A Multimodal Text- and Graph-Based Approach for Open-Domain Event Extraction from Documents

Praval Sharma
College of Information Science & Technology
University of Nebraska Omaha, USA
pravalsharma@unomaha.edu

Abstract

Event extraction is essential for event understanding and analysis. It supports tasks such as document summarization and decision-making in emergency scenarios. However, existing event extraction approaches have limitations: (1) closed-domain algorithms are restricted to predefined event types and thus rarely generalize to unseen types and (2) open-domain algorithms, capable of handling unconstrained event types, have largely overlooked the potential of large language models (LLMs) despite their advanced abilities. Additionally, they do not explicitly model document-level contextual, structural, and semantic reasoning, which are crucial for effective event extraction but remain challenging for LLMs due to lost-in-the-middle phenomenon and attention dilution. To address these limitations, we propose multimodal open-domain event extraction, MODEE, a novel approach for open-domain event extraction that combines graph-based learning with text-based representation from LLMs to model document-level reasoning. Empirical evaluations on large datasets demonstrate that MODEE outperforms state-of-the-art open-domain event extraction approaches and can be generalized to closed-domain event extraction, where it outperforms existing algorithms.

1 Introduction

Event extraction involves identifying an event in text and extracting related arguments such as location, time, and agents involved. It is crucial for event analysis and understanding, and can be used in planning and responding to extreme phenomena such as disease outbreaks and natural disasters (Yu et al., 2020). Additionally, it is useful for tasks such as summarizing documents, clustering them, and building knowledge graphs.

There are several existing event extraction approaches. The closed-domain sentence-level event extraction approaches (Du and Cardie, 2020b; Hsu et al., 2022) use a sentence to extract events. Since events are generally described in multiple sentences (Tong et al., 2022), they are not able to extract complete information about events. The closed-domain document-level event extraction approaches (Du and Cardie, 2020a; Liu et al., 2023) address this limitation. However, they rely on predefined event schemas tailored to specific event types and thus have limited ability to generalize to unseen types. Open-domain event extraction approaches (Hamborg et al., 2019; Liu et al., 2019) can extract unrestricted event types and are critical for improving natural language understanding and downstream applications (Araki and Mitamura, 2018). However, existing approaches primarily rely on heuristics or rule-based methods. As a result, they struggle when events are described in ways not covered by their rules. Recent advances in natural language processing, particularly in large language models (LLMs), have led to significant improvements in various information extraction tasks. Despite this, these methods remain underexplored in document-level open-domain event extraction. This task requires understanding the document-level context, structure, and semantic relationships of event-related tokens. However, LLMs often struggle to capture these aspects due to the lost-in-the-middle phenomenon and attention dilution (Li et al., 2025). Additionally, recent efforts (Cao et al., 2024) use LLMs with simple fine-tuning, but do not explicitly model document-level reasoning or leverage multimodal learning to derive richer information for improved event extraction.

In this research, we propose multimodal open-domain event extraction (MODEE), a graph neural network (GNN)- and LLM-based approach designed for document-level event extraction in open-domain settings. MODEE combines token-level embeddings from the LLM with node-level embeddings from the GNN using an attention-based gated fusion mechanism to model document-level

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 2027–2041
November 6-10, 2026 ©2026 Association for Computational Linguistics

context, structure, and semantics of event-related tokens and extracts events in an end-to-end generative manner. This approach is motivated by recent work in retrieval augmented generation (Hu et al., 2025) and question answering (He et al., 2024), where graph-based representations have enhanced LLM reasoning and contextual understanding. Experimental results on a large, manually annotated, and statistically verified dataset show that MODEE outperforms fine-tuned LLMs, LLMs under zero- and five-shot prompting, and an existing open-domain event extraction approach. Additional experiments on a large closed-domain document-level event extraction dataset demonstrate the generality and adaptability of MODEE in closed-domain settings, although it is primarily designed for open-domain settings. The main contributions of this research are: (1) a novel graph neural network- and LLM-based approach, MODEE, for open-domain document-level event extraction and (2) an attention-based gated multimodal fusion mechanism that integrates token-level LLM embeddings with node-level graph embeddings for effective event extraction.

Note that, MODEE extracts the five key elements, i.e., 5Ws (where, when, what, who, and why), of events from documents. It is designed to extract the 5Ws because the 5Ws framework forms the foundation of event reporting in documents, particularly in news media (Harrower, 2010), applies to unconstrained event types, and is used by prior open-domain event extraction approaches (Hamborg et al., 2018; Liu et al., 2019). MODEE follows the *one-event-per-document* setting and extracts the main event as in Tong et al. (2022). While documents contain multiple events, they typically center on a newsworthy main event, with background events providing supporting information (Dijk, 1988). Therefore, extracting the main event is critical for news discourse comprehension and has been used as the core unit of analysis in news discourse studies (Choubey et al., 2020) and event-centric clustering (Zhang et al., 2025).

2 Related Work

Closed-Domain Event Extraction: These approaches rely on predefined event schemas to extract events from text. For example, they use the schema for a 'Conflict-Attack' event that defines arguments such as 'attacker,' 'target,' and 'instrument' to extract this type of event. The closed-domain sentence-level event extraction approaches rely on hand-crafted features (Li et al., 2013), convolutional neural networks (Chen et al., 2015), recurrent neural networks (Liu et al., 2017), graph-based techniques (Nguyen et al., 2022), question-answering (Du and Cardie, 2020b), and generative modeling (Cao et al., 2023). Because events are often described across multiple sentences (Tong et al., 2022), these sentence-level methods often fail to extract complete event information. To address this, document-level approaches relying on machine reading comprehension (Du and Cardie, 2020a), deep valued networks (Huang and Peng, 2021), chain reasoning (Liu et al., 2023), and graph-based methods (Huang et al., 2023; Wan et al., 2023) have been proposed. While these operate at the document level, they typically rely on predefined event schemas and thus struggle to generalize to events outside the schema set. Graph-based methods further rely on multi-step processes, where entities are first identified, followed by graph construction and event extraction. In contrast, our approach employs the 5Ws framework, which enables it to extract unrestricted event types, and performs end-to-end event extraction by integrating graph representations into a generative model. It constructs a document-level token graph without relying on external entity identification and jointly models graph and textual information for multimodal reasoning. This end-to-end generative design distinguishes it from prior graph-based methods.

Open-Domain Event Extraction: These approaches extract events without relying on a predefined set of event schemas. They are therefore able to extract unconstrained types of events. They employ various techniques, including heuristics and linguistic rules (Hamborg et al., 2019), distant supervision (Araki and Mitamura, 2018), clustering (Huang et al., 2016), Bayesian models (Yuan et al., 2018), neural latent variable models (Liu et al., 2019), adversarial domain adaptation (Naik and Rose, 2020), and bi-directional LSTM (Veyseh et al., 2021). They rely on handcrafted rules or limited supervision and thus face generalization challenges across diverse content and writing styles. Various information extraction tasks have benefitted from the use of LLMs. However, their application to open-domain event extraction remains limited. LLMs often struggle to capture document-level context, structure, and semantics of event-related tokens because of the lost-in-the-middle phenomenon and attention dilution, which are criti-

cal for event extraction (Li et al., 2025). However, recent studies (Cao et al., 2024) that apply LLMs use simple fine-tuning and do not explicitly model these document-level aspects. In contrast, our approach integrates graph-based learning with text-based representation from LLMs to model these aspects for improved event extraction.

Multimodal Event Extraction: These approaches leverage multiple modalities, such as images and text, to extract events. They employ techniques such as vision-language models (Li et al., 2022), attention-based fusion (Sun et al., 2024), and modality-shared encoder (Cao et al., 2025). While using multiple modalities, particularly images and text, has proven effective, their utilization requires parallel annotations across modalities. This is both expensive and complex (Cao et al., 2025), and therefore difficult to scale. As a result, most existing multimodal approaches are generally trained on image-text pairs without parallel annotations and confined to closed-domain event extraction. In contrast, in this study, we explore a more scalable and generalizable approach that derives complementary modalities from a single annotated source and uses them for open-domain event extraction. Specifically, we construct a graph using a document's text and use the resulting text-graph pairs for event extraction.

3 Methodology

3.1 Problem Definition

Given a document D, which describes an event e, the goal of open-domain document-level event extraction is to extract five key elements of e, i.e., $5Ws_e$ $(where_e, when_e, what_e, who_e, why_e)$, based on the information in D.

3.2 Multimodal Open-Domain Event Extraction

3.2.1 Overview of the Approach

As shown in Figure 1, multimodal open-domain event extraction (MODEE) consists of four modules: (1) Text encoder that produces contextual token-level text embeddings for a document, (2) Graph encoder that produces node-level graph embeddings for a document-level token graph created using a document, (3) Attention-based gated multimodal fusion module that integrates text and graph embeddings to produce integrated multimodal embeddings, and (4) Text decoder that generates 5Ws for the main event described in a document conditioned on the integrated multimodal embeddings. They are described in the following sections.

3.2.2 Text Encoder

The text encoder in MODEE encodes the token sequence from a given document to contextualized embeddings. Given a document D with n tokens $\{t_1, t_2, \ldots, t_n\}$, the encoder produces token-level hidden representations $\boldsymbol{H}_{text} \in \mathbb{R}^{n \times d}$, where d is the hidden dimension:

$$\boldsymbol{H}_{text} = TextEncoder(D). \quad (1)$$

In MODEE, we use the encoder part of T5 (Raffel et al., 2020) as the text encoder and the decoder part as the text decoder (see Section 3.2.5). T5's encoder-decoder architecture allows integration of representations from multiple modalities (e.g., text and graph) and generation of task-specific outputs (e.g., event 5Ws) conditioned on the integrated representations. This makes it particularly well-suited for integrating multimodal data and generative inference for event extraction, which is the focus of this study. While we use T5, other encoder-decoder LLMs such as Flan-T5 (Chung et al., 2024) can also be incorporated into MODEE.

3.2.3 Graph Encoder

The graph encoder in MODEE produces node-level embeddings for a document-level token graph created using a document. Given a document $D = \{t_1, t_2, \ldots, t_n\}$ with n tokens, it first creates a document-level token graph $G = (V, E)$, where V is a set of vertices representing the tokens in D, i.e., $V = \{t_1, t_2, \ldots, t_n\}$, and E is the set of undirected edges between every pair of vertices, i.e., $E = \{(t_i, t_j) \mid t_i, t_j \in V,\ i, j = 1, 2, \ldots, n,\ i \neq j\}$, forming a complete graph. Then, it produces embeddings, $\boldsymbol{H}_{graph} \in \mathbb{R}^{n \times d}$, for the graph using a two-layer GraphSAGE (Hamilton et al., 2017) with LSTM aggregation, where n is the number of nodes in the graph, which is also the total number of tokens in D, and d is the hidden dimension:

$$\boldsymbol{H}_{graph} = GraphEncoder(D). \quad (2)$$

The complete graph allows the graph encoder to model long-range dependencies between tokens across an entire document. By using GraphSAGE to encode the graph, the encoder then effectively captures the document-level structure and semantic relationships of event-related tokens in latent space. This helps MODEE determine the relevance of tokens in a document for event extraction.

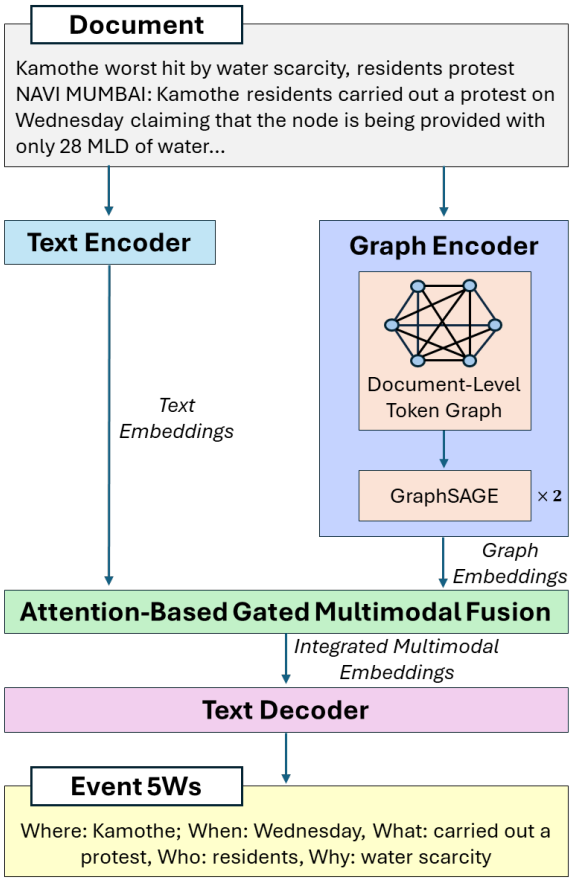


Figure 1: Overview of event extraction in MODEE.

3.2.4 Attention-Based Gated Multimodal Fusion

This module in MODEE integrates token-level contextual embeddings from the text encoder with document-level structure- and semantic-aware node-level representations from the graph encoder to produce rich integrated multimodal embeddings. The integration process involves two steps: Attention-based gating vector computation and Integrated multimodal embedding computation.

Attention-Based Gating Vector Computation: In this step, MODEE computes attention scores and generates a gating vector based on the text and graph embeddings to determine the relevance of individual tokens in a document with respect to the event described. This process is inspired by additive attention (Bahdanau et al., 2015), which facilitates richer interaction between representations from multiple modalities. Given $\boldsymbol{H}_{text} \in \mathbb{R}^{n \times d}$, the token embeddings from the text encoder (Equation 1), and $\boldsymbol{H}_{graph} \in \mathbb{R}^{n \times d}$, the node embeddings from the graph encoder (Equation 2), MODEE first projects these embeddings into a shared latent space using linear layers to enable direct cross-modal interaction:

$$\boldsymbol{H}_{textProj} = \boldsymbol{W}_{textProj} \boldsymbol{H}_{text} \tag{3}$$

$$\boldsymbol{H}_{graphProj} = \boldsymbol{W}_{graphProj} \boldsymbol{H}_{graph} \tag{4}$$

where $\boldsymbol{W}_{textProj}, \boldsymbol{W}_{graphProj} \in \mathbb{R}^{d \times d}$ are the weights of the linear layers. Although both text and graph embeddings have the same dimension in our setup, this projection supports modality fusion in settings where the embedding dimensions differ and enhances the generalizability of MODEE.

Next, it computes hidden representation $\boldsymbol{H}_{hidden}$ through an element-wise addition of $\boldsymbol{H}_{textProj}$ and $\boldsymbol{H}_{graphProj}$ followed by a non-linear activation:

$$\boldsymbol{H}_{hidden} = \tanh\left(\boldsymbol{H}_{textProj} + \boldsymbol{H}_{graphProj}\right). \tag{5}$$

This hidden representation captures both the contextual information from the text and the structural and semantic cues from the graph. Finally, to compute the token-wise gating vector $\alpha \in \mathbb{R}^{n \times 1}$, MODEE projects $\boldsymbol{H}_{hidden}$ to a vector of scalar scores using a linear layer, $\boldsymbol{V}_{attn} \in \mathbb{R}^{d \times 1}$ followed by a sigmoid activation:

$$\boldsymbol{\alpha} = \sigma\left(\boldsymbol{H}_{hidden} \cdot \boldsymbol{V}_{attn}\right) \tag{6}$$

where $\sigma(\cdot)$ denotes the sigmoid function. Since $\boldsymbol{H}_{hidden}$ is used to compute $\boldsymbol{\alpha}$, the cues from both text and graph embeddings are reflected in the gating scores and higher scores are assigned for tokens most relevant to the event described in D.

Integrated Multimodal Embedding Computation: In this step, MODEE produces integrated multimodal embeddings $\boldsymbol{H}_{integrated} \in \mathbb{R}^{n \times d}$ by applying the gating vector $\boldsymbol{\alpha}$ to the original text embeddings $\boldsymbol{H}_{text}$:

$$\boldsymbol{H}_{integrated} = \boldsymbol{H}_{text} \odot \boldsymbol{\alpha} \tag{7}$$

where $\odot$ denotes element-wise multiplication. This helps highlight the event-related tokens in a document. As a result, $\boldsymbol{H}_{integrated}$, when passed to the decoder, enables MODEE to selectively focus on tokens with high relevance to the event described in D and supports more accurate event extraction.

3.2.5 Text Decoder

The text decoder in MODEE generates 5Ws for the event described in a document. Given $\boldsymbol{H}_{integrated} \in \mathbb{R}^{n \times d}$, the integrated multimodal embeddings produced by the attention-based gated fusion module for a document D describing an event, the text

decoder autoregressively generates the 5Ws, i.e., where, when, what, who, and why:

$$\text{5Ws} = \mathit{TextDecoder}(\boldsymbol{H}_{\mathit{integrated}}) \quad (8)$$

In this work, we use the decoder part of the same T5 architecture used as the text encoder (see Section 3.2.2) to form a unified encoder-decoder framework. At inference time, the decoder generates 5Ws in a sequence-to-sequence manner beginning with a start token <pad> as initial decoder input and terminating when the end-of-sequence token </s> is generated. The decoder's output format is "where:<>; when:<>; what:<>; who:<>; why:<>," where each placeholder <> is replaced with the text generated for the corresponding 5W class or with 'none' if no text is generated, as shown in Figure 1.

3.3 Training Strategy

We train the four modules of MODEE jointly in an end-to-end fashion so that they support the unified objective of accurate event extraction. The text encoder, the attention-based gated multimodal fusion module, and the text decoder are optimized using cross-entropy loss:

$$\mathcal{L}_{\text{crossentropy}} = -\sum_{t=1}^{m} \log P\big(y_t \mid y_t^*, \boldsymbol{H}_{\mathit{integrated}}\big) \quad (9)$$

where m is the number of ground-truth tokens, y_t is the ground-truth token at position t, y_t^* denotes the sequence of all previous tokens, i.e., $(y_{t-1}, y_{t-2}, \ldots, y_2, y_1)$, and $P(\cdot)$ is the probability assigned by the model to y_t conditioned on the integrated multimodal embeddings $\boldsymbol{H}_{\mathit{integrated}}$. This training design allows the encoder to learn informative contextual representations, the fusion module to generate rich integrated multimodal embeddings conditioned on both text- and graph-based cues, and the decoder to generate accurate event 5Ws.

The graph encoder is optimized using contrastive loss (Chen et al., 2020). Given $\boldsymbol{H}_{\mathit{graph}} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n\}$, the embeddings for n nodes $\{t_1, t_2, \ldots, t_n\}$ from a document-level token graph G, and $y = \{y_1, y_2, \ldots, y_n\}$, their corresponding 5W classes, the contrastive loss for each node pair (t_i, t_j) such that $y_i = y_j$, $i \neq j$, and $1 \leq i \leq j \leq n$ is computed as follows:

$$\mathcal{L}_{\text{contrastive}} = -\log \frac{\exp\big(\mathit{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau\big)}{\sum_{l=1}^{n} \exp\big(\mathit{sim}(\boldsymbol{z}_i, \boldsymbol{z}_l)/\tau\big)} \quad (10)$$

where $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are the embeddings for nodes t_i and t_j, respectively, $\boldsymbol{z}_l$ are the embeddings for nodes such that $y_l \neq y_i$, $\mathit{sim}(\cdot)$ denotes cosine similarity, and τ is a temperature hyperparameter to adjust the loss function's sensitivity. This training strategy allows the graph encoder to learn to draw together tokens representing the same 5W class (e.g., all "where" tokens) and push apart those from different classes in the embedding space. As a result, the encoder learns to automatically capture the document-level structure and semantics of event-related tokens in latent space and provide valuable cues to the attention-based gated multimodal fusion module for computing the gating vector and determining the relevance of each token in a document with respect to the event described. Due to memory constraints, we compute contrastive learning on a sampled subset of nodes during training. Specifically, we randomly sample five nodes for each of the 5W classes and an additional set of five nodes that do not belong to any 5W class. This sampling strategy balances efficiency with class diversity and enables the graph encoder to learn discriminative representations. Note that we initialize each node in the document-level token graph using the encoder of the pretrained T5 model during training.

4 Experiments

4.1 Dataset and Evaluation

Existing publicly available document-level open-domain event extraction datasets are limited to a relatively small number of manually annotated and statistically verified documents (e.g., 100 in (Araki and Mitamura, 2018) and 680 in (Liu et al., 2019)). Such relatively small-scale datasets can restrict the development and evaluation of large-scale LLM-based approaches. Therefore, we developed a large-scale, manually annotated, and statistically verified dataset to train and evaluate MODEE. The dataset consists of 10,000 news reports published between 2015 to 2019 from seven Indian newspapers: Times of India, The Hindu, The Pioneer, Economic Times, Assam Tribune, Kashmir Observer, and Incredible Orissa. These newspapers vary in journalistic style and geographic focus, and thus contribute to the diversity of the dataset.

According to the inverse pyramid concept, key information typically appears early in a report (Harrower, 2010). Additionally, Ebner et al. (2020) observed that majority of the arguments for an event appeared within a five-sentence window in reports.

Guided by these observations and because we focus on extracting the main event, we included the title and first five sentences of each report in our dataset. This allowed us to balance annotation cost while including the portions of reports where main events are most frequently described.

The dataset was created in three stages: training, annotation, and resolution. In the training stage, three coders (university students familiar with Indian context) underwent three rounds of training to familiarize them with the task. In each round, they annotated seven reports (one from each of the seven newspapers) not included in the final dataset. After each round, they participated in discussions to resolve differences in their annotations and improve consistency. By the final round, they achieved an inter-coder reliability above 0.8, measured using Krippendorff's alpha (Krippendorff and Craggs, 2016), which indicates high consistency.

In the annotation stage, the coders independently annotated all 10,000 reports in the dataset. In the resolution stage, they resolved annotation differences based on a defined policy. If all coders agreed on an annotation, it was considered the gold standard. If two coders agreed on an annotation, the third was asked to reannotate. If unanimous agreement was subsequently reached, the annotation was considered the gold standard. If disagreement persisted, an expert (a researcher familiar with event extraction) determined the gold standard. If all coders disagreed on an annotation, all reannotated, and the process was repeated until at least two coders reached an agreement.

Once the dataset development was completed, we randomly split it into training, validation, and test sets containing 8,000, 1,000, and 1,000 reports, respectively (see Appendix A for dataset details). The training and validation sets were used to train MODEE and the test set was used for evaluation. In this research, we report the evaluation results using precision (P), recall (R), and $F1$ score (see Section 4.4) computed using exact match (EM) following prior work on event extraction (Liu et al., 2019; Tong et al., 2022). We also report results using ROUGE-L (Lin, 2004) and BERTScore (Zhang et al., 2020) to capture lexical and semantic similarity between predictions and gold standard, with precision, recall, and F1 score computed per document and averaged over the test set.

4.2 Training Configuration

The text encoder and decoder modules are trained with AdamW optimizer using learning rate of 1e-3 (T5-Small) and 1e-4 (T5-Base). Input and output sequence lengths are capped at 512 tokens and beam size is set to 5 during inference. The graph encoder and attention-based gated fusion modules are trained using Adam optimizer with 1e-3 learning rate and 5e-4 weight decay. MODEE is trained for 10 epochs on NVIDIA V100 GPUs, using gradient accumulation to achieve an effective batch size of 8 due to memory constraints.

4.3 Baselines

We evaluate the efficacy of MODEE by comparing it against several baselines. Although MODEE uses multiple modalities (i.e., graph and text), both are derived directly from textual data unlike existing multimodal approaches that rely on multiple sources (e.g., image-text pairs). To ensure a fair comparison, we therefore evaluate MODEE against baselines that operate solely on document-level text. As existing LLM-based open-domain event extraction approaches are not publicly available, we fine-tune standard T5 models (Raffel et al., 2020), T5-Small, T5-Base, and T5-Large, on our task using the configuration described in Section 4.2 (with T5-Large trained using the same configuration as T5-Base). We also compare MODEE against modern strong pretrained LLMs such as Llama 3.1 (8B and 70B), Qwen 3 (8B and 32B), and Mistral V0.3 (7B) under zero- and five-shot prompting[1] (see Appendix B for prompts used). For a fair comparison, we further fine-tune smaller LLMs, Llama 3.2 (1B) and Qwen 3 (0.6B), which are comparable in size to our approach (see Appendix C for parameter size of MODEE), using the same configuration as T5-Base (Section 4.2). In addition, we use Giveme5W1H (Hamborg et al., 2019), a heuristic-based open-source open-domain event extraction algorithm, as a non-generative baseline.

4.4 Main Result

Table 1 presents the performance of MODEE and all baselines on the test dataset. MODEE-Base, which uses T5-Base, achieves the best overall performance across EM, ROUGE-L, and BERTScore. Despite being based on the same architecture, its improved performance over T5-Base demonstrates that incorporating multiple modalities derived from

[1]We use 8-bit quantization to reduce GPU memory usage.

single source, particularly creating graphs from text and using the text-graph pairs, enhances open-domain event extraction. The better performance of MODEE-Small compared to T5-Small further bolsters this. T5-Large, despite being a larger model, does not outperform MODEE-Base. This shows that using a larger model does not compensate for the lack of document-level context, structure, and semantics of event-related tokens that MODEE captures through multimodal integration. Giveme5W1H is the least effective model and illustrates the limitations of rule-based approaches.

Modern LLMs do not perform well under both zero- and five-shot prompting. This indicates that prompting alone is insufficient for accurately extracting events. Increasing model size does not lead to improved performance. Additionally, fine-tuned LLMs, Llama 1B and Qwen 0.6B, underperform compared to MODEE, despite being trained under the same configuration and having more parameters. This suggests that effective event extraction requires capturing document-level context, structure, and semantics of event-related tokens, which MODEE does by combining text- and graph-based representations. Its higher scores on ROUGE-L and BERTScore further indicate that it produces more semantically coherent 5Ws than the LLMs.

To provide a more fine-grained analysis, we examine model performance across individual 5W classes (see Appendix D). While some models perform better on specific Ws (e.g., T5-Large on When), MODEE consistently achieves stronger performance on Why, which exhibits higher lexical variability, greater sparsity, and longer spans (see Appendix A), and therefore requires deeper understanding of document-level context, structure, and semantics. MODEE leverages contrastive learning in its graph encoder to pull tokens with similar semantic roles closer in the embedding space, and integrates structural graph representations from the document-level token graph with contextual textual representations to capture document-level context, structure, and semantics of event-related tokens. This allows MODEE to outperform all baselines even on the more challenging Why.

MODEE gains from scaling the underlying language model. Transitioning from T5-Small to T5-Base improved its performance. This shows the versatility and scalability of MODEE in effectively leveraging more powerful language models.

4.5 Ablation Study

To investigate the contributions of different modules in MODEE, we created ablated versions of MODEE-Base, our best performing model. First, to evaluate the role of contrastive learning in enhancing the graph encoder and the downstream event extraction, we train MODEE **without contrastive learning**. In this setting, the graph encoder is trained jointly with all other modules using the same cross-entropy loss (see Equation 9). Next, to assess the importance of attention-based gated multimodal fusion, we replace the fusion module with simple **element-wise addition** of the text and graph embeddings produced by the respective encoders. Finally, to examine the effect of document-level structural information derived from the complete graph, we modify the graph encoder to operate on a **linear graph**, where each token is connected only to its immediate neighbors in the document forming a linear chain. In this setting, the graph encoder can only capture local token context (i.e., context within a small window), which limits its ability to capture document-level event structure.

Table 2 presents the results of the ablation study. Removing contrastive learning leads to performance drops across all 5W classes compared to the full MODEE. This demonstrates that contrastive learning helps the graph encoder produce more meaningful embeddings by pulling together semantically similar tokens in the embedding space. As a result, MODEE can more effectively focus on relevant event-related tokens during integration, thereby leading to a more accurate event extraction. Replacing the attention-based gated multimodal fusion module with element-wise addition results in a significant decline in performance. This highlights the importance of attention-based gating mechanism in integrating text and graph representations. Additionally, using a linear graph instead of the document-level complete graph leads to a substantial performance drop. This illustrates the importance of the document-level structural context for accurate event extraction. Overall, the full MODEE outperforms all ablated models, which validates the efficacy of its different modules.

5 Generality of MODEE in Closed-Domain Event Extraction

To assess the generality of MODEE, we trained and evaluated our best-performing model, MODEE-Base, on DocEE (Tong et al., 2022). We used

Models	Exact Match			ROUGE-L			BERTScore		
	P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
T5-Small (Fine-tuned)	52.2	47.5	49.8	69.8	69.7	67.8	93.5	93.6	93.5
T5-Base (Fine-tuned)	54.9	51.8	53.3	71.7	72.8	70.5	94.1	94.2	94.1
T5-Large (Fine-tuned)	57.2	54.4	55.8	**75.2**	75.1	73.0	94.5	94.5	94.5
Giveme5W1H	15.6	16.6	16.1	31.1	31.4	28.8	86.6	86.3	86.3
Llama 8B (0-shot)	9.8	10.6	10.2	27.7	52.7	30.8	84.1	87.9	85.8
Llama 8B (5-shot)	9.6	10.3	10.0	27.1	51.8	30.1	83.9	87.8	85.7
Llama 70B (0-shot)	14.1	13.4	13.8	31.2	45.2	31.4	85.2	87.5	86.2
Llama 70B (5-shot)	14.4	13.7	14.0	31.3	45.2	31.5	85.3	87.6	86.3
Qwen 8B (0-shot)	15.8	19.2	17.3	37.1	55.9	40.5	86.7	89.4	88.0
Qwen 8B (5-shot)	15.8	19.3	17.4	37.2	56.4	40.5	86.7	89.4	88.0
Qwen 32B (0-shot)	9.9	11.6	10.7	30.3	68.0	37.2	84.6	89.9	87.1
Qwen 32B (5-shot)	9.0	10.5	9.7	29.4	67.7	36.5	84.5	89.8	87.0
Mistral 7B (0-shot)	15.5	18.8	17.0	36.9	51.4	38.6	86.8	89.2	87.9
Mistral 7B (5-shot)	15.4	18.6	16.8	36.6	51.2	38.5	86.8	89.2	87.9
Llama 1B (Fine-tuned)	58.2	55.3	56.8	73.6	72.5	71.3	94.4	94.4	94.4
Qwen 0.6B (Fine-tuned)	56.0	**57.7**	56.9	72.8	73.2	71.1	94.3	94.4	94.3
MODEE-Small	54.7	51.7	53.2	71.2	72.8	70.1	93.9	94.2	94.0
MODEE-Base	**58.7**	56.7	**57.7**	75.1	**75.9**	**73.7**	**94.7**	**94.8**	**94.7**

Table 1: Performance comparison on event extraction (%) using Exact Match, ROUGE-L, and BERTScore. Bold indicates the best performance.

Method	Where			When			What			Who			Why		
	P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
MODEE	**67.0**	**68.0**	**67.5**	**85.7**	**79.4**	**82.4**	**35.7**	**35.7**	**35.7**	**59.1**	**56.9**	**58.0**	**35.7**	**31.5**	**33.5**
Without Contrastive Learning	62.0	63.7	62.8	85.0	79.2	82.0	34.2	34.2	34.2	53.9	52.5	53.2	31.2	26.4	28.6
Element-Wise Addition	7.1	3.5	4.7	13.8	14.5	14.1	10.4	10.4	10.4	5.6	5.6	5.6	2.3	0.2	0.4
Linear Graph	44.2	45.3	44.7	71.5	71.1	71.3	16.3	16.3	16.3	33.7	30.8	32.2	7.8	4.6	5.8

Table 2: Ablation results of MODEE. Bold indicates the best performance.

Method	EM			HM		
	P	*R*	*F1*	*P*	*R*	*F1*
BERT_Seq	35.3	35.9	35.6	20.3	25.0	22.4
MG-Reader	30.3	35.9	32.9	45.6	50.8	48.1
Doc2EDAG	37.1	**36.1**	36.6	54.2	53.7	53.9
BERT_QA	41.9	28.1	33.5	75.8	50.6	60.7
Ontology_QA	51.3	34.2	41.0	80.3	53.6	64.3
MODEE-Base	**56.4**	35.0	**43.1**	**88.9**	**55.1**	**68.1**

Table 3: Performance comparison on document-level closed-domain event extraction (%). Bold indicates the best performance.

DocEE because it is one of the largest manually annotated closed-domain event extraction datasets and follows the one-event-per-document paradigm. We trained MODEE-Base using inputs that consist of a document followed by the argument types associated with the main event it describes so that the model learns to generate the corresponding event arguments. We compared the performance of MODEE-Base against the state-of-the-art (SOTA) algorithms: BERT_Seq (Du and Cardie, 2020a), MG-Reader (Du and Cardie, 2020a), Doc2EDAG (Zheng et al., 2019), BERT_QA (Du and Cardie, 2020b), and Ontology_QA (Tong et al., 2022). For a fair comparison, we adopted the same train/test split (22k train, 2.7k test) under normal-setting used in Tong et al. (2022). We trained MODEE-Base for 20 epochs using the configuration described in Section 4.2, except the input and output sequence lengths were limited to 1,024 tokens.

Table 3 presents the experimental results using precision (P), recall (R), and $F1$ scores computed with exact match (EM) and head noun phrase match (HM), following prior work (Tong et al., 2022). For the SOTA algorithms, we use the reported performance from Tong et al. (2022). MODEE-Base has better performance than all SOTA algorithms for both EM and HM, despite being primarily designed for open-domain event extraction. One reason for lower accuracy of the existing SOTA algorithms is their limited ability to understand the document-level semantics of event related tokens within overall document context. In contrast, MODEE-Base, through its integration of graph- and text-based representations, more effectively utilizes the document-level context and semantic relationships of the event related tokens in a document. As a result, it achieves improved performance, and this experimental result demonstrates the generality and adaptability of MODEE.

6 Conclusion

In this research, we introduced MODEE, a novel GNN- and LLM-based open-domain event extraction approach that combines multiple modalities to model document-level reasoning, which is essential for event extraction but remains challenging for LLMs due to lost-in-the-middle phenomenon and attention dilution. Experimental results showed that MODEE outperformed strong baselines in both open-domain and closed-domain settings. Future work will extend MODEE to multi-document settings and incorporate additional modalities such as images, audio, and video.

Limitations

We evaluate MODEE's event extraction capability using news reports. While this domain is well suited for event 5Ws extraction, additional experiments on other domains such as social media, scientific articles, or financial documents are necessary to fully assess MODEE across diverse text genres. MODEE assumes a one-event-per-document setting. When a document reports multiple salient events, the model is not explicitly designed to extract all events and may focus on a single dominant event. In real-world scenarios, documents often report multiple events. Therefore, extending MODEE to multi-event extraction will improve its applicability in real-world settings. In many documents, main events are related to background events through temporal, causal, or contextual dependencies, with some background events serving as precursors or consequences of the main events. For such related events, these relationships could be modeled using a hierarchical graph that captures multiple interconnected events within a document. Given datasets with annotations for multiple events, MODEE's graph module can be extended to reason over such hierarchical graphs and extract multiple events instead of a single main event. In cases where events reported in documents are independent and have no explicit relationships among them, MODEE could be extended to generate 5W arguments for multiple events rather than a single main event. An additional event aggregation module could then be incorporated into MODEE to group and organize the generated arguments and produce multiple distinct events.

The document-level token graph is a critical component of MODEE as it enables the model to capture the global structure and semantic relationships among event-related tokens. However, constructing and processing a document-level token graph for a very long document introduces additional computational overhead, which may limit MODEE's scalability without further optimization. Techniques such as graph sparsification or edge pruning could help reduce this complexity. Additionally, an empirical analysis of the computational efficiency of MODEE, including memory usage and inference latency compared with baseline T5 models, would help characterize the computational trade-offs introduced by the document-level token graph and guide the development of more effective optimization techniques. Finally, MODEE uses T5 as its backbone due to its encoder-decoder architecture. Most modern LLMs are decoder-only models, and we have not evaluated MODEE with larger and more recent LLM backbones, which could potentially allow better utilization of the capabilities of modern LLMs.

Ethical Considerations

Creating a large, manually annotated dataset requires substantial time and effort. We employed student annotators and compensated them at the basic minimum pay rate set by the university for student workers. Participation in the annotation process was voluntary, and to protect privacy and confidentiality, no personal or identifying information about the annotators is disclosed.

References

Jun Araki and Teruko Mitamura. 2018. Open-Domain Event Detection using Distant Supervision. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 878–891, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015*.

Jianwei Cao, Yanli Hu, Zhen Tan, and Xiang Zhao. 2025. Cross-modal Multi-task Learning for Multimedia Event Extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 11454–11462.

Pengfei Cao, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. 2023. Zero-Shot Cross-Lingual Event Argument Extraction with Language-Oriented Prefix-Tuning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):12589–12597.

Yang Cao, Yangsong Lan, Feiyan Zhai, and Piji Li. 2024. 5W1H Extraction With Large Language Models. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, 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.

Prafulla Kumar Choubey, Aaron Lee, Ruihong Huang, and Lu Wang. 2020. Discourse as a Function of Event: Profiling Discourse Structure in News Articles around the Main Event. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5374–5386, Online. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. Scaling Instruction-Finetuned Language Models. *Journal of Machine Learning Research*, 25(70):1–53.

Teun A. van Dijk. 1988. *News As Discourse*. Routledge, New York.

Xinya Du and Claire Cardie. 2020a. Document-Level Event Role Filler Extraction using Multi-Granularity Contextualized Encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8010–8020, Online. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020b. Event Extraction by Answering (Almost) Natural Questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

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

Felix Hamborg, Corinna Breitinger, and Bela Gipp. 2019. Giveme5W1H: A Universal System for Extracting Main Events from News Articles. In *Proceedings of the 7th International Workshop on News Recommendation and Analytics*, Copenhagen, Denmark.

Felix Hamborg, Soeren Lachnit, Moritz Schubotz, Thomas Hepp, and Bela Gipp. 2018. Giveme5W: Main Event Retrieval from News Articles by Extraction of the Five Journalistic W Questions. In *Transforming Digital Worlds*, Lecture Notes in Computer Science, pages 356–366, Cham. Springer International Publishing.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Tim Harrower. 2010. *Inside Reporting: A Practical Guide to the Craft of Journalism*. McGraw-Hill Companies,Incorporated.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. In *Advances in Neural Information Processing Systems*, volume 37, pages 132876–132907.

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.

Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2025. GRAG: Graph Retrieval-Augmented Generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4145–4157, Albuquerque, New Mexico. Association for Computational Linguistics.

Guanhua Huang, Runxin Xu, Ying Zeng, Jiaze Chen, Zhouwang Yang, and Weinan E. 2023. An Iteratively Parallel Generation Method with the Pre-Filling Strategy for Document-level Event Extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10834–10852, Singapore. Association for Computational Linguistics.

Kung-Hsiang Huang and Nanyun Peng. 2021. Document-level Event Extraction with Efficient End-to-end Learning of Cross-event Dependencies. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 36–47, Virtual. Association for Computational Linguistics.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. Voss, Jiawei Han, and Avirup Sil. 2016. Liberal Event Extraction and Event Schema Induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume*

1: Long Papers), pages 258–268, Berlin, Germany. Association for Computational Linguistics.

Klaus Krippendorff and Richard Craggs. 2016. The Reliability of Multi-Valued Coding of Data. *Communication Methods and Measures*, 10(4):181–198.

Bobo Li, Xudong Han, Jiang Liu, Yuzhe Ding, Liqiang Jing, Zhaoqi Zhang, Jinheng Li, Xinya Du, Fei Li, Meishan Zhang, Min Zhang, Aixin Sun, Philip S. Yu, and Hao Fei. 2025. Event Extraction in Large Language Model. *arXiv preprint*. ArXiv:2512.19537 [cs].

Manling Li, Ruochen Xu, Shuohang Wang, Luowei Zhou, Xudong Lin, Chenguang Zhu, Michael Zeng, Heng Ji, and Shih-Fu Chang. 2022. CLIP-Event: Connecting Text and Images with Event Structures. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16399–16408.

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.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jian Liu, Chen Liang, Jinan Xu, Haoyan Liu, and Zhe Zhao. 2023. Document-Level Event Argument Extraction With a Chain Reasoning Paradigm. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9570–9583, Toronto, Canada. Association for Computational Linguistics.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting Argument Information to Improve Event Detection via Supervised Attention Mechanisms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Vancouver, Canada. Association for Computational Linguistics.

Xiao Liu, Heyan Huang, and Yue Zhang. 2019. Open Domain Event Extraction Using Neural Latent Variable Models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2860–2871, Florence, Italy. Association for Computational Linguistics.

Aakanksha Naik and Carolyn Rose. 2020. Towards Open Domain Event Trigger Identification using Adversarial Domain Adaptation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7618–7624, Online. Association for Computational Linguistics.

Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Nguyen. 2022. Joint Extraction of Entities, Relations, and Events via Modeling Inter-Instance and Inter-Label Dependencies. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4363–4374, Seattle, United States. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):140:5485–140:5551.

Lin Sun, Kai Zhang, Qingyuan Li, and Renze Lou. 2024. UMIE: Unified Multimodal Information Extraction with Instruction Tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19062–19070.

MeiHan Tong, Bin Xu, Shuai Wang, Meihuan Han, Yixin Cao, Jiangqi Zhu, Siyu Chen, Lei Hou, and Juanzi Li. 2022. DocEE: A Large-Scale and Fine-grained Benchmark for Document-level Event Extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3970–3982, Seattle, United States. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Minh Van Nguyen, Bonan Min, and Thien Huu Nguyen. 2021. Augmenting Open-Domain Event Detection with Synthetic Data from GPT-2. In *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 644–660, Cham. Springer International Publishing.

Qizhi Wan, Changxuan Wan, Keli Xiao, Dexi Liu, Chenliang Li, Bolong Zheng, Xiping Liu, and Rong Hu. 2023. Joint Document-Level Event Extraction via Token-Token Bidirectional Event Completed Graph. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10481–10492, Toronto, Canada. Association for Computational Linguistics.

Manzhu Yu, Myra Bambacus, Guido Cervone, Keith Clarke, Daniel Duffy, Qunying Huang, Jing Li, Wenwen Li, Zhenlong Li, Qian Liu, Bernd Resch, Jingchao Yang, and Chaowei Yang. 2020. Spatiotemporal event detection: a review. *International Journal of Digital Earth*, 13(12):1339–1365.

Quan Yuan, Xiang Ren, Wenqi He, Chao Zhang, Xinhe Geng, Lifu Huang, Heng Ji, Chin-Yew Lin, and Jiawei Han. 2018. Open-Schema Event Profiling for Massive News Corpora. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 587–596, New York, NY, USA. Association for Computing Machinery.

Longyin Zhang, Bowei Zou, and AiTi Aw. 2025. Enhancing Event-centric News Cluster Summarization

via Data Sharpening and Localization Insights. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16412–16426, Vienna, Austria. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. *arXiv preprint*. ArXiv:1904.09675 [cs].

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2EDAG: An End-to-End Document-level Framework for Chinese Financial 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 337–346, Hong Kong, China. Association for Computational Linguistics.

A Dataset Details

Figure 2 shows an example of the main event 5Ws annotated in a document from our dataset. The dataset has an average document length of 127 words, with each document containing 4.1 of the five Ws on average. What is present in all documents, while Where, When, Who, and Why appear in 9,390 (93.9%), 9,493 (94.9%), 7,391 (73.9%), and 4,451 (44.5%) documents, respectively. This distribution highlights the relative sparsity of causal information (Why) in news reports, which makes identifying and extracting it complex. In terms of lexical diversity, the dataset contains 5,469 (58.2%) unique Where values, 1,279 (13.5%) When, 7,462 (74.6%) What, 4,424 (59.9%) Who, and 3,944 (88.6%) Why. This indicates substantial variability, particularly for What and Why, making generalization more challenging. The average span lengths (in words) are 1.6 (Where), 1.0 (When), 5.8 (What), 2.3 (Who), and 7.4 (Why) and suggest that What and Why are typically longer, more descriptive, and semantically complex. Overall, the dataset exhibits high lexical diversity, substantial variation across values for the 5Ws, multi-word spans, and sparse causal information, which collectively make the task of event 5Ws extraction challenging.

B Prompt Examples

We adopt Alpaca-style prompting for both zero- and five-shot settings, and use the same format for fine-tuning the LLMs in this study (see Figures 3 and 4). Our focus was on comparing the performance of different models under a consistent setup. Although more sophisticated prompt engineering may further improve performance, such optimization is outside the scope of this work.

C Parameter Size of MODEE

The parameter sizes of the LLM backbones in MODEE are well established: 60 million for T5-Small and 220 million for T5-Base. The graph encoder and fusion module contain 5 million and 525K parameters, respectively, in MODEE-Small, resulting in a total of 66 million parameters. For MODEE-Base, the graph encoder and fusion module contain 12 million and 1 million parameters, respectively, resulting in a total of 233 million parameters.

D Main Results (Full Tables)

Tables 4, 5, and 6 report precision (P), recall (R), and F1 scores for each 5W class using Exact Match, ROUGE-L, and BERTScore. Models perform better on Where, When, and Who. Performance on What and especially Why is lower, which reflects their greater difficulty. Why is particularly challenging due to higher lexical variability and sparsity. Notably, MODEE achieves the strongest performance on Why, outperforming all baselines, including fine-tuned LLMs, despite its smaller size. This improvement likely stems from its integration of textual and graph-based representations for modeling document-level reasoning.

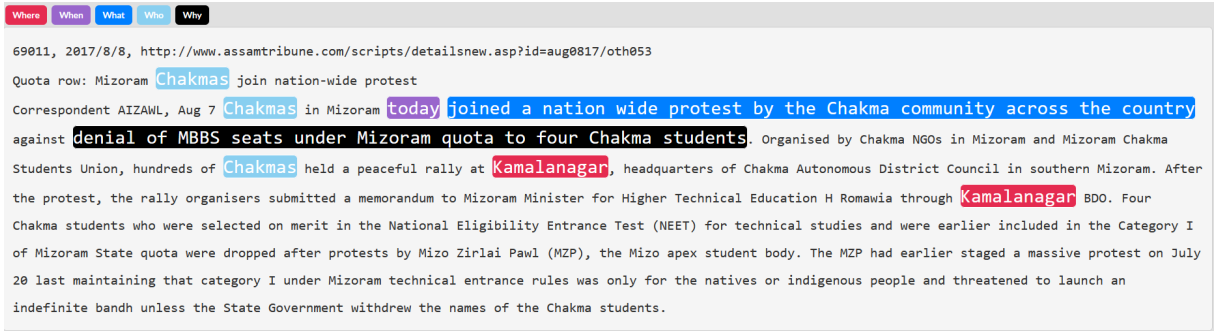


Figure 2: Example of a document from our dataset with the 5Ws for the main event annotated (highlighted).

Below is an instruction that describes a task, paired with an input that provides further context. Write an output in the format: Where: "..."; When: "..."; What: "..."; Who: "..."; Why: "..." that appropriately completes the request.

Instruction:
Extract the 5Ws for the main event described in the text:

Input:
Man thrown off bullock cart; dies P. Suresh Kumar (33) of Ondipudur was killed when he was thrown off the bullock cart he was riding at Chinthamani Pudur, near Sulur, on Wednesday night. The accident occurred when the bulls got frightened and the cart fell on its side. His cousin Saravana Kumar (32), who was with him on the cart, escaped with injuries. House burgled Unidentified persons gained entry into R. Balasundaram house in TNHB Colony in Ganapathy Managar sometime between August 10 and 11 and decamped with jewellery weighing about 160 gm. The burglars broke open the front door to enter the house and access the safe.

Output:

Figure 3: Zero-shot prompt example.

Models	Where			When			What			Who			Why			Overall		
	P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
T5-Small (Fine-tuned)	57.1	57.9	57.5	85.3	76.7	80.7	29.2	29.2	29.2	45.6	40.0	42.6	30.5	17.4	22.2	52.2	47.5	49.8
T5-Base (Fine-tuned)	60.8	62.1	61.4	85.7	78.7	82.0	30.8	30.8	30.8	54.0	49.5	51.7	31.7	24.0	27.3	54.9	51.8	53.3
T5-Large (Fine-tuned)	65.3	66.8	66.0	87.6	79.2	**83.2**	33.7	33.6	33.6	56.2	53.7	54.9	30.1	24.0	26.7	57.2	54.4	55.8
Giveme5W1H	21.9	16.5	18.8	60.7	47.1	53.0	1.5	1.5	1.5	6.4	8.3	7.2	0.1	0.2	0.1	15.6	16.6	16.1
Llama 8B (0-shot)	11.0	11.6	11.3	21.0	19.3	20.1	3.7	3.2	3.4	9.9	11.2	10.5	3.0	5.5	3.9	9.8	10.6	10.2
Llama 8B (5-shot)	11.1	11.7	11.4	20.0	18.1	19.0	4.2	3.6	3.9	10.2	11.5	10.8	2.1	3.7	2.7	9.6	10.3	10.0
Llama 70B (0-shot)	15.4	15.8	15.6	30.2	24.4	27.0	6.0	4.4	5.1	14.7	14.2	14.4	2.8	4.4	3.4	14.1	13.4	13.8
Llama 70B (5-shot)	16.0	16.3	16.1	31.1	24.8	27.6	6.7	4.9	5.7	13.9	13.4	13.6	3.0	4.6	3.6	14.4	13.7	14.0
Qwen 8B (0-shot)	26.0	27.4	26.7	38.0	39.7	38.8	5.5	5.5	5.5	8.5	11.4	9.7	0.9	2.0	1.2	15.8	19.2	17.3
Qwen 8B (5-shot)	26.8	28.3	27.5	37.6	39.5	38.6	5.3	5.3	5.3	8.3	11.1	9.5	1.0	2.2	1.4	15.8	19.3	17.4
Qwen 32B (0-shot)	15.3	15.5	15.4	30.7	30.9	30.8	0.8	0.8	0.8	2.4	3.1	2.7	0.4	0.9	0.6	9.9	11.6	10.7
Qwen 32B (5-shot)	13.2	13.3	13.2	26.9	27.3	27.1	1.6	1.5	1.5	2.4	3.1	2.7	0.6	1.3	0.9	9.0	10.5	9.7
Mistral 7B (0-shot)	18.8	19.8	19.3	34.0	35.5	34.7	6.9	6.9	6.9	15.0	19.9	17.1	2.8	6.2	3.9	15.5	18.8	17.0
Mistral 7B (5-shot)	18.1	19.0	18.6	34.0	35.4	34.7	7.0	7.0	7.0	14.9	19.8	17.0	2.8	6.2	3.9	15.4	18.6	16.8
Llama 1B (Fine-tuned)	64.4	64.2	64.3	78.8	78.2	78.5	38.4	38.4	38.4	61.5	56.2	58.7	34.9	24.9	29.0	58.2	55.3	56.8
Qwen 0.6B (Fine-tuned)	64.5	64.1	64.3	77.6	77.8	77.7	39.2	39.2	**39.2**	60.6	64.5	**62.5**	27.2	31.9	29.4	56.0	57.7	56.9
MODEE-Small	57.7	61.0	59.3	86.0	77.6	81.6	35.0	35.0	35.0	52.4	47.8	50.0	29.0	21.4	24.6	54.7	51.7	53.2
MODEE-Base	67.0	68.0	**67.5**	85.7	79.4	82.4	35.7	35.7	35.7	59.1	56.9	58.0	35.7	31.5	**33.5**	58.7	56.7	**57.7**

Table 4: Performance comparison on event extraction (%) using Exact Match. Bold indicates the best performance.

Below is an instruction that describes a task, paired with an input that provides further context. Write an output in the format: Where: "..."; When: "..."; What: "..."; Who: "..."; Why: "..." that appropriately completes the request.

Instruction:
Extract the 5Ws for the main event described in the text:

Input:
PAUTA to hold protest at kisan mela Ludhiana: Members of the PAU Teachers Association (PA UTA) will wear black badges as a ...
Output: Where: "Ludhiana"; When: "Friday"; What: "protest"; Who: "University Fourth Class Workers Union, University Employees Union, PAU Teachers Association (PAUTA)"; Why: "the alleged autocratic attitude of vice chancellor Dr Baldev Singh Dhillon"

Input:
[document text]
Output: Where: "..."; When: "..."; What: "..."; Who: "..."; Why: "..."

...

Now extract the 5Ws for the following document.
Input:
[test document]
Output:

Figure 4: Five-shot prompt example.

Models	Where			When			What			Who			Why			Overall		
	P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
T5-Small (Fine-tuned)	65.5	66.9	65.3	87.1	87.0	87.0	60.8	60.4	56.5	68.0	64.7	64.5	67.9	71.4	66.1	69.8	69.7	67.8
T5-Base (Fine-tuned)	67.8	69.2	67.8	87.9	87.8	87.8	61.4	63.6	58.5	73.0	72.2	71.0	69.3	73.3	67.7	71.7	72.8	70.5
T5-Large (Fine-tuned)	73.5	75.0	73.4	89.5	89.4	**89.4**	65.9	64.5	60.7	76.2	76.1	74.4	67.7	67.2	63.3	75.2	75.1	73.0
Giveme5W1H	31.9	28.6	29.1	64.4	66.6	64.9	21.4	21.9	18.2	22.4	28.1	21.5	11.7	5.6	5.3	31.1	31.4	28.8
Llama 8B (0-shot)	25.2	58.7	30.0	32.6	54.4	36.9	25.6	45.3	25.3	27.8	53.8	31.6	27.3	48.9	30.5	27.7	52.7	30.8
Llama 8B (5-shot)	24.7	57.1	29.2	30.6	52.2	34.8	27.4	46.6	26.6	27.6	54.0	31.4	24.4	45.7	27.5	27.1	51.8	30.1
Llama 70B (0-shot)	24.0	49.6	26.3	40.2	55.1	42.9	33.3	35.8	27.6	31.7	42.9	33.1	25.4	34.9	25.1	31.2	45.2	31.4
Llama 70B (5-shot)	25.3	53.0	28.1	41.6	56.4	44.4	31.4	32.6	25.4	29.9	39.4	31.1	26.7	35.7	26.4	31.3	45.2	31.5
Qwen 8B (0-shot)	38.7	51.4	40.8	49.0	64.0	52.6	34.6	53.2	36.6	29.2	61.3	35.5	27.2	45.5	31.1	37.1	55.9	40.5
Qwen 8B (5-shot)	39.2	52.9	41.6	48.9	64.4	52.4	35.2	53.5	36.6	28.8	61.1	35.2	26.7	45.7	30.4	37.2	56.4	40.5
Qwen 32B (0-shot)	37.0	77.8	45.5	43.4	68.9	48.5	24.0	62.5	30.8	20.9	72.4	29.0	18.4	50.1	24.0	30.3	68.0	37.2
Qwen 32B (5-shot)	35.4	77.5	44.3	40.3	68.6	45.9	24.5	62.0	30.7	21.1	71.3	29.4	18.4	51.6	24.3	29.4	67.7	36.5
Mistral 7B (0-shot)	39.0	62.1	44.2	46.2	59.5	50.0	33.6	40.3	29.1	31.7	46.6	34.3	28.6	44.7	31.1	36.9	51.4	38.6
Mistral 7B (5-shot)	38.3	61.6	43.6	46.6	60.2	50.5	33.3	40.2	29.1	31.4	46.0	34.0	28.2	43.3	30.6	36.6	51.2	38.5
Llama 1B (Fine-tuned)	71.2	71.0	70.5	81.9	81.8	81.8	65.1	63.3	60.3	78.8	77.2	76.6	72.9	69.7	67.0	73.6	72.5	71.3
Qwen 0.6B (Fine-tuned)	71.5	71.0	70.6	80.7	80.6	80.6	64.2	66.5	60.9	79.8	78.8	**77.9**	66.6	67.2	62.8	72.8	73.2	71.1
MODEE-Small	65.6	66.3	65.2	88.3	88.2	88.2	63.2	66.7	60.8	71.8	71.4	70.1	66.1	74.0	65.8	71.2	72.8	70.1
MODEE-Base	74.4	75.1	**74.0**	87.8	87.7	87.7	64.3	65.9	**61.1**	78.4	78.0	76.6	70.3	73.3	**68.2**	75.1	75.9	**73.7**

Table 5: Performance comparison on event extraction (%) using ROUGE-L. Bold indicates the best performance.

Models	Where			When			What			Who			Why			Overall		
	P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
T5-Small (Fine-tuned)	92.4	92.6	92.5	98.3	98.2	98.2	91.2	91.4	91.2	92.8	92.3	92.4	92.5	93.5	93.0	93.5	93.6	93.5
T5-Base (Fine-tuned)	93.0	93.2	93.1	98.3	98.2	98.3	91.7	92.2	91.9	93.8	93.6	93.6	93.3	93.9	**93.6**	94.1	94.2	94.1
T5-Large (Fine-tuned)	94.2	94.4	94.2	98.5	98.4	**98.4**	92.0	92.0	92.0	94.4	94.4	94.4	92.5	92.5	92.5	94.5	94.5	94.5
Giveme5W1H	84.6	84.4	84.4	94.3	94.8	94.5	84.7	85.4	85.0	85.5	83.8	84.4	83.3	81.6	82.4	86.6	86.3	86.3
Llama 8B (0-shot)	82.1	86.8	84.3	86.6	90.1	88.3	83.5	87.1	85.2	83.9	87.8	85.7	84.7	87.8	86.1	84.1	87.9	85.8
Llama 8B (5-shot)	82.2	86.9	84.4	86.4	89.9	88.0	83.7	87.3	85.4	83.6	87.5	85.4	84.1	87.4	85.7	83.9	87.8	85.7
Llama 70B (0-shot)	82.7	85.9	84.1	88.6	90.9	89.7	85.4	86.6	85.9	84.7	87.5	86.0	84.9	86.5	85.6	85.2	87.5	86.2
Llama 70B (5-shot)	83.3	86.4	84.7	89.0	91.3	90.1	85.0	86.2	85.5	84.3	87.0	85.5	85.2	86.7	85.9	85.3	87.6	86.3
Qwen 8B (0-shot)	85.9	87.7	86.8	90.7	92.8	91.7	86.4	88.6	87.4	83.4	89.0	86.0	86.1	88.1	87.1	86.7	89.4	88.0
Qwen 8B (5-shot)	86.2	88.0	87.0	90.6	92.8	91.6	86.5	88.6	87.5	83.3	88.8	85.8	86.0	87.9	86.9	86.7	89.4	88.0
Qwen 32B (0-shot)	84.8	90.2	87.4	88.8	92.3	90.5	83.8	89.0	86.3	80.8	88.8	84.5	83.3	87.7	85.4	84.6	89.9	87.1
Qwen 32B (5-shot)	84.6	90.1	87.2	88.2	91.9	90.0	84.0	88.9	86.3	81.1	89.1	84.8	83.3	87.6	85.3	84.5	89.8	87.0
Mistral 7B (0-shot)	86.0	89.1	87.4	91.0	93.0	91.9	85.6	87.0	86.2	84.9	88.1	86.3	85.6	87.8	86.6	86.8	89.2	87.9
Mistral 7B (5-shot)	85.9	89.0	87.3	91.1	93.1	92.1	85.6	87.0	86.2	84.7	88.1	86.3	85.4	87.8	86.5	86.8	89.2	87.9
Llama 1B (Fine-tuned)	93.6	93.7	93.6	97.7	97.7	97.7	92.1	92.2	92.1	95.0	94.5	94.7	93.4	93.1	93.2	94.4	94.4	94.4
Qwen 0.6B (Fine-tuned)	93.4	93.7	93.5	97.5	97.6	97.5	92.0	92.4	92.1	95.2	95.1	**95.1**	92.4	92.5	92.4	94.3	94.4	94.3
MODEE-Small	92.5	92.6	92.5	98.4	98.3	98.4	91.9	92.6	92.2	93.8	93.7	93.7	92.4	93.8	93.0	93.9	94.2	94.0
MODEE-Base	94.3	94.4	**94.3**	98.2	98.3	98.2	92.2	92.5	**92.3**	94.8	94.6	94.6	93.4	93.8	**93.6**	94.7	94.8	**94.7**

Table 6: Performance comparison on event extraction (%) using BERTScore. Bold indicates the best performance.