsible for the transport event? |
| | Artifact | Who was transported? |
| | Vehicle | What vehicle was used for transporting? |
| | Origin | Where did the transporting originate? |
| | Destination | Where was the transporting directed? |
| Transaction:Transfer-Ownership | Buyer | Who is the buying agent? |
| | Seller | Who is the selling agent? |
| | Beneficiary | Who benefits from the transaction? |
| | Artifact | What was bought? |
| | Place | Where did the sale take place? |
| Transaction:Transfer-Money | Giver | Who gave money to others? |
| | Recipient | Who was given money? |
| | Beneficiary | Who benefited from the transfer? |
| | Place | Where was the amount transferred? |
| Business:Start-Org | Agent | Who started the organization? |
| | Org | What organization was started |

| | | |
|---|---|---|
| | Place | Where was the organization started? |
| Business:Merge-Org | Org | What organization was merged? |
| Business:Declare-Bankruptcy | Org | What organization declared bankruptcy? |
| | Place | Where was the bankruptcy declared? |
| Business:End-Org | Org | What organization was ended? |
| | Place | Where was the organization ended? |
| Conflict:Attack | Attacker | Who was the attacking agent? |
| | Target | Who was the target of the attack? |
| | Victim | Who was the victim of the attack? |
| | Instrument | What instrument was used in the attack? |
| | Place | Where did the attack take place? |
| Conflict:Demonstrate | Entity | Who demonstrated? |
| | Place | Where did the demonstration take place? |
| Contact:Meet | Entity | Who met with others? |
| | Place | Where did the meeting takes place? |
| Contact:Phone-Write | Entity | Who communicated with others? |
| | Place | Where did the communication take place? |
| Personnel:Start-Position | Person | Who is the employee? |
| | Entity | Who is the the employer? |
| | Place | Where did the employment relationship begin? |
| Personnel:End-Position | Person | Who ended the position? |
| | Entity | Who fired employee? |
| | Place | Where did the employment relationship end? |
| Personnel:Nominate | Person | Who was nominated? |
| | Agent | Who is the nominating agent? |
| Personnel:Elect | Person | Who was elected? |
| | Agent | Who was the voting agent? |
| | Place | Where did the election takes place? |
| Justice:Arrest-Jail | Person | Who was arrested? |
| | Agent | Who made the arrest? |
| | Place | Where did the arrest take place? |
| Justice:Release-Parole | Person | Who was released? |
| | Entity | Who released the person? |
| | Place | Where did the release take place? |
| Justice:Trial-Hearing | Defendant | Who was on trial? |
| | Prosecutor | Who tried defendant? |
| | Adjudicator | Who adjudicated the trial? |
| | Place | Where did the trial take place? |
| Justice:Charge-Indict | Defendant | Who was indicated for crime? |
| | Prosecutor | Who executed the indictment? |
| | Adjudicator | Who adjudicated the indictment? |
| | Place | Where did the indictment take place? |
| Justice:Sue | Plaintiff | Who sued defendant? |
| | Defendant | Who was sued? |
| | Adjudicator | Who adjudicated the suing? |
| | Place | Where did the suit take place? |
| Justice:Convict | Defendant | Who was convicted for crime? |
| | Adjudicator | Who convicted defendant for crime? |
| | Place | Where did the conviction take place? |
| Justice:Sentence | Defendant | Who was sentenced for crime? |

| | | |
|---|---|---|
| | Adjudicator | Who sentenced the defendant for crime? |
| | Place | Where did the sentencing take place? |
| Justice:Fine | Entity | Who was fined for crime? |
| | Adjudicator | Who fined the entity for crime? |
| | Place | Where did the fining take place? |
| Justice:Execute | Person | Who was executed for crime? |
| | Agent | Who executed person for crime? |
| | Place | Where did the execution take place? |
| Justice:Extradite | Agent | Who extradited person? |
| | Person | Who was extradited for crime? |
| | Destination | Where was the person extradited to? |
| | Origin | Where was the person extradited from |
| Justice:Acquit | Defendant | Who was acquited of crime? |
| | Adjudicator | Who acquited the defendant of crime? |
| Justice:Pardon | Defendant | Who was pardoned for crime? |
| | Adjudicator | Who pardoned defendant for crime? |
| | Place | Where did the pardon take place? |
| Justice:Appeal | Plaintiff | Who made the appeal? |
| | Adjudicator | Who adjudicated the appeal? |
| | Place | Where did the appeal take place? |

Table 8: The complete description of event role.