# TacoERE: Cluster-aware Compression for Event Relation Extraction

**Yong Guan**[1], **Xiaozhi Wang**[1], **Lei Hou**[1*], **Juanzi Li**[1], **Jeff Pan**[2],
**Jiaoyan Chen**[3], **Freddy Lecue**[4]

[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]School of Informatics, The University of Edinburgh, UK
[3]Department of Computer Science, The University of Manchester, UK
[4]INRIA, France
{gy2022, wangxz20}@mails.tsinghua.edu.cn, {houlei, lijuanzi}@tsinghua.edu.cn

## Abstract

Event relation extraction (ERE) is a critical and fundamental challenge for natural language processing. Existing work mainly focuses on directly modeling the entire document, which cannot effectively handle long-range dependencies and information redundancy. To address these issues, we propose a clusTer-aware compression method for improving Event Relation Extraction (TacoERE), which explores a compression-then-extraction paradigm. Specifically, we first introduce document clustering for modeling event dependencies. It splits the document into intra- and inter-clusters, where intra-clusters aim to enhance the relations within the same cluster, while inter-clusters attempt to model the related events at arbitrary distances. Secondly, we utilize cluster summarization to simplify and highlight important text content of clusters for mitigating information redundancy and event distance. We have conducted extensive experiments on both pre-trained language models, such as RoBERTa, and large language models, such as ChatGPT and GPT-4, on three ERE datasets, i.e., MAVEN-ERE, EventStoryLine and HiEve. Experimental results demonstrate that TacoERE is an effective method for ERE.

**Keywords:** Event Relation Extraction, Compression-then-Extraction, Large Language Model

## 1. Introduction

Event Relation Extraction (ERE) aims to predict relations, such as causal and subevent relations, between event mentions or trigger words in a document (Fan et al., 2022). As shown in Figure 1, given a document with event mentions/trigger words, an ERE model is expected to predict relations among the three mentioned events, such as cyclone(**e1**) $\xrightarrow{subevent}$ originated(**e2**) $\xrightarrow{precondition}$ reached(**e4**). ERE can not only facilitate deep understanding of text (Wang et al., 2020), but also benefit various downstream tasks, such as question answering (Khashabi et al., 2018) and information retrieval (Pang et al., 2020).

With the widespread adoption of deep neural networks in natural language processing (NLP), event relation extraction systems have undergone a paradigm shift to supervised neural models that encode the document as a clue for predicting relations (Cao et al., 2021; Xu et al., 2021; Chen et al., 2022). However, there are still two challenges **long-range dependencies** and **information redundancy**. Specifically, *long-range dependencies* indicates that events may be scattered across multiple sentences potentially far away from each other. In such context, existing ERE methods have difficulties capturing the dependencies among events. Consider example in Figure 1, events cyclone

(**e1**) and destroyed (**e7**), located respectively in sentence S1 and S11, are related with a cause relation. *Information redundancy* refers to the existence of information non relevant for relation prediction. For example, identifying the relation between events originated (**e2**) and reaching (**e3**) only depends on the sentences S2 and S3, while sentences S9, S10, and S11 are non relevant for identifying the relation.

To tackle these challenges, the major approaches currently are to select sentences (Wang et al., 2020; Man et al., 2022) or remove sentences (Xu et al., 2022). But these methods do not completely eliminate long range dependencies, or presence of irrelevant information at finer granularities (e.g., clauses/phrases). As shown in Figure 1, for long-range dependencies, deletes sentences S6, S7, and S8, but the distance between sentences S1 and S11 remains considerable, indicating that the dependency on sentence distance remains present. For information redundancy, the sub-sentence of sentence S2, "*tenth hurricane, and fifth major hurricane of the season*", is still useless for predicting the relation between originated (**e2**) and reaching (**e3**) from human understanding. This motivates our hypothesis that compression via summarization might be a better strategy than sentence filtering.

In this paper, we propose TacoERE, a clusTer-aware compression method for improving Event Relation Extraction, which explores a compression-
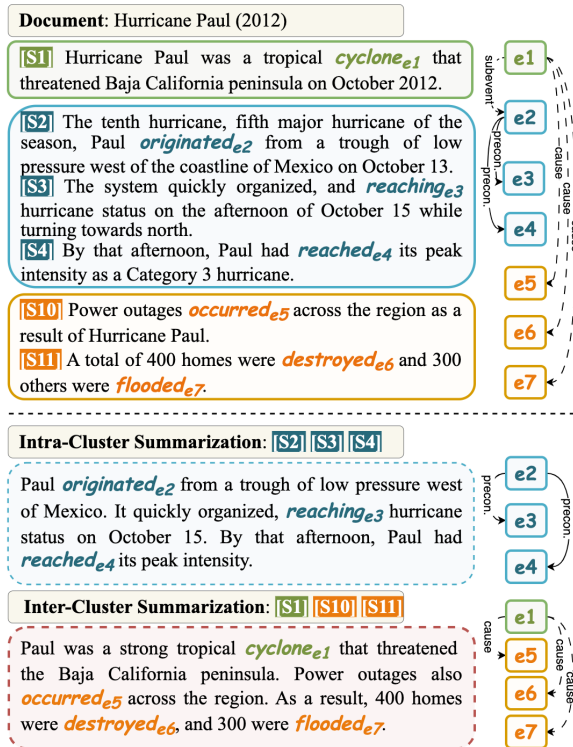
---

*Corresponding author

Figure 1: An example form MAVEN-ERE. Words in bold italics are trigger words of events. [S*i*] denotes the *i*-th sentence index, and precon. is the abbreviation of precondition. The solid and dashed arrows indicate relations within and among clusters, respectively. Different colored sentences represent different clusters.

then-extraction paradigm to extract event relations. Specifically, TacoERE first uses document clustering to split the document into intra- and inter-clusters[1], where intra-clusters aim to enhance the relations within the same cluster, while inter-clusters attempt to model the related events at arbitrary distances. For instance, events `cyclone` (**e1**) in sentence S1 and `destroyed` (**e7**) in sentence S11 belong to different clusters, and combining these two clusters can effectively reduce the event distance and facilitate modelling event relations. In this way, all dependencies between related events of any distance in the document can be modelled. Following that, cluster summarization is adopted to generate summaries for every cluster, which can further simplify and highlight important content. At last, the generated summaries of intra- and inter-clusters are utilized to predict relations.

---

[1]Document generally utilizes multiple sub-topics to organise the content (Hearst and Plaunt, 1993). As shown in Figure 1, the three sub-topics, event background, development process, and resulting impact, collectively constitute the entire article. For simplified understanding, we use the term "cluster" to represent sentences within the same sub-topic.

For evaluation, we respectively validate our ideas on both small-scale pre-trained language models (PLMs), such as RoBERTa, and large language models (LLMs), such as ChatGPT[2] and GPT-4 (OpenAI, 2023), and conduct extensive experiments on three ERE datasets, namely, MAVEN-ERE (Wang et al., 2022), EventStoryLine (Caselli and Vossen, 2016) and HiEve (Glavas et al., 2014). Experimental results demonstrate that our approach can effectively improve the performance of event relation extraction models. Our contributions are summarized as follows:

- We propose a novel cluster-aware compression method for event relation extraction, namely, TacoERE, which explores a compression-then-extraction paradigm to extract relations.

- We utilize document clustering to split the document into intra- and inter-clusters to allow the modeling of dependencies without any reliance on event distance. We propose cluster summarization to simplify and spotlight important text content to mitigate the impact of information redundancy and event distance.

- Extensive experiments have been conducted on both PLMs, such as RoBERTa, and LLMs, such as ChatGPT and GPT-4, on three ERE datasets, i.e., MAVEN-ERE, EventStoryLine and HiEve. Our TacoERE outperforms existing methods, especially on LLMs, with improvements by 11.2% and 9.1% on ChatGPT and GPT-4 respectively.

## 2. TacoERE

We consider a document $D = \{s_1, s_2, ..., s_n\}$ with $n$ sentences, annotated with event mentions/trigger words $E = \{e_1, e_2, ..., e_m\}$. The task of event relation extraction is to, given an annotated document $D$ and a pre-defined relation set $\mathcal{R}$, predict relations between all event pairs $\{(e_i, e_j)\}$ in $D$.

In this paper, we propose TacoERE, a novel cluster-aware compression method for improving event relation extraction, exploring a compression-then-extraction paradigm to extract relations. Figure 2 shows the overview of our framework: we first present *Document Compression*, including *Document Clustering* (Section 2.1) and *Cluster Summarization* (Section 2.2). *Document Clustering* splits document into intra- and inter-clusters to allow the modeling of dependencies without considering event distance among sentences. *Cluster Summarization* encourages model to simplify and spotlight important content of clusters for mitigating
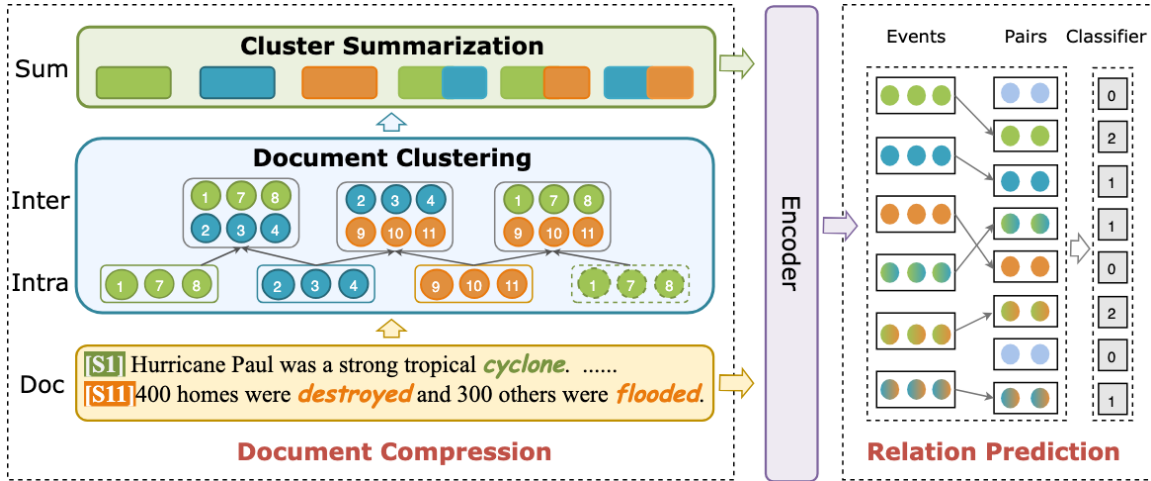
---

[2]https://chat.openai.com/chat

Figure 2: Model structure of TacoERE.

information redundancy and event distance. We then describe *Relation Prediction* in Section 2.3, which utilizes the content from *Cluster Summarization* to predict relations. Last, in Section 2.4, we employ a reinforce algorithm that jointly optimizes the *Cluster Summarization* and *Relation Prediction*. Before joint training, we introduce pretraining module for *Cluster Summarization* with event chains to teach model to do better content representation.

## 2.1. Document Clustering

The document clustering aims to split document into intra- and inter-clusters, where intra-clusters aim to enhance the relations within the same cluster, while inter-clusters attempt to model the related events at arbitrary distances. Our observation is that predicting the relations for event pairs relies on a limited content rather than the entire document. For example in Figure 1, just based on sentence S2 and S3, the relation between event `originated` (**e2**) and `reaching` (**e3**) can be deduced. Moreover, sentences in document with the same cluster are more probably to involve event relations, such as the cluster of development process with blue color (sentences S2-S4) in Figure 1 has three relations. As a result, we split the $D$ into $K$ mutually independent intra-clusters at sentence level. To obtain the intra-clusters, arbitrary clustering methods, such as traditional machine learning or deep neural networks, can be applied. In our experiments, we directly utilize the effective and widely used K-means algorithm (Guan et al., 2022; Rakib et al., 2020). Specifically, we extract multiple features to enhance the clustering, including cluster words extracted by LDA model, trigger words, tf-idf, and sentence representation encoded by RoBERTa.

$$J = \sum_{i=1}^{k} \sum_{j=1} p_{ij} ||v_j - u_i||^2 \qquad (1)$$

where $J$ is objective, $v_j$ is the feature of sentence $s_j$, $u_i$ is center of *i*-th cluster, and $p_{ij}$ is indicator.

Beyond individual intra-cluster, among different intra-clusters also occur with event relations which can be seen in Figure 1, relation between event `cyclone` (**e1**) in background and `destroyed` (e7) in impact. Thus, we fuse any two intra-clusters as inter-clusters. The order of sentences in intra- and inter-clusters will follow their original order in $D$. And the intra- and inter-clusters can involve all possible relations in $D$.

## 2.2. Cluster Summarization

The cluster summarization aims to simplify and highlight important text content of clusters for mitigating information redundancy and event distance. Intra- and inter-clusters directly group the sentences with related sub-topics. It will inevitably contain redundant information and ignore coherence, which may hinder the performance (Gao et al., 2021). Text summarization as a technique can effectively simplify and spotlight important text content, which would solve this problem to some extent. To this end, we further utilize a summarization model to generate summaries $C^a$ and $C^r$ for intra- and inter-clusters respectively, where $\{C^a, C^r\} \in C$. In this paper, we utilize a transformer-based (Vaswani et al., 2017) encoder-decoder framework for summarization. In particular, a pretrained language model as the encoder to learn the contextual representation of input, and a transformer-based decoder is utilized to generate its summary word-by-word.

## 2.3. Relation Prediction

The relation prediction aims to construct the event relation prediction process by giving the text content from cluster summarization. We first need

to obtain the contextual representation of each token in the document $D$, $C^a$, and $C^r$ respectively. Take $D$ as an example, we leverage pre-trained language model RoBERTa (Liu et al., 2019) as the encoder. Since event mention/trigger words perhaps contain multiple words, i.e., *take place*, and individual word in document may be split into subwords by wordpiece, i.e., word "*summarization*" will be split into three sub-words "*sum*", "*mar*", and "*ization*", we adopt LogSumExp pooling method (Zhou et al., 2021) over all its mentions (sub-words) embeddings in the last encoding layer as the event representation. Then, we can obtain the overall representation of event pairs in $D$. For cluster summaries $C^a$ and $C^r$, we extract the events which appear in original document $D$.

Event pairs of $C^a$ and $C^r$ all appear in $D$, which means one event pair may occur multiple times. Different from existing work (Adhikari et al., 2019a; Beltagy et al., 2020) which aggregate all identical event representations to get the final event pair representations, for the same event pair, we select the most relevant one, and the detailed selection process is as follows: (1) for the same event pair, the priority of event pair in $C^r$ is higher than $C^a$; (2) for the event pair which not in $C^a$ and $C^r$, we directly use the representation in $D$. Finally, a two layers feed-forward network with softmax is adopted to learn the classes probability based on the event pair representations. For training objective of relation prediction $\mathcal{L}_{rp}$, we use cross-entropy function as follows:

$$\mathcal{L}_{rp} = -\sum_{i \neq j} \sum_{\mathcal{R}} \{r_{ij} \log P_{ij} + (1 - r_{ij}) \log(1 - P_{ij})\} \tag{2}$$

where $P_{ij}$ is the probabilities between events $e_i$ and $e_j$, and $r_{ij} \in \mathcal{R}$.

## 2.4. Training Phase

The Training Phase describes both *Joint Training* and *Pretraining for Cluster Summarization*, where *Joint Training* aims to jointly optimize the cluster summarization and relation prediction, while *Pretraining for Cluster Summarization* aims to teach model to do better content representation.

**Joint Training** As the performance cannot be directly back-propagated to the cluster summarization process, we employ a reinforce algorithm (Williams, 1992) that treats the event relation prediction performance as reward function for the cluster summaries to train cluster summarization process. Besides, we also consider another event chains information from clusters and its summaries to enrich the overall reward function $\mathcal{R}(C)$.

*Performance-based Reward* $\mathcal{R}^{per}(C)$ aims to use the performance as the direct training signal. We calculate the reward $\mathcal{R}^{per}(C)$ based on the per-

formance for all document event pairs. In particular, for each event pair $(e_i, e_j)$, $\mathcal{R}^{per}(C) = 1$ if relation prediction model calculates the true relations between $e_i$ and $e_j$, and 0 otherwise.

*Summary-based Reward* $\mathcal{R}^{ec}(C)$ aims to encourage the correlation between the clusters and its summaries to train cluster summarization. The motivation of $\mathcal{R}^{ec}(C)$ is that summary expresses important information about the text and they are consistent at the semantic level (Guan et al., 2021b). In Section 2.1, we split document into intra- and inter-clusters. Intra-clusters are independent of each other and cover the content of the entire document. Thus, we only adopt the intra-clusters and its summaries to calculate the reward $\mathcal{R}^{ec}(C)$. Similar to existing work Pasunuru and Bansal (2018); Paulus et al. (2018), we utilize the popularly known automatic evaluation metric for summarization, ROUGE (Lin, 2004), as reward function. However, this metric mainly focuses on phrase matching/n-gram overlap while assuming equal contributions from each word. Addressing these issues, we introduce salient function to give higher weight to the trigger words.

$$P = \frac{\sum_k (LCS_\cup(C_k^a, T_k^a) + \sum_l \eta(w_{kl}))}{|D|} \tag{3}$$

$$R = \frac{\sum_k (LCS_\cup(C_k^a, T_k^a) + \sum_l \eta(w_{kl}))}{\sum_k |C_k^a|} \tag{4}$$

$$\mathcal{R}^{ec}(C) = \frac{(1 + \sigma^2)RP}{R + \sigma^2 P} \tag{5}$$

where $C_k^a$ is the summary of $k$-th intra-cluster $T_k^a$, $LCS_\cup(\cdot)$ is the union longest common subsequence, $\sigma$ is defined in Lin (2004), and $\eta(\cdot)$ is function to measure whether $w_{kl}$ is a trigger word.

After obtaining reward $\mathcal{R}^{per}(C)$ and $\mathcal{R}^{ec}(C)$, the overall reward can be calculated as $\mathcal{R}(C) = \alpha \mathcal{R}^{per}(C) + \beta \mathcal{R}^{ec}(C)$, where $\alpha$ and $\beta$ are trade-off parameters. Following existing work (Man et al., 2022), we minimize the negative expected reward $\mathcal{R}(C)$ over the possible choices of summaries:

$$\mathcal{L}_{sum} = -\mathbb{E}_{C' \sim P(C'|e_i, e_j, D)}[\mathcal{R}(C')] \tag{6}$$

Then, the gradient can be further formalised by utilizing the one roll-out sample method.

$$\nabla \mathcal{L}_{sum} = -(\mathcal{R}(C) - \theta) \nabla \log P(C|e_i, e_j, D) \tag{7}$$

where $\theta$ is used to reduce variance.

**Pretraining for Cluster Summarization** This module aims to teach model to do better content representation for cluster summarization process with event chains before joint training phase. As presented in Section 2.2, we utilize summarization model to compress the different clusters for predicting the event relations. However, abstractive summarization method always generates new words.

How to ensure generated summaries contain the events in the original document is a key problem. Inspired by (Narayan et al., 2021, 2022), we utilize the event chains which order the events in the summary as an intermediate summary representation to better guide the summarization generation. In particular, we concatenate the event chain with the corresponding summary as a unified sequence, such as "*[EVENTCHAIN] originated | organized | reaching | reached | ... [SUMMARY] paul originated from a trough of low pressure ...*". During the decoder phase, the model must generate both the event chain followed by the summary. The model structure is similar to the summarization model in Section 2.2, and we select the existing abstractive summarization datasets, such as CNN/DailyMail, for the pretraining.

## 3. Experiments

In this section, we first introduce experiment setup (Section 3.1), and then report the results and analysis (Section 3.2 - 3.5). Furthermore, we also conduct experiments on LLMs in Section 3.6.

### 3.1. Experiment Setup

**Datasets** We conduct experiments on three datasets: MAVEN-ERE (Wang et al., 2022), EventStoryLine (Caselli and Vossen, 2016) and HiEve (Glavas et al., 2014). MAVEN-ERE is a unified large-scale human-annotated ERE dataset, which contains 4,480 documents, 112,276 events, 57,992 causal relations, and 15,841 subevent relations in total. We split the data into train, dev, and test sets in 2,913, 710, and 857 documents as in Wang et al. (2022). EventStoryLine (i.e., version 0.9) contains 258 documents, 22 topics, 5,334 events, and 5,655 causal relations. Following existing works (Gao et al., 2019; Tran Phu and Nguyen, 2021), we use the last two topics as development set, and the other 20 topics are verified with a 5-fold cross-validation. For HiEve, the dataset contains 100 documents, 3,185 events, and 3,648 subevent relations. Similar to Zhou et al. (2020); Wang et al. (2020), we split the 100 documents into 60 training, 20 validation, and 20 testing.

**Evaluation Details** We adopt similar settings to Wang et al. (2022), which are closer to real situations, yet more challenging, from three aspects: (1) we consider the relation directions for relation prediction. In addition, causal/subevent relations may be defined in several sub-relations. For example, the causal relation in MAVEN-ERE has two sub-relations "*CAUSE*" and "*PRECONDITION*"; (2) we report the overall score rather than the individual sub-relation score; (3) we do not down-sample the negative instances.

| Methods | P | R | F1 |
|---|---|---|---|
| MAVEN-ERE | | | |
| BERT | 31.6 | 28.2 | 29.9 |
| RoBERTa | 33.8 | 29.5 | 31.5 |
| Hierarchical | 31.8 | 29.2 | 30.6 |
| SIEF | 33.6 | 30.8 | 32.3 |
| TacoERE (PLMs) | **34.8** | **32.4** | **34.1** |
| EventStoryLine | | | |
| BERT | 30.3 | 9.4 | 12.8 |
| RoBERTa | 31.1 | 10.7 | 14.4 |
| Hierarchical | 30.1 | 10.2 | 13.1 |
| SIEF | 32.4 | 11.3 | 14.8 |
| SCS-EERE | 32.7 | 10.9 | 15.1 |
| TacoERE (PLMs) | **32.9** | **12.3** | **16.4** |

Table 1: Model performance of causal relation on MAVEN-ERE and EventStoryLine.

Following the existing works (Gao et al., 2019; Zhou et al., 2020; Tran Phu and Nguyen, 2021), we use standard Precision (P), Recall (R), and F1-score (F1) as the metrics.

**Training Details** For cluster summarization, we utilize BERT as the document encoder, and a transformer-based framework as the decoder with 2 layers and 8 attention heads. For document clustering, the value $K$ of intra-cluster is a hyper parameter and we set $K = 3$ with the best performance for the experiment. For relation prediction, we adopt RoBERTa as the document encoder. The implementation of BERT and RoBERTa are based on the pytorch version from HuggingFace Transformers library [3]. We adopt adam optimizer with learning rate of 5e-4. The trade-off parameters of $\alpha$ and $\beta$ are set to 1.0 and 0.1 respectively. Each training and testing process is running on two NVIDIA GeForce RTX 3090 GPU.

For LLMs implementation, we use the model API provided by OpenAI, where "gpt-4", "gpt-3.5-turbo", and "text-davinci-003" refer to models GPT-4, ChatGPT, and Text-Davinci-003 respectively.

**Baselines** The following models, including small-scale PLMs and LLMs, have been compared in our experiments.

For small-scale PLMs, we first select *BERT* (Devlin et al., 2019) and *RoBERTa* (Liu et al., 2019). In addition, we also experiment with three strong and relevant models: (1) *Hierarchical* (Adhikari et al., 2019b) which uses a pretrained model to encode different chunks of the document, and sets an additional BILSTM model to aggregate representations; (2) *SIEF* (Xu et al., 2022) which proposes to randomly remove the useless sentences for prediction;

---

[3] https://github.com/huggingface

| Methods | P | R | F1 |
|---|---|---|---|
| MAVEN-ERE | | | |
| BERT | 27.5 | 24.7 | 26.8 |
| RoBERTa | 29.8 | 25.6 | 27.5 |
| Hierarchical | 28.4 | 25.4 | 27.1 |
| SIEF | 30.2 | 26.4 | 28.7 |
| TacoERE (PLMs) | **31.8** | **28.9** | **30.6** |
| HiEve | | | |
| BERT | 19.8 | 15.2 | 16.3 |
| RoBERTa | 20.2 | 16.1 | 17.8 |
| Hierarchical | 21.4 | 17.3 | 16.7 |
| SIEF | 21.8 | 17.4 | 18.6 |
| SCS-EERE | 20.6 | **19.7** | 19.2 |
| TacoERE (PLMs) | **22.6** | 19.5 | **20.8** |

Table 2: Model performance of subevent relation on MAVEN-ERE and HiEve.

(3) *SCS-EERE* (Man et al., 2022) which selects a sentence set for each event pair for prediction.

The evaluated LLMs include: (1) *GPT-4* which is an advanced and improved iteration of the GPT series, demonstrating human-level performance and significant enhancements in various aspects; (2) *ChatGPT* which is an advanced conversational AI model, is able to provide contextually relevant and coherent responses aligned with human expectation; (3) *Text-Davinci-003* which is a variant of GPT-3.5 series, offering improved performance over GPT-3 through further instruction tuning.

## 3.2. Overall Results

We report the results of causal relation on MAVEN-ERE and EventStoryLine, respectively. Subevent relation performance is evaluated on MAVEN-ERE and HiEve, respectively.

**Performance of causal relation** The results are shown in Table 1, TacoERE (PLMs) means the implementation of our compression-then-extraction method with PLMs. Experimental results demonstrate that TacoERE (PLMs) outperforms all the baselines on both MAVEN-ERE and EventStoryLine datasets. We also have the following four observations: (1) compared with pretrained model BERT, TacoERE (PLMs) achieves 4.2% improvements of F1-score on MAVEN-ERE, and 3.6% on EventStoryLine data; (2) compared with Hierarchical, which models different chunks of document, TacoERE (PLMs) achieves improved performance. This validates the effectiveness of cluster summarization, which can further mitigate information redundancy and event distance; (3) compared with SIEF, we note that TacoERE (PLMs) achieves better performance, even though SIEF is designed
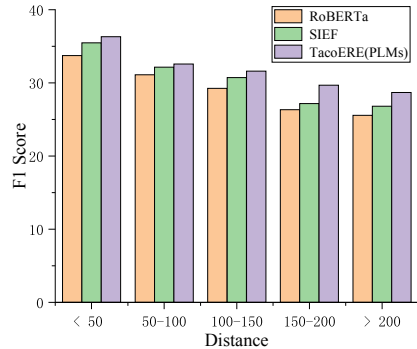


Figure 3: Model performance on different distance between related events (measured in #words).

for modeling the important content by removing sentences from the original document. This indicates that our method equipped with compression-then-extraction can effectively alleviate the problem of long-range dependencies; (4) SCS-EERE performs better than Hierarchical and SIEF. It adopts a straightforward idea that selects a set of sentences for each event pair. However, this might cost too much computation time when applied to large-scale datasets, and we directly predict all event relations in a document at one time.

**Performance of subevent relation** Table 2 presents the detailed results on MAVEN-ERE and HiEve, and our method again achieves better performance than all the baselines. Our method improves upon the pretrained model BERT by 3.8% and 4.5% in terms of F1-score on both datasets, respectively. In all, such performance on Table 1 and 2 clearly demonstrates the benefits of compression-then-extraction on event relation extraction.

## 3.3. Impact on Event Distance

To better understand the contributions of our method on long-range dependencies, we show the performance under different word distances between related events in Figure 3.

Compared with existing methods, our method achieves a certain improvement in dealing with long-range dependencies. We can also find: (1) as the word distance continuously increases, the improvement of our method over the baselines shows an overall monotonic upward trend; (2) the overall F1-score on event pairs with long distance is much lower than that of short ones, indicating the challenge of long-range dependency; (3) particularly, compared with the two baselines which directly models the entire document (RoBERTa), or compresses the document by removing irrelevant sentences (SIEF), our method obtain a larger improvement, especially when the word distance of related events is greater than or equal to 4, which

| Methods | EP | P | R | F1 |
|---|---|---|---|---|
| OneSum | 27.3 | 32.3 | 30.4 | 31.8 |
| AvgSum | 50.4 | 33.7 | 31.4 | 32.5 |
| TacoERE (PLMs) | 78.1 | 34.8 | 32.4 | 34.1 |

Table 3: Model performance on different document compression strategies. EP is the ratio of events which has relations in summaries to document.

| Methods | MAVEN-ERE | | |
|---|---|---|---|
| | P | R | F1 |
| TacoERE (PLMs) | 34.8 | 32.4 | 34.1 |
| w/o intra-clusters | 32.4 | 30.9 | 32.3 |
| w/o inter-clusters | 32.7 | 31.2 | 32.8 |
| w/o summarization | 31.8 | 31.3 | 31.9 |

Table 4: Ablation study.

can further indicate that our TacoERE (PLMs) can better handle the long-range dependencies.

### 3.4. Document Compression Evaluation

In this section, we try different strategies to obtain the cluster and evaluate the performance to verify the effectiveness of our document compression. We consider the following two strategies: (1) "*OneSum*" means only generating one summary for each document; (2) "*AvgSum*" refers to splitting document evenly into chunks based on sentence, and then generating summary for each chunk.

The results are shown in Table 3, and we have the following three observations: (1) compared with the other two methods, i.e., *OneSum* and *AvgSum*, our proposed TacoERE (PLMs) achieves the best performance; (2) *AvgSum* obtains second-best results. However, the chunks are independent to each other in their setting, which means they ignore the relation information among chunks. This may prevent it from achieving better performance; (3) for the metric EP, *OneSum* only preserves 27.3% events of the document, which is due to the fact that summary mainly focuses on the important content of the document and the length is relatively short compared to the input document. As a whole, model performance gradually gets better as more events are preserved.

### 3.5. Ablation Study

In addition to the document compression strategy, we also conduct experiments to ablate intra-clusters (*w/o intra-clusters*), inter-clusters (*w/o inter-clusters*) and cluster summarization (*w/o summarization*) to understand their contributions. The results are shown in Table 4. We can observe that: (1) compared the results of *w/o (without) intra-clusters* with *w/o inter-clusters*, we can find that the overall model works better. The use of intra- and inter-clusters helps the model to better understand the event relations both within and among the sentences; (2) the performance of TacoERE drops on the three variations, which proves that both variations have contributed to the overall performance; (3) removing *w/o intra-clusters* causes a sharp performance drop compared to *w/o inter-clusters*.

### 3.6. Evaluation on LLMs

The aforementioned series of experiments has highlighted the efficacy of our TacoERE, based on PLMs, in enhancing ERE performance. To further validate its effectiveness and robustness, we conduct extensive experiments on LLMs. Illustrated in Figure 2, our framework primarily comprises three key components: Document Clustering, Cluster Summarization, and Relation Prediction. We introduce TacoERE (LLMs), which directly leverages LLMs to implement these three components. To thoroughly validate our approach, we configure the task to enable relation prediction between event pairs individually. For testing, we randomly sample 50 documents from MAVEN-ERE, resulting in 646 causal relations. We compare TacoERE (LLMs) with three variants: (1) *Document*, which involves utilizing the entire document to predict relations; (2) *Sentence Pair*, which entails using sentences containing the event pairs to predict relations; (3) *Document Clustering*, which involves using sentences within a cluster to predict relations.

The results are shown in Table 5, we notice that our TacoERE (LLMs) achieves the best performance across all three models, with improvements of 11.2% and 9.1% on ChatGPT and GPT-4, respectively. We also have the following four observations: (1) from model perspective, GTP-4 achieves the highest F1 score of 41.9%, followed by ChatGPT; (2) compared with *Document*, *Sentence Pair* obtains better results, indicating that predicting relation between events does not depend on the whole document; (3) compared with *Document* and *Sentence Pair*, *Document Clustering* achieves improved results, indicating that our method can reduce redundant information while retaining useful information for relation prediction; (4) compared with *Document Clustering*, our TacoERE (LLMs) achieves the best performance, suggesting that reducing redundant information and shortening event distance can further facilitate the improvement of performance.

### 3.7. Case Study

We display a case in Figure 4 to quantitatively analyze the model prediction by our TacoERE (LLMs)

| Methods | Text-Davinci-003 | | | ChatGPT | | | GPT-4 | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| Document | 13.8 | 6.2 | 8.5 | 21.7 | 32.2 | 25.9 | 27.1 | 41.5 | 32.8 |
| Sentence Pair | 21.9 | 7.1 | 10.7 | 24.3 | 31.2 | 27.3 | 33.4 | 38.6 | 35.7 |
| Document Clustering | 17.3 | 8.1 | 10.9 | 24.6 | 32.9 | 28.2 | 31.9 | 47.1 | 38.1 |
| TacoERE (LLMs) | **30.2** | **8.9** | **13.8** | **31.3** | **45.6** | **37.1** | **38.9** | **45.5** | **41.9** |

Table 5: Model performance of causal relation on different LLMs. Experiments are under 2-shot setting.

---

**Document:** [S1] The Harrow and Wealdstone rail crash was a three-train <Event> **collision** </Event> at Harrow and Wealdstone station in Wealdstone , Middlesex during the morning rush hour of 8 October 1952 ; 112 were killed and 340 injured; it remains the worst peacetime rail crash in the UK.
[S2] An overnight express train from Perth crashed at speed into the rear of a local passenger train standing at a platform at the station .
[S3] The wreckage blocked adjacent lines and was struck within seconds by a `` double-headed " express train travelling north at .
[S4] The Ministry of Transport report on the crash found that the driver of the Perth train had passed a caution signal and two danger signals before colliding with the local train .
[S6] The accident <Event> **accelerated** </Event> the introduction of Automatic Warning System – by the time the report on the accident had been published , British Railways had agreed to a five-year plan to install the system to give drivers an in-cab audible and visual warning of being about to pass a distant signal at caution.

Text-Davinci-003: ❌   ChatGPT: ❌   GPT-4: ❌

**Ground Truth:** PRECONDITION

**Sentence Pair:** [S1], [S6]

Text-Davinci-003: ❌   ChatGPT: ❌   GPT-4: ❌

**Topic Grouping:** [S1], [S2], [S3], [S6]

Text-Davinci-003: ❌   ChatGPT: ❌   GPT-4: ✅

**TacoERE (LLMs):** The Harrow and Wealdstone rail crash, a devastating <Event> **collision** </Event> involving three trains, occurred at Harrow and Wealdstone station. The crash resulted in the loss of 112 lives and injuries to 340 individuals, making it the worst peacetime rail crash in the UK. The accident prompted the <Event> **accelerated** </Event> introduction of the Automatic Warning System, ensuring caution when approaching distant signals.

Text-Davinci-003: ❌   ChatGPT: ✅   GPT-4: ✅

Figure 4: Case analysis of relation prediction.

---

and different comparison modules, such as *Document*, *Sentence Pair*, and *Document Clustering*. TacoERE (LLMs) means we use the content from *Cluster Summarization* to predict relations. We can see, for each event pair, the prediction does not rely on the whole document. Some dependency information may be ignored by using *Document Clustering* for prediction. Our proposed TacoERE (LLMs) with compression-then-extraction is an effective means to enhance event relation extraction.

## 4. Related Work

### 4.1. Event Relation Extraction

Event Relation Extraction is a challenging task in natural language processing, especially for events that are scattered in different sentences (Gao et al.,

2019; Chen et al., 2022). Recently, deep learning based methods are becoming the mainstream (Cao et al., 2021; Xu et al., 2021), and extensive explorations have been made, such as joint reasoning methods which extract multiple relations simultaneously (Ning et al., 2018; Han et al., 2019), and graph-based methods which use event mentions as nodes and model document as graph (Tran Phu and Nguyen, 2021; Fan et al., 2022; Guo et al., 2023). However, these works use document as input, which cannot well handle the long-range dependency problem. In contrast, we improve event relation extraction by processing document in advance with cluster-aware compression.

Currently, a series of LLMs have been developed, such as GPT series, LaMDA (Thoppilan et al., 2022), and PaLM (Chowdhery et al., 2022), and

have achieved remarkable performance in various fields. Among them, GPT series, i.e., GPT-4 and ChatGPT, is undoubtedly the most popular work. Thus, to verify the effectiveness of our method, we conduct extensive experiments on them.

### 4.2. Controlled Text Summarization

With the development of deep learning and the increasing demand for generation quality, increasing studies are focusing on controlled text summarization (Dou et al., 2021; Guan et al., 2021a), such as designing copy mechanism to directly copy important word from input (See et al., 2017), extracting actual fact triples for modeling (Cao et al., 2017), extracting templates from the training data to guide the summarization generation (Wang et al., 2019). However, these methods typically design an additional module or using existing third-party content selectors. On the contrary, our method adopts the events in document as intermediate representation to better guide the summary generation.

## 5. Conclusion

In this paper, we propose a novel cluster-aware compression method for event relation extraction, namely, TacoERE, which explores a compression-then-extraction paradigm to extract event relations. TacoERE first splits document into intra- and inter-clusters to allow the modeling of dependencies without considering event distance among sentences. Then, cluster summarization is adopted to simplify and highlight the important text of clusters for further mitigating information redundancy and event distance. Extensive experiments have been conducted on both small-scale PLMs such as RoBERTa, and LLMs such as ChatGPT and GPT-4. Experimental results demonstrate that our proposed TacoERE with compression-then-extraction is an effective method for augmenting event relation extraction.

## 6. Bibliographical References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019a. Docbert: Bert for document classification. *arXiv preprint arXiv:1904.08398*.

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy J. Lin. 2019b. Docbert: Bert for document classification. *ArXiv*, abs/1904.08398.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Pengfei Cao, Xinyu Zuo, Yubo Chen, Kang Liu, Jun Zhao, Yuguang Chen, and Weihua Peng. 2021. Knowledge-enriched event causality identification via latent structure induction networks. In *Proceedings of ACL-IJCNLP*, pages 4862–4872.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2017. Faithful to the original: Fact aware neural abstractive summarization. *In Proceedings of AAAI*, 32.

Meiqi Chen, Yixin Cao, Kunquan Deng, Mukai Li, Kun Wang, Jing Shao, and Yan Zhang. 2022. ERGO: Event relational graph transformer for document-level event causality identification. In *Proceedings of COLING*, pages 2118–2128.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pages 4171–4186.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of NAACL*, pages 4830–4842.

Chuang Fan, Daoxing Liu, Libo Qin, Yue Zhang, and Ruifeng Xu. 2022. Towards event-level causal relation identification. In *Proceedings of SIGIR*, page 1828–1833.

Lei Gao, Prafulla Kumar Choubey, and Ruihong Huang. 2019. Modeling document-level causal structures for event causal relation identification. In *Proceedings of NAACL*, pages 1808–1817.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of ACL-IJCNLP*, pages 3816–3830.

Renchu Guan, Hao Zhang, Yanchun Liang, Fausto Giunchiglia, Lan Huang, and Xiaoyue Feng. 2022. Deep feature-based text clustering and its explanation. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3669–3680.

Yong Guan, Shaoru Guo, Ru Li, Xiaoli Li, and Hongye Tan. 2021a. Frame semantic-enhanced sentence modeling for sentence-level extractive text summarization. In *Proceedings of EMNLP*, pages 4045–4052.

Yong Guan, Shaoru Guo, Ru Li, Xiaoli Li, and Hu Zhang. 2021b. Frame semantics guided network for abstractive sentence summarization. *Knowledge-Based Systems*, 221:106973.

Shaoru Guo, Chenhao Wang, Yubo Chen, Kang Liu, Ru Li, and Jun Zhao. 2023. EventOA: An event ontology alignment benchmark based on FrameNet and Wikidata. In *Findings of the ACL*, pages 10038–10052.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint event and temporal relation extraction with shared representations and structured prediction. In *Proceedings of EMNLP-IJCNLP*, pages 434–444.

Marti A. Hearst and Christian Plaunt. 1993. Subtopic structuring for full-length document access. In *Proceedings of the SIGIR*, page 59–68.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2018. Question answering as global reasoning over semantic abstractions. In *In Proceedings of AAAI*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

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 arXiv:1907.11692*.

Hieu Man, Nghia Trung Ngo, Linh Ngo Van, and Thien Huu Nguyen. 2022. Selecting optimal context sentences for event-event relation extraction. *In Proceedings of AAAI*, 36(10):11058–11066.

Shashi Narayan, Gonçalo Simões, Yao Zhao, Joshua Maynez, Dipanjan Das, Michael Collins, and Mirella Lapata. 2022. A well-composed text is half done! composition sampling for diverse conditional generation. In *Proceedings of ACL*, pages 1319–1339.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with learned entity prompts for abstractive summarization. *In Proceedings of TACL*, 9:1475–1492.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018. Joint reasoning for temporal and causal relations. In *Proceedings of ACL*, pages 2278–2288.

OpenAI. 2023. Gpt-4 technical report.

Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xueqi Cheng, and Jirong Wen. 2020. Setrank: Learning a permutation-invariant ranking model for information retrieval. In *Proceedings of SIGIR*, pages 499–508.

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. In *Proceedings of the NAACL*, pages 646–653.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *Proceedings of the ICLR*.

Md Rashadul Hasan Rakib, Norbert Zeh, Magdalena Jankowska, and Evangelos Milios. 2020. Enhancement of short text clustering by iterative classification. In *Natural Language Processing and Information Systems*, pages 105–117.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of ACL*, pages 1073–1083.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Minh Tran Phu and Thien Huu Nguyen. 2021. Graph convolutional networks for event causality identification with rich document-level structures. In *Proceedings of NAACL*, pages 3480–3490.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *In Proceedings of NeurIPS*, 30.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020. Joint constrained learning for event-event relation extraction. In *Proceedings of the EMNLP*, pages 696–706.

Kai Wang, Xiaojun Quan, and Rui Wang. 2019. BiSET: Bi-directional selective encoding with template for abstractive summarization. In *Proceedings of ACL*, pages 2153–2162.

Xiaozhi Wang, Yulin Chen, Ning Ding, Hao Peng, Zimu Wang, Yankai Lin, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022. MAVEN-ERE: A unified large-scale dataset for event coreference, temporal, causal, and subevent relation extraction. In *Proceedings of the EMNLP*, pages 926–941.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256.

Wang Xu, Kehai Chen, Lili Mou, and Tiejun Zhao. 2022. Document-level relation extraction with sentences importance estimation and focusing. In *Proceedings of NAACL*, pages 2920–2929.

Wang Xu, Kehai Chen, and Tiejun Zhao. 2021. Discriminative reasoning for document-level relation extraction. In *Findings of ACL-IJCNLP*, pages 1653–1663.

Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020. Temporal common sense acquisition with minimal supervision. In *Proceedings of ACL*, pages 7579–7589.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. *In Proceedings of AAAI*, 35(16):14612–14620.

## 7.   Language Resource References

Caselli, Tommaso and Vossen, Piek. 2016. *The Storyline Annotation and Representation Scheme (StaR): A Proposal*. In Proceedings of the 2nd Workshop on Computing News Storylines. PID https://github.com/tommasoc80/EventStoryLine.

Goran Glavas and Jan Snajder and Marie-Francine Moens and Parisa KordJamshidi. 2014. *HiEve: A Corpus for Extracting Event Hierarchies from News Stories*. In Proceedings of LREC. PID http://takelab.fer.hr/hievents.rar.

Wang, Xiaozhi and Chen, Yulin and Ding, Ning and Peng, Hao and Wang, Zimu and Lin, Yankai and Han, Xu and Hou, Lei and Li, Juanzi and Liu, Zhiyuan and Li, Peng and Zhou, Jie. 2022. *MAVEN-ERE: A Unified Large-scale Dataset for Event Coreference, Temporal, Causal, and Subevent Relation Extraction*. In Proceedings of EMNLP. PID https://github.com/THU-KEG/MAVEN-ERE.