COFFEE: A Contrastive Oracle-Free Framework for Event Extraction

Meiru Zhang[♠] Yixuan Su[♠] Zaiqiao Meng[♠◇]
Zihao Fu[♠] Nigel Collier[♠]
[♠]Language Technology Lab, University of Cambridge
[◇]School of Computing Science, University of Glasgow
[♠]{mz468, ys484, zf268, nhc30}@cam.ac.uk
[◇]zaiqiao.meng@glasgow.ac.uk

Abstract

Event extraction is a complex task that involves extracting events from unstructured text. Prior classification-based methods require comprehensive entity annotations for joint training, while newer generation-based methods rely on heuristic templates containing oracle information such as event type, which is often unavailable in real-world scenarios. In this study, we consider a more realistic task setting, namely the Oracle-Free Event Extraction (OFEE) task, where only the input context is given, without any oracle information including event type, event ontology, or trigger word. To address this task, we propose a new framework, COFFEE. This framework extracts events solely based on the document context, without referring to any oracle information. In particular, COFFEE introduces a contrastive selection model to refine the generated triggers and handle multi-event instances. Our proposed COFFEE outperforms state-of-the-art approaches in the oracle-free setting of the event extraction task, as evaluated on two public variants of the ACE05 benchmark. The code used in our study has been made publicly available[1].

1 Introduction

The event extraction task aims to identify events and their arguments from the given textual input context (Nguyen et al., 2016; Wadden et al., 2019; Yang et al., 2019). Conventionally, this task can be decomposed into four sub-tasks (Nguyen et al., 2016): (i) detecting the trigger word that most directly describes the event; (ii) event type classification for defining its event-specific attributes; (iii) argument identification and (iv) argument classification that maps the argument entities to the corresponding role attributes based on the structure of each event type, namely event schema. For instance, Figure 1 shows the input context of an event extraction example that contains two events: a 'Transport' event triggered by the trigger word 'went' and an 'Attack' event triggered by the trigger word 'killed', where 'Transport' and 'Attack' are two event types.

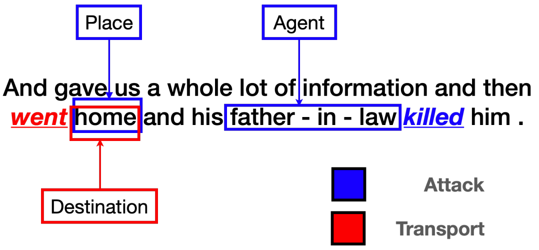

Place
Agent
And gave us a whole lot of information and then
went home and his father - in - law killed him .
Destination
🟦 Attack
🟥 Transport

Figure 1: An event extraction example with two events: **Transport** and **Attack**. In the 'Transport' event, '*went*' is the trigger word, and 'home' is the 'Destination' argument. In the 'Attack' event, '*killed*' is the trigger word while 'father-in-law' and 'home' are the 'Agent' and 'Place' arguments, respectively.

Many prior studies formulate the event extraction task as a token-level classification problem, which extracts event triggers and arguments using sequence tagging models based on tailor-designed neural networks (Nguyen et al., 2016; Liu et al., 2018; Li et al., 2019; Yang et al., 2019; Wadden et al., 2019; Huang et al., 2020; Lin et al., 2020; Nguyen et al., 2021). However, such methods cannot leverage rich label semantics since the target outputs (e.g., event triggers and arguments) are fixed tagging labels. Recently, with advances in generative pre-trained language models, several generation-based approaches (Hsu et al., 2022; Huang et al., 2022; Li et al., 2021; Zhang et al., 2021) have been applied to solve this problem. These approaches transform the event extraction task into a conditional generation task. By utilizing the autoregressive generation nature of generative pre-trained language models (e.g., BART-Gen (Li et al., 2021), DEGREE (Hsu et al., 2022)) and some manual prompts, it becomes possible to harness the semantics of labels and conduct both entity extraction and classification in an autoregressive manner simultaneously.

[1] https://github.com/meiru-cam/COFFEE

Proceedings of the First Workshop on Matching From Unstructured and Structured Data (MATCHING 2023), pages 33–44
July 13, 2023 ©2023 Association for Computational Linguistics

While impressive results are reported, we identify two major limitations of the current generation-based event extraction methods. Firstly, most of these methods rely on heuristic templates and extensive human knowledge engineering. According to the experiments conducted by Hsu et al. (2022), a slight change in the template might lead to significant performance changes, thus raising the issue of using sub-optimal templates. Secondly, most of these generation-based approaches still require certain oracle information, such as event type and event schema, which necessitate extensive manual annotations. For example, the DEGREE model's inference process, as demonstrated by Hsu et al. (2022), requires manually designed event-specific templates for each example and iterates over all event types. On the other hand, Text2Event (Lu et al., 2021) also constrains the generation with manually designed templates, which require event schema to be given. However, obtaining this oracle information, such as event type and schema, is unrealistic for a real-world inference system to achieve automatically. Hence, this paper aims to address the Oracle-Free Event Extraction (OFEE) task where only the input context is given.

In this study, we propose a novel Contrastive Oracle-Free Framework for Event Extraction (COFFEE), which addresses the event extraction task without using any oracle information. Our COFFEE consists of two parts, a generator that performs the extraction of events and a selector that aims to refine the generated results. The generator of our COFFEE generates both the candidate triggers and event arguments, where the shared generator allows for cross-task knowledge sharing between these sub-tasks. The selector of our COFFEE learns to re-rank and select the candidate triggers to obtain more accurate trigger predictions, which is inspired by (Su et al., 2021). One challenge of the sentence-level event extraction is that a sentence may contain more than one event record (Si et al., 2022; Subburathinam et al., 2019) (e.g., the example in Figure 1), and event specific templates can help the model to identify and extract events in a targeted manner. Prior approaches tackling this challenge have necessitated either multi-label tagging (Ramponi et al., 2020; Lin et al., 2020), event-specific templates (Hsu et al., 2022), or multi-turn question answering techniques (Du and Cardie, 2020; Li et al., 2020). In contrast, our proposed model can concurrently generate and select multiple event candidates, encompassing both the event trigger and its associated type, thereby effectively addressing the aforementioned challenge.

The contribution of this work is as follows:

- We highlight the challenge of the current event extraction task setting and introduce the oracle-free setting of this task that requires the model to produce the structural event without using oracle information beyond the context.
- We propose COFFEE, a novel Contrastive Oracle-Free Framework for Event Extraction which use a generator and a selector to generatively obtain structural event information from context without using any oracle information.
- We conduct experiments on two variants of the ACE05 benchmark under the oracle-free setting to evaluate our COFFEE. The results demonstrate that the template-based baselines heavily rely on the additional oracle information, whereas our COFFEE exhibits superior empirical performance over these baselines in the absence of an oracle.

2 Task Definition

Conventionally, the event extraction task entails the following terminologies (Nguyen et al., 2016; Liu et al., 2020; Paolini et al., 2021).

- *Input Context*: The input sentence or sentences that contain one or more events.
- *Trigger Word*: The main word that most clearly expresses the occurrence of an event (e.g., words 'went' and 'killed' in Figure 1).
- *Event Type*: The event type that defines the semantic structure of a specific event (e.g., events 'Transport' and 'Attack' in Figure 1).
- *Event Argument*: Event arguments identify the entities involved in events and their roles based on their relationships with the event triggers. An entity can be an object, place or person that participates in the event. For example, 'home' is an entity that serves as both the 'Place' argument of the 'Transport' event and the 'Destination' argument of the 'Attack' event in Figure 1.

Given the input context $\mathbf{c}$, which is a sequence of tokens $[c_1, \cdots, c_n]$, the conventional event extraction task aims to identify the trigger words, classify the events triggered by these words and extract the arguments in each of the events with their

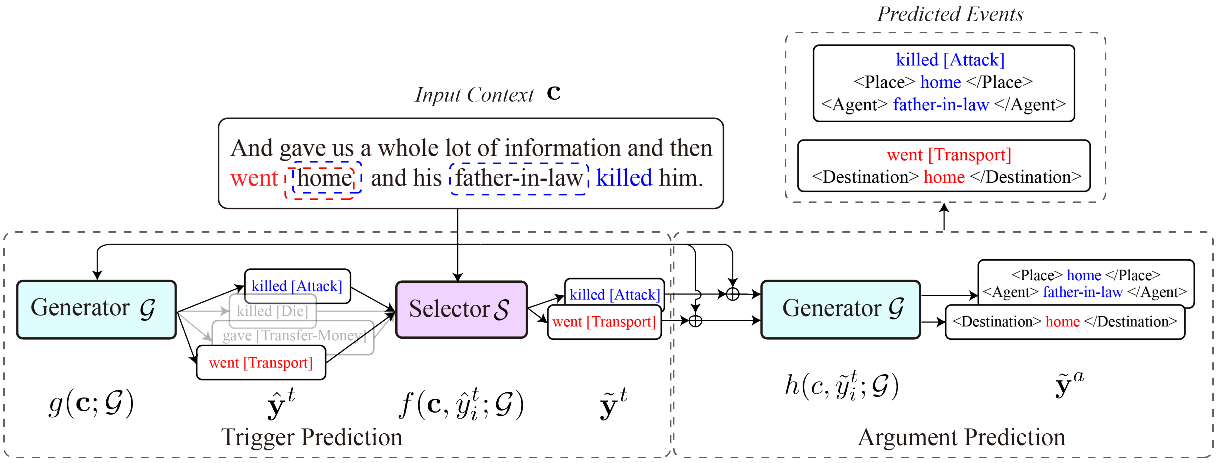


Figure 2: Overview of our proposed COFFEE framework. We train $\mathcal{G}$ to generate trigger candidates $\hat{\mathbf{y}}^t$ that contain trigger word and event type first. These trigger candidates then used to train $\mathcal{S}$ to select the final trigger predictions $\tilde{\mathbf{y}}^t$. In the argument prediction stage, the trained generator is re-used to generate arguments $\tilde{\mathbf{y}}^a$ based on $\tilde{\mathbf{y}}^t$ selected by $\mathcal{S}$. Only the input context $\mathbf{c}$ is required to predict events.

corresponding roles (Nguyen et al., 2016; Chen et al., 2015). Assume that an input sentence context $\mathbf{c}$ contains $|e|$ different events, then its ground truth triggers $\mathbf{y}^t$ can be represented as $[y_1^t, \cdots, y_{|e|}^t]$, where y_i^t denotes the i-th trigger word and event type of the given context sentence. For each event, there is a list of ground truth arguments, denoted by $\mathbf{y}^a$, which is a list of $\langle$*role*, *argument*$\rangle$ pairs, i.e., $\mathbf{y}^a = [\langle r_1, a_1\rangle, \cdots, \langle r_m, a_m\rangle]$, where a_j is the j-th entity participating in the event and r_j is the corresponding role type for that entity.

For this conventional event extraction task, the current state-of-the-art generation-based approaches rely on manual templates, which require trigger words or event types to be given, to simplify this task (Hsu et al., 2022; Lu et al., 2021). However, in a realistic scenario, although argument roles are event-specific, gold trigger words or event type information may not be readily available during event argument extraction We focus on the Oracle-Free Event Extraction (OFEE) task, which presents a more practical scenario by only providing the input context during inference. The goal of OFEE is to infer event triggers and arguments without relying on pre-defined event specific templates, making it more challenging to solve due to the absence of external guidance or oracle information.

3 Methodology

As mentioned in the task definition, our goal is to extract event frames without using any templates. This adds complexity to the generation model, particularly when dealing with contexts containing multiple events, such as the example given in Figure 1. To address the challenging OFEE task, we propose a novel contrastive framework called COFFEE, which comprises two primary components: a generator $\mathcal{G}$, responsible for generating event frames present in the provided context, and a selector $\mathcal{S}$, which re-ranks and selects the triggers generated by $\mathcal{G}$. In our proposed COFFEE framework, $\mathcal{G}$ is fine-tuned using ground truth triggers and arguments (i.e. $\mathbf{y}^t$ and $\mathbf{y}^a$) to generate candidate triggers $\hat{\mathbf{y}}^t$ and arguments $\tilde{\mathbf{y}}^a$ (see §3.1). At the inference stage, $\mathcal{S}$ is fine-tuned to refine and select final trigger predictions $\tilde{\mathbf{y}}^t$ based on the generated candidate triggers $\hat{\mathbf{y}}^t$ and gold triggers $\mathbf{y}$ on the training set (see §3.2). The final trigger predictions are forwarded to $\mathcal{G}$ for argument prediction (see §3.1). We next present the details of COFFEE's components, i.e. the generator and the selector.

3.1 Generator

The generator is fine-tuned on both trigger prediction and argument prediction simultaneously by training on the pairs of instances with different prefixes 'TriggerEvent: ' and 'Argument: ' (see §4.4). In order to take the context as input and generate structured event frames, the generator $\mathcal{G}$ of COFFEE is employed using an encoder-decoder transformer model, such as BART, T5 and mT5 (Lewis et al., 2020; Raffel et al., 2020; Xue et al., 2021). We resort T5 (Raffel et al., 2020) as the base model and encode only '[and]' and '[none]' as additional special tokens based on experimental results.

During the inference stage, we apply beam search (Jelinek, 1976) to generate candidate triggers $\hat{\mathbf{y}}^t$ and output the beam score of these triggers. Given the context $\mathbf{c}$, the generator outputs the top-l triggers with the highest beam scores, denoted by $\hat{\mathbf{y}}^t = g(\mathbf{c}; \mathcal{G})$, where $\hat{\mathbf{y}}^t$ is a list of triggers $[\hat{y}_1^t, \cdots, \hat{y}_l^t]$, with beam scores $[b_1, \cdots, b_l]$, and $\hat{y}_i^t$ represents a generated candidate trigger in the context $\mathbf{c}$. After obtaining the list of candidate triggers, we use a contrastive-learning based selector $\mathcal{S}$ (see §3.2 for details) to further re-rank the generated candidates using $f(\mathbf{c}, \hat{\mathbf{y}}^t; \mathcal{S})$ and select the final set of trigger predictions $\tilde{\mathbf{y}}^t$. The predicted trigger words are then concatenated to the context iteratively, and the generator performs argument prediction on each event using $h(\mathbf{c}, \tilde{y}_i^t; \mathcal{G})$. Specifically, given the input $[c_1, \cdots, c_n, \tilde{y}_i^t]$, G generates $\tilde{\mathbf{y}}^a = [\langle \tilde{r}_1, \tilde{a}_1 \rangle, \cdots, \langle \tilde{r}_m, \tilde{a}_m \rangle]$ for a predicted trigger $\tilde{y}_i^t$.

3.2 Selector

In our approach, we employ contrastive learning to re-rank the candidate triggers $\hat{\mathbf{y}}^t$. Contrastive learning (Chen et al., 2020) is a technique that aims to learn meaningful representations by maximizing the similarity between positive pairs while minimizing the similarity between negative pairs. In the context of our problem, we define the ground truth triggers $\mathbf{y}^t$ for context $\mathbf{c}$ as the positive anchors, while the negative samples are the other incorrect candidates generated, i.e., $\hat{y}_j^t \notin \mathbf{y}^t$.

To apply contrastive learning for re-ranking, we first encode the context and candidate triggers using a shared encoder. Specifically, given a list of candidate triggers $\hat{\mathbf{y}}^t$, for each $\hat{y}_i^t \in \hat{\mathbf{y}}^t$ we concatenate it to the context and use $\mathcal{S}$ to map the concatenated text $[\mathbf{c} : \hat{y}_i^t]$ into a real-valued ranking score by preforming linear projection $f(\mathbf{c}, \hat{y}_i^t; \mathcal{S})$. In this study, we employ RoBERTa (Liu et al., 2019) as the backbone selector model to encode the text input, and $\mathcal{S}$ predicts the ranking score for each of the candidate triggers in $\hat{\mathbf{y}}^t$ through optimizing over a contrastive objective $\mathcal{L}_\mathcal{S}$, which encourages $\mathcal{S}$ to predict higher scores for true trigger candidates and lower scores for false trigger candidates.

Formally, given a context $\mathbf{c}$ and the generated candidate triggers $\hat{\mathbf{y}}^t$, $\mathcal{S}$ is fine-tuned to optimize:

$$\mathcal{L}_\mathcal{S} = \sum_{i=1}^{|e|} \sum_{j=1}^{k} \max\{0, \rho - f(\mathbf{c}, y_i^t; \mathcal{S}) + f(\mathbf{c}, \hat{y}_j^t; \mathcal{S})\}, \quad (1)$$

where $\hat{y}_j^t \notin \mathbf{y}^t$, $\rho \in [-1, 1]$ is a pre-defined margin and k represents the number of negatives sampled from $\hat{\mathbf{y}}^t$. By taking into account the implicit correlation between the context and generated candidates, $\mathcal{S}$ captures the semantic relevance between context and correct trigger candidates, thus enhancing trigger extraction and positively impacting the performance of argument extraction.

		ACE05-E		ACE05-E+	
	# sent	# triggers	# args	# triggers	# args
train	17172	4202	4859	4419	6607
val	923	450	605	468	759
test	832	403	576	424	689

Table 1: The statistics of our used datasets.

Since the number of events in the context is unknown, we use a threshold to automatically control the number of events predicted. Let α represent the weight parameter and θ represent the threshold parameter in our model. These hyperparameters are used for combining the beam score b_i with the ranking score s_i and filtering out the false candidate triggers, respectively. We determine the threshold θ and the weight α on the development set, which is exclusively utilized for hyperparameter tuning, to ensure an unbiased evaluation on the test set. The final set of trigger predictions is defined as $\tilde{\mathbf{y}}^t = \{\tilde{y}_1^t, \cdots, \tilde{y}_{|\tilde{\mathbf{y}}^t|}^t\}$, which satisfies that $\forall \tilde{y}_i^t$,

$$\alpha \cdot \sigma(f(\mathbf{c}, \tilde{y}_i^t; \mathcal{S})) + (1 - \alpha) \cdot \sigma(b_i) > \theta, \quad (2)$$

where σ denotes the softmax function.

4 Experiments

4.1 Dataset

In this work, we evaluate our COFFEE based on a public event extraction benchmark ACE05 (Walker and Consortium, 2005), which consists of 599 English documents, 33 event types, and 22 argument roles. Building upon previous works (Wadden et al., 2019; Lin et al., 2020) that split and preprocess this dataset, we use two variants for the event extraction dataset, namely **ACE05-E** and **ACE05-E+**. Detailed split and statistics of the two datasets can be found in Table 1.

4.2 Evaluation Metrics

The evaluation of trigger identification, event type classification, argument identification, and argument role classification tasks utilizes the F1-score metric, consistent with the previous studies (Zhang et al., 2019; Wadden et al., 2019). A correct trigger classification prediction requires accurate trigger word and event type prediction, i.e., $\tilde{y}_i^t = y_i^t$. Correct argument identification necessitates accurate classification of the event type and argument entity, while a correct argument role classification demands accurate identification of the argument and role type prediction. Specifically, a predicted event type $\tilde{t}_e$, argument $\tilde{a}$, and role type $\tilde{r}$ are considered correct if $(\tilde{a}, \tilde{r}, \tilde{t_e}) = (a, r, t_e)$.

4.3 Baselines

To validate the effectiveness of our proposed method, we compared our COFFEE with five state-of-the-art baselines:

- **OneIE** (Lin et al., 2020) is a joint neural model that simultaneously extracts entities and relations using a dynamic relation graph.
- **Text2Event** (Lu et al., 2021) is a sequence-to-structure controlled generation model with constrained decoding for event extraction. It focuses on the structured generation that uses event schema to form event records.
- **BART-Gen** (Li et al., 2021) is designed for document-level event extraction that can deal with the long-distance dependence issue and coreference problem. Constrained generation is applied for argument extraction that requires event-specific templates.
- **DEGREE** (Hsu et al., 2022) is a generative event extraction approach that highly relies on the designed template.
- **TANL** (Paolini et al., 2021) is a model that extracts event triggers and arguments by so called augmented translation that embeds target outputs into the context sentence.

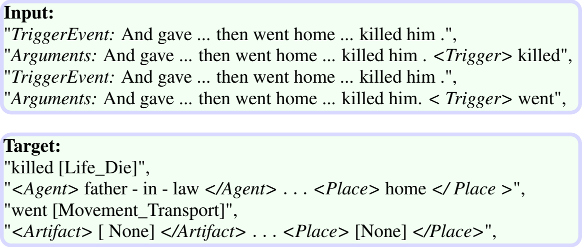
4.4 Implementation

We preprocess the data by separating original samples into event samples and inserting placeholders for target entities. The instances are processed with distinct prefixes for subtasks: 'TriggerEvent: ' and 'Arguments: '. Figure 3 shows a data preprocessing example. Details pertaining to our pipeline training and inference process, including specifics about the two-stage fine-tuning, such as the learning rate and batch size, as well as the beam search strategy employed during inference, are elaborated in Appendix A.1.

5 Results

5.1 OFEE performance

As described in Section 4.3, Text2Event, BART-Gen and DEGREE utilize different oracle information. To compare the performance of our COFFEE framework with these methods under the OFEE setting, we implemented the following adaptations to these baseline approaches:



Input:
"*TriggerEvent:* And gave ... then went home ... killed him .",
"*Arguments:* And gave ... then went home ... killed him . <*Trigger*> killed",
"*TriggerEvent:* And gave ... then went home ... killed him .",
"*Arguments:* And gave ... then went home ... killed him. < *Trigger*> went",

Target:
"killed [Life_Die]",
"<*Agent*> father - in - law </*Agent*> . . . <*Place*> home </ *Place* >",
"went [Movement_Transport]",
"<*Artifact*> [None] </*Artifact*> . . . <*Place*> [None] </*Place*>",

Figure 3: Example of input and target for the model.

- Text2Event (Lu et al., 2021) relies on a complex constrained decoding mechanism that depends on the event schema. For the oracle-free setting, we utilized the default decoding of the T5 model to generate results.
- BART-Gen (Li et al., 2021) adopts a constrained generation mechanism, which necessitates the use of templates. We removed the template and the constrained decoding, thereby enabling the model to function. The trigger extraction performance of BART-Gen is not reported in our study due to an implementation error stemming from different preprocessing methods, which prevented us from applying this approach to the ACE-05E+ dataset. Consequently, we depended on the ground truth triggers for argument extraction in this instance.
- The DEGREE (Hsu et al., 2022) model is designed to generate 'invalid' instances during both the training and inference phases, wherein event-specific knowledge is combined with context even if no such event is mentioned in the context. We eliminated these event-specific templates, leaving only the context sentence as input.

As presented in Table 2, we report F1 scores of the compared methods over four sub-tasks described in 4.1, namely trigger identification, trigger classification, argument identification, and argument classification. We observe the following:

- Firstly, it is crucial to highlight that the oracle-free setting poses a more challenging scenario. When all oracle information is removed, generation-based baselines relying on templates exhibit a varying degree of performance decline on both datasets (↓ 0.5% to 37.42% in argument classification). Although DEGREE is effective with the oracle information, it struggles to filter

	Model	ACE-05E				ACE-05E+			
		Trig I	Trig C	Arg I	Arg C	Trig I	Trig C	Arg I	Arg C
	OneIE	76.83	73.05	57.26	54.31	77.31	74.01	56.66	54.29
	Text2Event$^{\natural}$	73.93	69.06	51.59	49.52	73.40	68.99	52.64	50.39
	BARTGen$^{\natural}$	74.36	71.13	55.22	53.71	-	-	66.62	64.28
	DEGREE$^{\natural\diamondsuit\flat}$	74.57	70.96	56.03	53.41	74.90	70.30	55.74	53.61
Oracle Free	Text2Event	73.49	68.60	51.24	49.32	72.73	68.30	52.48	50.35
	BARTGen	70.96	66.59	48.47	46.36	-	-	51.43	47.49
	DEGREE	43.64	$2.18^{\dagger}$	28.54	25.99	54.32	$2.26^{\dagger}$	30.09	28.79
	TANL	**81.10**	**77.09**	55.28	52.16	**80.28**	**76.03**	54.56	52.57
	COFFEE	79.61	75.73	**59.88**	**55.43**	78.28	74.70	56.87	54.11
	+*TANL*	**81.10**	**77.09**	58.74	55.24	**80.28**	**76.03**	**59.78**	**57.06**

Table 2: Performance comparison of COFFEE and SOTA generation-based approaches. † The trigger classification F1 of DEGREE is nearly zero because the model cannot exclude the negative samples constructed without a template.$^{\natural}$, $^{\diamondsuit}$, and $^{\flat}$ denote the model that requires a manually designed template, example keywords, and event description, respectively. The highest results are in **bold** and the second highest results are underlined.

out the 'invalid' events in the oracle-free setting, resulting in an almost zero (2.18%) trigger classification F1. This indicates that the information leaked in the template significantly contributes to the performance of DEGREE.

- Our proposed COFFEE outperforms the classification-based approach OneIE and the generation-based approaches Text2Event, BARTGen, and DEGREE in both the presence and absence of oracle information across **all four metrics**. This demonstrates that our COFFEE can effectively leverage the input context to extract event frames.
- In comparison to TANL, our COFFEE achieves similar results in trigger extraction, with a difference of only 1.36%. One possible explanation is that the threshold-based method results in a smaller recall value due to more false positives. However, our model possesses robust argument extraction capabilities and attained superior performance in argument extraction with these extracted triggers (↑ 3.33% and ↑ 2.46% on ACE-05E and ACE05E+, respectively). These findings corroborate the effectiveness of the shared generator on trigger and argument prediction.

5.2 Ablation study

We conducted an ablation study on the threshold and weight parameters to demonstrate the effectiveness of our selector $\mathcal{S}$ and the influence of these parameters on the COFFEE performance.

Figure 4 illustrates the effect of the threshold parameter on COFFEE. The threshold determines the minimum score a candidate must achieve to be selected. Increasing the threshold results in fewer candidates being selected but with higher accuracy. Conversely, an overly high threshold could filter out some of the correct candidates, decreasing performance. The optimal threshold value is 0.2, which achieves the best performance on all four subtasks.

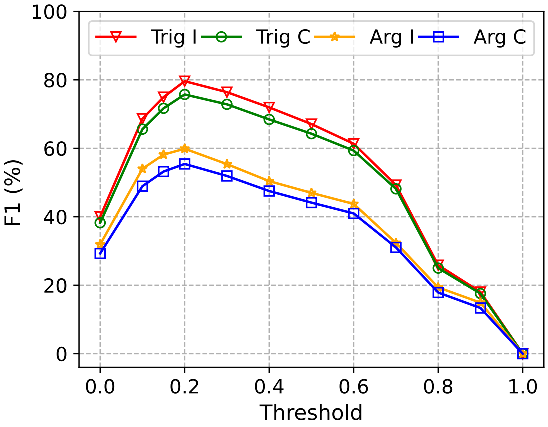


Figure 4: Effect of threshold in COFFEE framework.

In addition, Figure 5 demonstrates the influence of the weight parameter on COFFEE. The weight represents the ratio of combining the ranking score and generation score. When the weight is set to 0, only the generation score is considered, while a weight of 1 means that only the ranking score is considered. As depicted in Figure 5, the best extraction performance is achieved with a fixed threshold and an optimal weight value of $\alpha = 0.4$. The initial improvement in the F1 score with increasing weight suggests that the ranking score can effectively refine the results of the beam search.

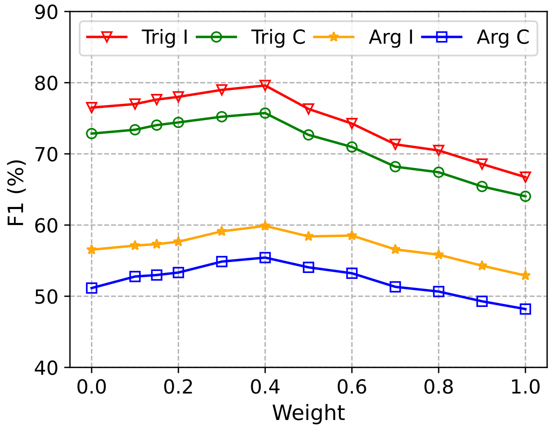


Figure 5: The influence of the weight α on performance.

However, the ranking scores exhibit significant variations, leading to a corresponding fluctuation in softmax probability as the weight increases. As the final probability becomes increasingly reliant on the ranker probability, fewer candidates are selected at the same threshold, resulting in a decline in performance.

5.3 Qualitative Case Analysis

In order to demonstrate the ability of our model to select event candidates, we analyze the results of two instances selected from the test set. For comparison, we select COFFEE without ranking and TANL, given its high performance. As shown in Table 3, our proposed model successfully extracts the missing events not detected by the baselines. The re-ranking mechanism enables the model to select more accurate candidates.

In particular, only COFFEE successfully predicts all the events within the context. In Example 1, both TANL and COFFEE without ranking fail to extract **E1**, triggered by ‘pay’, suggesting that the baselines may have difficulty identifying complex event triggers. In this case, there is not a specific amount of money to be paid, but a mention of cost. In Example 2, TANL fails to extract **E2**, which is triggered by ‘becoming’, and COFFEE without ranking fails to extract **E1**, highlighting the inability of the baselines to identify events and their corresponding arguments consistently. In contrast, our COFFEE successfully identifies the events and extracts the target arguments, demonstrating its superior performance.

Comparing COFFEE with and without ranking, we can conclude that re-ranking in the selector is crucial. In both examples, COFFEE fails to detect all events without re-ranking. Even though both candidates are the correct targets, the beam scores differ more than expected, which leads to incorrect ranking. The re-ranking can increase the probability of the second candidate and thus allowing it to be selected under the chosen threshold.

These examples demonstrate the improvements in event extraction offered our selector $\mathcal{S}$, which allows the framework to re-rank and select the correct triggers for multi-event instances, outperforming the baselines and establishing our model as a more effective and reliable solution for OFEE tasks.

6 Related Work

6.1 Event Extraction

Early event extraction research primarily relied on rule-based methods involving hand-written patterns to identify event triggers and arguments in text (Li et al., 2013, 2015). Supervised machine learning techniques became popular, with various feature-based classification models employed (Hsi et al., 2016). However, these methods faced limitations due to manual feature engineering and the need for large annotated datasets. Researchers then turned to deep learning approaches, utilizing convolutional neural networks (CNNs) (Chen et al., 2015; Nguyen and Grishman, 2015; Björne and Salakoski, 2018; Yang et al., 2019), recurrent neural networks (RNNs) (Nguyen et al., 2016), and Tree-LSTM (Li et al., 2019) for event extraction, which automatically learned relevant features and improved performance.

The introduction of pre-trained language models revolutionized event extraction. Fine-tuning these models achieved state-of-the-art performance across various benchmarks (Lin et al., 2020; Ramponi et al., 2020; Wadden et al., 2019; Yang et al., 2021). These models captured deep contextual information and benefited from knowledge transfer, enhancing performance with limited annotated data. Some studies framed event extraction as a multi-turn question answering task (Du and Cardie, 2020; Li et al., 2020; Liu et al., 2020; Zhou et al., 2021), while others approached it as a sequence-to-sequence generation task (Hsu et al., 2022; Lu et al., 2021; Li et al., 2021). Although effective, these methods heavily relied on manually designed prompts and templates, except for Text2Event (Paolini et al., 2021), which depended solely on context information. In contrast, our work focuses on oracle-free event extraction and addresses the task via generation without tem-

	Example 1	
Context	Kommersant business daily joined in , declaring in a furious front - page headline : " The United States is demanding that Russia , France and Germany pay for the Iraqi war .	
Reference	**E1**: pay [Transfer-Money]	**Args**: [Giver] Germany
	E2: war [Attack]	**Args**: [Place] Iraqi
TANL + COFFEE	**E1**:	**Args**:
	E2: war [Attack]	**Args**: [Place] Iraq
COFFEE w/o Ranker	**E1**:	**Args**:
	E2: war [Attack]	**Args**: [Place] Iraq
COFFEE	**E1**: pay [Transfer-Money]	**Args**: [Giver] Germany
	E2: war [Attack]	**Args**: [Place] Iraq
	Example 2	
Context	Welch specifically is seeking performance evaluations , correspondence between his estranged wife and partners while she worked at the law firm 's office in London , and documents related to her prospects of becoming a partner .	
Reference	**E1**: correspondence [Phone-Write]	**Args**: [Entity] partners; [Place] office
	E2: becoming [Start-Position]	**Args**: [Entity] firm
TANL + COFFEE	**E1**: correspondence [Phone-Write]	**Args**: [Entity] partners; [Place] office
	E2:	**Args**:
COFFEE w/o Ranker	**E1**:	**Args**:
	E2: becoming [Start-Position]	**Args**:
COFFEE	**E1**: correspondence [Phone-Write]	**Args**: [Entity] partners; [Place] office
	E2: becoming [Start-Position]	**Args**:

Table 3: Event extraction examples from the test set using COFFEE, COFFEE without ranking and TANL+COFFEE. The triggers and arguments missed by the baselines but captured by COFFEE are highlighted. It is evident that COFFEE is generally more effective in detecting the events.

plates. The most recent studies have focused on event detection or event argument extraction separately (Zhang et al., 2022; Huang et al., 2022; Ma et al., 2022), which is not directly comparable to our study as we consider the complete event extraction process.

6.2 Post-Generation Ranking

Post-generation re-ranking is usually applied in two-stage systems, that is, generation and re-ranking, to re-score the output from the first stage by training an additional re-ranking module. This technique has been widely used in neural translation and summarization. For example, Ng et al. (2019); Yee et al. (2019) re-score and select the best hypotheses using Noisy Channel Modeling to improve translation quality. Zhong et al. (2020) formulate the summarization as text matching and re-ranks the summary candidates based on similarity score. Liu and Liu (2021) introduce an additional scoring model with contrastive training to predict the score of generated summaries. Both methods utilize a margin-based ranking loss that initializes candidates with orders. For the event trigger selection, we assume that the beam score is not a reliable indicator and consequently treat the candidates equally. Su et al. (2021) use a contrastive re-ranking module with hinge loss to select prototypes for a table-to-text generation. To the best of our knowledge, our work is the first to focus on enhancing the oracle-free generation-based event extraction models using re-ranking.

7 Conclusion

In this work, we study a more realistic setting of the event extraction task, namely the oracle-free event extraction, where no additional information beyond the context is required for event inference. To address this task, we propose a generation-based event extraction framework called COFFEE. Our COFFEE introduces a contrastive selector to improve trigger extraction performance by re-ranking and automatically determining the number of triggers to be selected in a given context. Additionally, we investigate the dependence of current generation-based models on extra knowledge, such as designed event-specific templates, event trigger keywords, and event descriptions. Our results show that this reliance on templates and human-designed trigger sets is unnecessary, and a pure oracle-free model applied directly can perform very well on general event extraction. In the future, we plan to extend sentence-level event extraction to document-level and explore zero-shot settings to handle the emergence of unseen events.

8 Limitations

Despite its promising results, our study has limitations. Our model primarily works with English text, limiting its applicability to other languages. Its focus on sentence-level extraction doesn't consider document context, which could be investigated in future research. The employed training dataset is relatively small, potentially not encompassing all possible event types, thus affecting the model's performance and generalizability. Additionally, our two-stage inference framework, while enhanced by a ranking module, is prone to error propagation. If a trigger isn't identified in the first stage, its associated arguments cannot be extracted. Future work should address these issues for improved performance and broader applicability.

9 Ethics Statement

In preparing and submitting this research paper, we affirm that our work adheres to the highest ethical standards and is devoid of any ethical issues. The study presented in the manuscript was conducted in a manner that respects the principles of academic integrity, transparency, and fairness.

References

Jari Björne and Tapio Salakoski. 2018. Biomedical event extraction using convolutional neural networks and dependency parsing. In *Proceedings of the BioNLP 2018 workshop*, pages 98–108, Melbourne, Australia. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

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

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.

Andrew Hsi, Yiming Yang, Jaime Carbonell, and Ruochen Xu. 2016. Leveraging multilingual training for limited resource event extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1201–1210, Osaka, Japan. The COLING 2016 Organizing Committee.

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.

Kuan-Hao Huang, I-Hung Hsu, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. Multilingual generative language models for zero-shot cross-lingual event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4633–4646, Dublin, Ireland. 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.

Frederick Jelinek. 1976. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64(4):532–556.

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

Diya Li, Lifu Huang, Heng Ji, and Jiawei Han. 2019. Biomedical event extraction based on knowledge-driven tree-LSTM. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1421–1430, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

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.

Xiang Li, Thien Huu Nguyen, Kai Cao, and Ralph Grishman. 2015. Improving event detection with Abstract Meaning Representation. In *Proceedings of the First Workshop on Computing News Storylines*, pages 11–15, Beijing, China. Association for Computational Linguistics.

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, Online. 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, Yubo Chen, Kang Liu, and Jun Zhao. 2018. Event detection via gated multilingual attention mechanism. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4865–4872. AAAI Press.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

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.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Minh Van Nguyen, Viet Dac Lai, and Thien Huu Nguyen. 2021. Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 27–38, Online. Association for Computational Linguistics.

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

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with 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 2: Short Papers)*, pages 365–371, Beijing, China. Association for Computational Linguistics.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

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

Jinghui Si, Xutan Peng, Chen Li, Haotian Xu, and Jianxin Li. 2022. Generating disentangled arguments with prompts: A simple event extraction framework that works. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6342–6346. IEEE.

Yixuan Su, Zaiqiao Meng, Simon Baker, and Nigel Collier. 2021. Few-shot table-to-text generation with prototype memory. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 910–917, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. Cross-lingual structure transfer for relation and event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 313–325, Hong Kong, China. Association for Computational Linguistics.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv preprint*, abs/1910.03771.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. 2021. Document-level event extraction via parallel prediction networks. 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 6298–6308, Online. Association for Computational Linguistics.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy. Association for Computational Linguistics.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5696–5701, Hong Kong, China. Association for Computational Linguistics.

Junchi Zhang, Yanxia Qin, Yue Zhang, Mengchi Liu, and Donghong Ji. 2019. Extracting entities and events as a single task using a transition-based neural model. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5422–5428. ijcai.org.

Ningyu Zhang, Hongbin Ye, Shumin Deng, Chuanqi Tan, Mosha Chen, Songfang Huang, Fei Huang, and Huajun Chen. 2021. Contrastive information extraction with generative transformer. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3077–3088.

Senhui Zhang, Tao Ji, Wendi Ji, and Xiaoling Wang. 2022. Zero-shot event detection based on ordered contrastive learning and prompt-based prediction. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2572–2580, Seattle, United States. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

Yang Zhou, Yubo Chen, Jun Zhao, Yin Wu, Jiexin Xu, and Jinlong Li. 2021. What the role is vs. what plays the role: Semi-supervised event argument extraction via dual question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14638–14646.

A Appendices

A.1 Implementation Details

Our pipeline training comprises two stages: generation model fine-tuning and re-ranking model fine-tuning. The T5-base model (Raffel et al., 2020) fine-tuning is achieved through the HuggingFace Transformers library (Wolf et al., 2019) on an RTX3090 GPU, using an AdamW optimizer (Loshchilov and Hutter, 2017), with a learning rate of 0.0001 with a decay schedule of 1e-5. We set a batch size of 8 and maximum input/output sequence lengths at 650/200.

For inference, we generate candidate triggers using a beam search strategy with 10 beams. These candidates are then re-ranked and filtered by the selector $\mathcal{S}$, based on optimal thresholds and weights derived through grid search.

In the second stage, we fine-tune a RoBERTa-base model (Liu et al., 2019) for re-ranking. This stage reduces the maximum input length to 512 and sets the number of negative candidates for contrastive learning to 5, with a learning rate of 0.005.

Upon refining triggers, they are concatenated to the context and reintroduced to the generator for argument generation via a greedy search. The final extraction of entities and roles is conducted using regular expressions. Our COFFEE generator and selector take approximately 4 hours and 2 hours to train, respectively.

Performance evaluation of the trigger and argument extraction is based on regular expressions to detect entities extracted from the placeholders. The results can be observed in Table 2.