ACL Anthology ID 2026.findings-eacl.314 / revision 2 / 04 Sep 2026

Event Detection with a Context-Aware Encoder and LoRA for Improved Performance on Long-Tailed Classes

Abdullah Al Monsur
University of South Florida
almonsur@usf.edu

Nitesh Vamshi Bommisetty
University of South Florida
bn319@usf.edu

Gene Louis Kim
University of South Florida
genekim@usf.edu

Abstract

The current state of event detection research has two notable re-occurring limitations that we investigate in this study. First, the unidirectional nature of decoder-only LLMs presents a fundamental architectural bottleneck for natural language understanding tasks that depend on rich, bidirectional context. Second, we confront the conventional reliance on Micro-F1 scores in event detection literature, which systematically inflates performance by favoring majority classes. Instead, we focus on Macro-F1 as a more representative measure of a model's ability across the long-tail of event types. Our experiments demonstrate that models enhanced with sentence context achieve superior performance over canonical decoder-only baselines. Using Low-Rank Adaptation (LoRA) during finetuning provides a substantial boost in Macro-F1 scores in particular, especially for the decoder-only models, showing that LoRA can be an effective tool to enhance LLMs' performance on long-tailed event classes.

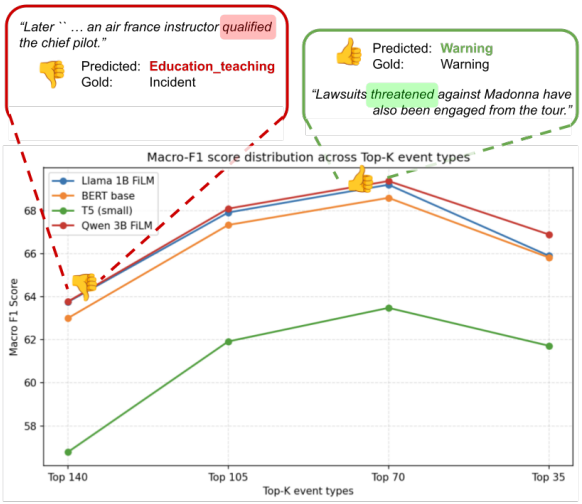

Figure 1: Macro-F1 scores across quartiles of event types, ordered by event mention frequency for four models, BERT, T5, Llama 1B FiLM, and Qwen 3B FiLM. We find that models consistently underperform on the events with fewest mentions (left side of the plot). The top of the diagram shows examples of high-frequency event, `Warning`, with a correct prediction and low-frequency event, `Incident`, with an incorrect prediction from Qwen 3B FiLM.

1 Introduction

The task of event detection (ED) is to identify and categorize the events in natural language (Grishman, 1997; Chinchor and Marsh, 1998; Ahn, 2006). This forms an important basis for structured interpretation of text and downstream applications that focus on event information (Zhang et al., 2020; Han et al., 2021).

The event detection literature is dominated by the encoder-only models, which are typically based on the outdated BERT models (Huang et al., 2024). This stands in contrast to the decoder-only models that dominate the current LLM landscape, such as Llama (Grattafiori et al., 2024) and Qwen (Bai et al., 2023). The pretraining on vast corpora has led to remarkable generative capabilities, but their unidirectional attention mechanism has been a roadblock in getting strong performance on embedding-based uses that access token embeddings before full access to the context processing (Saattrup Nielsen et al., 2025; Dukić and Snajder, 2024). We aim to address this issue in the context of event detection. We hypothesize that whole sentence context is critical for generalizable event detection performance, enabling models to distinguish underrepresented events more accurately. These event types with few training examples require differentiation of subtle variations with limited information and thus the full sentence context is likely needed.

Looking at evaluation, most recent event detection research use Micro-F1 scores, which hides generalizability of the model to long-tailed classes (Huang et al., 2024). In this paper, we make the case that Macro-F1 is an important complementary metric to assess the competence of models on

long-tailed classes. The importance of Macro-F1 is highlighted by the divergence of model rankings in our paper when using the two metrics.

Recent advances in parameter-efficient finetuning, such as LoRA, have shown promise in mitigating forgetting during adaptation. Our intuition is that this might help the LLMs to generalize more across the classes, improving the long-tailed class performance. Also, large models might overfit to the training data due to huge number of parameters. As a result, they might not perform well on the test or validation set. LoRA might be able to mitigate this issue.

In-context learning is, at times, faster and more efficient as it does not require training the model. As such, it may be preferred over the supervised finetuning (SFT) where training cost is a bottleneck. To evaluate how SFT differs in performance from in-context learning, we conducted few-shot and zero-shot experiments.

Our research questions are:

RQ1: What benefits, if any, are gained by introducing sentence-level information to decoder-only models for event detection? And which methods of doing so provide the greatest benefits?

RQ2: Does Macro-F1 capture a distinct and important aspect of event detection performance that is overlooked when only using Micro-F1 scores? That is, do event detection model improvements diverge between Micro-F1 and Macro-F1 scores in significant ways?

RQ3: How does LoRA impact event detection performance along both Micro- and Macro-F1 metrics?

Our key contributions of this paper are the following:

- We demonstrate that the performance of decoder-only LLMs for event detection can be significantly improved by incorporating sentence-level information during the finetuning process. The feature-wise linear modulation (FiLM; Section 3.3.3) method was overall most effective among them.
- Event detection methods show distinct Micro- and Macro-F1 performance behaviors, motivating the need to include Macro-F1 for complete comparisons.
- We find that beyond computational efficiency, LoRA acts as a potent regularizer that enhances generalization to low-frequency event types, leading to substantial gains in Macro-F1 scores across nearly all experimental setups.

2 Related Work

The introduction of the transformer architecture and large-scale pretrained language models altered the landscape of event detection. Models like BERT not only established a new state-of-the-art in performance but also inspired a conceptual shift in how the task itself was approached, moving from simple classification to more sophisticated, semantically-grounded paradigms (Hettiarachchi and Ranasinghe, 2023). Latest powerful decoder-only LLMs, have not yet been deeply investigated over their BERT-based counterparts on event detection benchmarks (Huang et al., 2024).

The MAVEN (Wang et al., 2020) dataset represents a significant leap forward in scale of event detection datasets, constructed from English Wikipedia articles. The RAMS (Ebner et al., 2020) is made from news articles which is another dataset with large number of event types.

LoRA offers a more computationally efficient method of finetuning LLMs by reducing the number of trainable parameters via low-rank matrices. This method allows the modification of every parameter of the original network without the full computational cost and has shown to achieve competitive performance to full-parameter finetuning (Hu et al., 2022).

Prior research has aimed to mitigate the shortcomings of decoder-only models. LLM2Vec (BehnamGhader et al., 2024) and Dec2Enc (Huang et al., 2025) are the two existing methods that introduce bidirectional attention to the decoder-only models. These methods have been evaluated across a wide range of tasks, but event detection evaluations are notably lacking.

The evaluation of ED and its sub-tasks has long been standardized around the metrics of Precision, Recall, and F1-score, which are micro-averaged (Simon et al., 2024). This poses a limitation of overlooking the model performance on long-tailed classes which are underrepresented in the dataset, and triggers the need to evaluate using macro-averaged metrics.

3 Methodology

3.1 Task Definition

We follow the event detection task definition of the MAVEN dataset (Wang et al., 2020) which splits

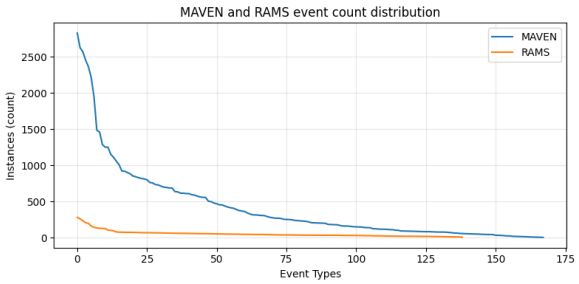

Figure 2: Distribution of event types by event mention instances in the MAVEN and RAMS dataset.

the problem into two parts **trigger identification (TI)**—the task of annotating which word spans in the text express an event in the sentence—and **trigger classification (TC)**—the task of classifying the event type that is expressed. Formally, in a sentence s of n words, $s = [w_1, w_2, ..., w_n]$, which expresses m events, $e_j \in E$ where $|E| = m$, the events are defined by the triple $e_j = (f_{e_j}, l_{e_j}, t_{e_j})$ where $[w_{f_{e_j}}, w_{l_{e_j}-1}]$ is the span of text that expresses event e_j and t_{e_j} is its event type. TI is then the task of extracting out a set of first and last word index pairs (f, l) and TC is the task of extracting out the set of triples (f, l, t) by adding in the type. Both tasks are evaluated using F1 scores. We focus on TC as TI is only an intermediate step for accomplishing TC.

3.2 Prompted LLMs

We categorize model designs into two broad categories. One where LLMs are only prompted using in-context learning and another where an LLM is used as a text embedding model in a wider neural architecture.

We additionally explore the use of LLMs through prompting, leveraging their in-context learning capabilities (Brown et al., 2020). This approach bypasses the need for computationally intensive model training by querying the model directly, offering a lightweight method for generalization.

We evaluate both few-shot and zero-shot prompting. All evaluations are performed without updating model parameters. In the few-shot setting, we supply natural language demonstrations with labeled examples, each consisting of a sentence, a trigger word, and an event type. We did not provide the event descriptions, as there can be some datasets without publicly available annotation guidelines. Therefore, we tried to experiment without such descriptions. The experiments were conducted with varying temperature and number of shots (Section C).

```
# Few-shot Prompt
input_sentence = join_tokens([[sentence_tokens]])

If a sentence contains multiple triggers, each trigger
should be output separately as an independent JSON entry.
Ignore any errors in the response and extract only the
relevant information needed for the JSON output.

prompt = " ".join([[few_shot_example]])

Identify the trigger word(s) and event type(s) from
the list in the sentence below.
The output must be strictly formatted as follows:
'[{{"sentence": ["word1", "word2", ...],
   "trigger": <trigger_text>,
   "e_start": <start_index>,
   "eventtype": <event_type>}}]'
```

Figure 3: Few-shot Prompt: Uses example-based contextual learning with explicit JSON formatting and multi-trigger handling.

```
# Zero-shot Prompt
For sentence: [[tokenized_sentence]], identify all words or
phrases that trigger an event.

Each trigger must be one or two words long. Two-word triggers
must be enclosed in the same double quotes.

Do not add comments, disclaimers, or examples before or
after the JSON output. Do not use any symbols other than
those explicitly required.
If a sentence contains multiple triggers, each trigger
should be output separately as an independent JSON entry.
Ignore any errors in the response and extract only the
relevant information needed for the JSON output.

The output must be strictly formatted as follows:
'[{{"sentence": ["word1", "word2", ...],
   "trigger": <trigger_text>,
   "e_start": <start_index>,
   "eventtype": <event_type>}}]'
```

Figure 4: Zero-shot Prompt: Relies purely on explicit instruction-based reasoning with structured JSON output.

3.3 LLM-as-Embedding

We use the base model from Wang et al.'s (2023) experiments (Section Q). This is a relatively simple but effective neural model using text embeddings (Figure 17). We use LLMs to generate embeddings for each word in the sentence which are then processed through a linear layer using dropout (Srivastava et al., 2014) and layer normalization (Ba et al., 2016) followed by a linear softmax classifier. The TC task is then converted to a word-level classification task where each word is classified into an event type or "NA" if it is not an event trigger. Outputs for TI are computed by merging all valid event types into a single "trigger" classification label. This is a supervised model where the classification layers are trained concurrently with the finetuning of the LLM, providing the embeddings. We do

not use the data augmentation, knowledge transfer, or pivotal knowledge distillation techniques in our experiments, as these techniques are designed for continual learning.

Following are the variations for incorporating sentence-level information that we tested for the decoder-only models. The BaseTE model neither introduces additional layers to the base model nor incorporates sentence-level context into the token representations. In contrast, the other architectures apply at least one of these modifications. All variations in this section have access to the LLM final layer hidden states, $H \in \mathbb{R}^{L \times D_{\text{LLM}}}$, where L is the sequence length and D_{LLM} is the LLM's hidden dimension. The variations are illustrated in Figure 5.

3.3.1 BaseTE: Baseline Trigger Encoder

Our baseline Trigger Encoder (TE) linearly projects final layer hidden states from LLMs into the target dimensions with dropout ($p = 0.2$) for regularization, GELU (Hendrycks and Gimpel, 2016) activation, and layer normalization (Ba et al., 2016).

3.3.2 ConcatPool

ConcatPool performs mean pooling and concatenation to produce sentence-contextualized hidden states.

1. Global Context Generation. First, a single sentence-level embedding, $s \in \mathbb{R}^D$, is produced by applying mean pooling across the sequence of initial L token representations, $s = \frac{1}{L}\sum_{i=1}^{L} h_i$.

2. Feature Concatenation. s is concatenated with each token representation, $h_i \in H$ to produce $c_i = [h_i; s]$, a combined feature vector for each token position where $c_i \in \mathbb{R}^{2D}$.

3. Linear Fusion. Finally, a linear fusion layer, with a learned weight matrix $W_f \in \mathbb{R}^{D \times 2D}$ and bias vector $b_f \in \mathbb{R}^D$, maps these concatenated vectors back to the original hidden dimension D, producing $h'_i = W_f c_i + b_f$. Here, h'_i is the final output representation.

3.3.3 FiLM: Feature-wise Linear Modulation

This variation is inspired by Perez et al. (2017), where a similar technique is successful in visual question answering. The core of the architecture is the FiLM-based context modulation, which consists of three steps.

1. Global Context Generation. Mean pooling-based sentence-level embedding production in the same manner as ConcatPool (Section 3.3.2).

2. FiLM Parameter Generation. This global context vector s is then passed through two separate linear layers, which act as a FiLM generator, to produce a feature-wise scaling vector γ and a shifting vector β.

$$\gamma = W_\gamma s + b_\gamma \quad \text{and} \quad \beta = W_\beta s + b_\beta \tag{1}$$

Feature-wise Modulation. Finally, the generated γ and β parameters are used to modulate the initial token representations H via an element-wise affine transformation.

$$H_{\text{FiLM}} = \gamma \odot H + \beta \tag{2}$$

3.3.4 BiSE-BiLSTM: Bidirectional Sequential Encoding using BiLSTM

This variation uses a bidirectional LSTM (Huang et al., 2015) over the sequence of transformer hidden states to produce new bidirectionally contextualized tokens. For each token h_i in the sequence H, the forward and backward LSTMs produce the forward $\overrightarrow{h_i}$ and backward $\overleftarrow{h_i}$ hidden states, respectively, using the following recursive computations: $\overrightarrow{h_i} = \text{LSTM}_{\text{fwd}}(h_i, \overrightarrow{h_{i-1}})$, $\overleftarrow{h_i} = \text{LSTM}_{\text{bwd}}(h_i, \overleftarrow{h_{i+1}})$. The final representation for each token, h'_i, is formed by concatenating the hidden states from both directions, $h'_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$.

3.3.5 Gated Context Fusion

This variation fuses token- and sequence-level information using a gating function through these three stages.

1. Global Context Generation. A single sequence-level embedding, $s \in \mathbb{R}^D$, is acquired by taking the final layer hidden state of the last token in the sequence.

2. Gate Computation. A fusion gate, $g_i \in [0, 1]^D$, is computed for each token representation $h_i \in H$. The concatenation of the token, h_i, and sequence, s, embeddings are passed through a linear layer (W_g, b_g) followed by a sigmoid activation function, σ.

$$g_i = \sigma(W_g[h_i; s] + b_g) \tag{3}$$

The gate g_i acts as an adaptive switch, determining the proportion of information to retain from each source in the next step.

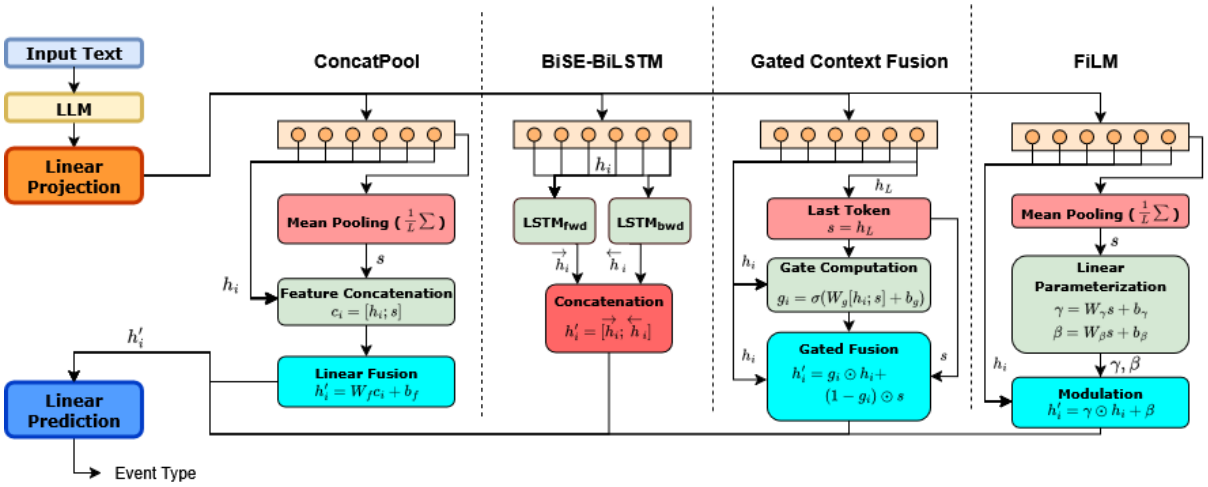

Figure 5: Diagrams for the four non-baseline LLM-as-Embedding variations to introduce sentence-level information to decoder-only models.

3. Dynamic Fusion. The final representation for each token, h'_i, is produced by performing a g_i-controlled weighted sum. This interpolates between the original token representation h_i and the sequence vector s.

$$h'_i = g_i \odot h_i + (1 - g_i) \odot s \quad (4)$$

where $\odot$ denotes element-wise multiplication.

4 Experimental Setup

4.1 Datasets

The chosen datasets are sentence level ED datasets which have large number of classes, as our experiments will be on long-tailed classes. We used 2 datasets: MAVEN (Wang et al., 2020) and RAMS (Ebner et al., 2020). MAVEN is a dataset of 4,480 Wikipedia documents, 118,732 event mentions, and 168 event types. The RAMS dataset comprises 139 event types and 9,124 event mentions from news articles.

4.2 Evaluation Metrics

We use Micro- and Macro-F1 trigger classification scores along with recall and precision. A prediction is labeled correct if and only if the word span and event type all match the gold labels. Micro-F1 is the standard metric used in prior work, where each event mention is weighted equally. We add Macro-F1 to our analysis, which is computed by taking the mean of the F1 score for each event type. This is a useful measure of generalizability because event type frequencies follow a long-tail distribution. Figure 2 shows the distribution of event type instances across the entire MAVEN and RAMS datasets. The MAVEN has a larger imbalance among the classes.

4.3 LLM Models

For LLM-as-embedding models, we consider BERT (Devlin et al., 2019)—specifically the bert-base-uncased model from Hugging Face (Wolf et al., 2020)—as well as Llama 3.2 1B, Qwen2 1.5B, bert-base-uncased, bert-large-uncased, RoBERTa-base, RoBERTa-large (Liu et al., 2019), T5-base, T5-large (Raffel et al., 2020), Llama 3.2 3B, Qwen2.5 3B, DeBERTa-base, and DeBERTa-large (He et al., 2021). The largest possible Llama models were selected based on the computational resources available.

For few-shot and zero-shot prompting experiments, we used Llama 3 8B (Grattafiori et al., 2024), Qwen2.5 7B (Qwen et al., 2025), Gemma 8B (Team et al., 2024), and DeepSeek 7B (DeepSeek-AI et al., 2024).

4.4 Baselines

We compare our model variations against fully fine-tuned models, zero-shot and few-shot results.

4.5 Hyperparameter Selection

The LLM-as-embedding model with BERT uses the best hyperparameters from Wang et al. (2023). A short hyperparameter search of the learning rate was performed for the other models. The exact default values for all models are provided in Section B.

Wherever LoRA is used in our experiments, it is applied to all the linear layers of the model.

5 Results

5.1 Supervised Finetuning Results

LoRA enhances Micro-F1 score of the decoder-only models. Table 4 illustrates that finetuning the decoder-only models (Llama 1B, Qwen 1.5B) alone did not surpass the smaller BERT or RoBERTa models in terms of Macro-F1 or Micro-F1 scores. With LoRA, however, the decoder-only models improved substantially, surpassing finetuned BERT models in F1 scores. We find that increasing the attention dropout of these models alone did not lead to the same performance gains (Table 3). This indicates that LoRA itself provides a boost in performance that is separate from typical regularization in addition to improving computational efficiency.

LoRA improves long-tailed class performance. LoRA increased the Macro-F1 scores for most of the models, especially all of the Llama and Qwen models, indicating that LoRA effectively handled long-tailed classes. How evaluation metrics change compared to the frequency of classes is shown in Section I. Scaling up model size tends to improve the performance, albeit with computational cost (Table 4).

Sentence-level context improves Llama and Qwen models, especially in terms of Macro-F1 scores. FiLM with mean pooling and BiSE-BiLSTM was the best overall across the models(Figure 6). Qwen 3B model with FiLM variation had the best Macro-F1 score (62.01%) for all experimental settings. For Qwen 1.5B, even the BiSE-BiLSTM variant was competitive with a Macro-F1 of 61.20%. For the Llama 1B model, FiLM had the best Macro-F1 score (61.38%). Almost all the context variations improved the decoder-only models.

Dec2Enc (Llama 1B) with LoRA outperforms the BaseTE Llama with LoRA model. But the FiLM, along with some other variations of Llama, turned out to have superior performance in terms of Macro-F1 scores. However, LLM2Vec (Llama 1B) with LoRA did not show any meaningful gain.

The decoder-only models, especially with those variations, showed a tendency to hold a high recall score. So, for tasks where high recall score is paramount, these models can be a good choice.

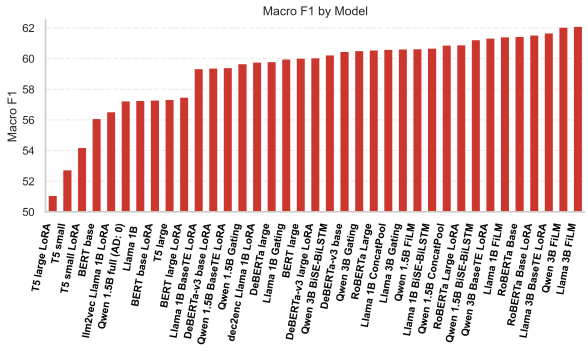

Figure 6: Macro-F1 performance of all finetuned models on MAVEN.

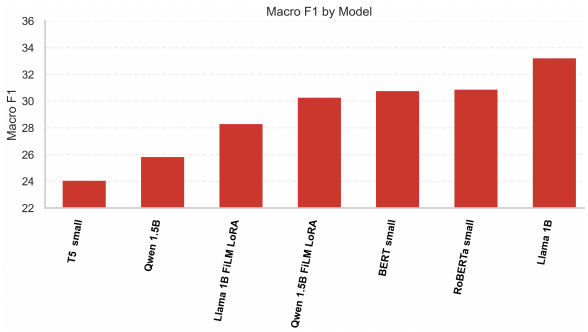

Figure 7: Macro-F1 performance of finetuned models on RAMS.

The T5 models have a unique ability to hold high precision scores. If any task requires conservative prediction of events, then T5 is a good candidate. The DeBERTa-v3 large model with LoRA finetuning had the best Macro-Recall (67.47%) among all the models. The Micro-F1 score, Precision, and Recall are illustrated in Appendices E, G, and F.

Experiments on the RAMS dataset showed a similar pattern in the results. The only notable exception is that the full Llama finetuning has a better Macro-F1 than the Llama-FiLM variation (Table 2). Also, in this case, BERT and RoBERTa showed slightly better Macro-F1 scores than Llama and Qwen variations. Qwen FiLM still has the second-best Micro-F1 score after RoBERTa (Figure 7).

However, the difference between the few-shot results and finetuning results for RAMS is minimal. This suggests that for smaller datasets, finetuning might not boost the performance of these models compared to the few-shot setting.

Dataset	Model	Size	Temp	Shots	Min Count	Shot Type	Macro-P(%)	Macro-R(%)	Micro-F1(%)	Macro-F1(%)
RAMS	Llama 3	8B	0.4	3%	3	Few-shot	**26.25**	29.87	**34.23**	27.94
	Qwen 2.5	7B	0.4	3%	3	Few-shot	26.14	**30.52**	33.32	**28.16**
MAVEN	Llama 3	8B	0.4	3%	3	Few-shot	**56.82**	**51.72**	**63.95**	**54.13**
	Qwen 2.5	7B	0.4	3%	3	Few-shot	55.51	50.68	61.87	52.99
	Gemma	8B	0.4	3%	1	Few-shot	54.02	49.21	61.31	51.49
	DeepSeek	7B	0.4	1%	3	Few-shot	53.13	48.14	61.01	50.53
	Llama 3	8B	0.4	0%	0	Zero-shot	51.36	46.19	36.10	24.11
	Qwen 2.5	7B	0.4	0%	0	Zero-shot	50.11	45.15	33.23	22.75
	Gemma	8B	0.4	0%	0	Zero-shot	51.24	46.29	34.76	22.95

Table 1: Best-performing few-shot and zero-shot configurations

Model	Variation	Size	LoRA	Micro-P(%)	Micro-R(%)	Macro-P(%)	Macro-R(%)	Micro-F1(%)	Macro-F1(%)
Llama	(None)	1B	no	35.01	31.48	32.20	27.89	33.15	**33.19**
Llama	FiLM	1B	yes	37.43	34.30	34.73	28.48	35.78	28.28
Qwen	(None)	1.5B	no	36.39	32.68	32.88	24.64	34.43	25.82
Qwen	FiLM	1.5B	yes	38.58	37.37	35.76	31.38	37.97	30.25
BERT	(None)	small	no	37.66	36.40	34.96	30.39	37.02	30.74
RoBERTa	(None)	small	no	**41.45**	**41.45**	**37.05**	**33.25**	**41.45**	30.85
T5	(None)	small	no	26.52	38.63	25.00	30.50	31.45	24.04

Table 2: Evaluation metrics for the RAMS dataset.

Model	Dropout	Size	Micro-F1(%)	Macro-F1(%)
Llama (full)	0 (Default)	1B	64.37	57.23
	0.3	1B	65.89	57.22
Qwen (full)	0 (Default)	1.5B	65.89	57.21
	0.3	1.5B	65.05	57.68

Table 3: Performance when varying dropout rates while finetuning the whole model.

5.2 Results of Prompt-based Techniques

We evaluated the performance of several instruction-tuned large language models (LLMs) on the MAVEN event detection dataset.

5.2.1 Zero-Shot Performance

As expected, zero-shot prompting yielded substantially lower performance, especially in Macro-F1, highlighting the difficulty LLMs face when generalizing to tail classes without in-context examples. Llama 3 (8B) achieved the best zero-shot Micro-F1 of 36.10%, but the Macro-F1 dropped to 24.11%. Qwen 2.5 (7B) and Gemma (8B) showed similar trends, with Macro-F1 values of 22.75% and 22.95%, respectively.

5.2.2 Few-Shot Performance

Few-shot prompting consistently outperformed zero-shot prompting across all models in both Micro-F1 and Macro-F1 metrics, underscoring the effectiveness of in-context learning for long-tailed event detection. Among the models, Llama 3 (8B) achieved the highest performance on the MAVEN dataset, with its best configuration—3% shots, min count of 3, and temperature 0.4—yielding a Micro-F1 of 63.95% and Macro-F1 of 54.13%.

On the RAMS dataset, the same configuration of Llama 3 (8B) achieved a Micro-F1 of 34.23% and Macro-F1 of 27.94%, while Qwen 2.5 (7B) obtained a Micro-F1 of 33.32% and Macro-F1 of 28.16%. These results indicate that few-shot learning significantly enhances generalization across both large-scale and domain-specific event extraction benchmarks. The few-shot results hallucinated by introducing new event types, which is discussed in Section P.

5.3 Statistical Significance Testing

We conducted a paired t-test (Virtanen et al., 2020) on the Macro-F1 scores to determine whether our findings differ significantly from each other. The results showed that the Variations of Llama and Qwen models are statistically significantly different ($p<0.05$) than the fully finetuned models, BERT, and T5 models. But, some of their F1 scores are not significantly different than the RoBERTa model predictions (Figure 8).

5.4 Impact of LoRA on TextEE Models

TextEE leverages generative models as decoders for conditional text generation, which is quite different from how we use them. Instead, our work utilizes generative models as encoders to obtain rich contextual representations. Among the different types of models, Dygiepp is one that perform span-based event detection similar to us. It is one of the best models for event detection according to the paper (Huang et al., 2024) We reproduced their results and also used LoRA to see their impact on

Base Model	Variation	Size	LoRA	Macro-P(%)	Macro-R(%)	Micro-F1(%)	Macro-F1(%)
Llama	BaseTE	1B	Yes	64.02	58.54	67.79	59.31
	Gating		Yes	63.54	60.11	68.26	59.94
	FiLM		Yes	62.44	64.14	68.13	61.38
	BiSE-BiLSTM		Yes	63.33	62.31	67.76	60.64
	ConcatPool		Yes	59.40	65.14	67.70	60.56
	(None)		No	61.32	55.84	64.37	57.23
	BaseTE	3B	Yes	65.71	61.14	68.61	61.63
	Gating		Yes	65.96	59.60	67.90	60.59
	FiLM		Yes	60.76	**65.97**	68.13	**62.07**
Qwen	BaseTE	1.5B	Yes	63.63	59.17	68.11	59.37
	Gating		Yes	65.41	58.29	67.16	59.63
	FiLM		Yes	59.25	65.86	67.86	60.61
	BiSE-BiLSTM		Yes	65.18	61.47	67.76	61.20
	ConcatPool		Yes	60.79	64.41	67.64	60.85
	(None)		No	62.03	57.03	65.89	57.21
	BaseTE	3B	Yes	65.49	61.56	68.77	61.30
	FiLM		Yes	62.52	64.34	68.05	62.01
	Gating		Yes	64.62	60.83	67.91	60.48
	BiSE-BiLSTM		Yes	63.44	60.31	67.71	60.20
Dec2Enc Llama		1B	Yes	65.36	58.47	68.24	59.74
LLM2Vec Llama		1B	Yes	61.33	55.23	66.1	56.49
BERT	_	base	No	58.31	59.33	67.61	56.05
		base	Yes	63.65	55.36	67.26	57.26
		large	No	64.51	58.81	67.48	59.99
		large	Yes	64.72	55.03	67.52	57.45
T5	_	large	No	63.10	56.56	66.10	57.30
		large	Yes	**66.70**	44.73	61.93	51.03
		small	No	60.92	50.32	64.25	52.71
		small	Yes	65.11	50.53	64.45	54.16
Roberta	_	base	No	63.18	62.48	**68.93**	61.41
		large	No	64.49	60.23	68.01	60.53
		base	Yes	64.77	63.07	68.83	61.50
		large	Yes	61.56	63.46	67.61	60.86
DeBERTa-v3	_	base	No	62.39	60.45	66.78	60.43
		base	Yes	61.28	61.68	67.93	59.35
		large	No	59.75	62.69	68.19	59.76
		large	Yes	61.99	61.2	68.37	60.01

Table 4: Evaluation metrics for the MAVEN dataset.

Base Model	LoRA	Micro-F1(%)	Micro-P(%)	Micro-R(%)
RoBERTa	No	65.33	62.47	68.46
RoBERTa	Yes	65.39	64.92	64.83

Table 5: Results of DyGIEpp (with and without LoRA) on MAVEN.

the outcome.

We experimented with only our split of the dataset. The results are shown in Table 5. It shows that even though the Micro-F1 is very similar for both approaches (with and without LoRA), the Micro-Precision improves when LoRA is used. This gain, however, comes at the cost of a decrease in Micro-Recall.

6 Conclusion

Introducing bidirectional attention or sentence context boosted the effectiveness of the decoder-only models. It enabled Llama and Qwen surpass the BERT models and made them at least as effective as RoBERTa models. This shows that the decoder-only models can be powerful even for ED tasks when necessary adaptations are made.

Additionally, different models exhibit varying strengths in terms of precision and recall. Our study offers practical guidance for selecting models based on the specific needs of an application, whether it prioritizes precision or recall.

Further research on how sentence length affects these models will give us better insight. As most of the recent LLMs are generative in nature and built upon decoder-only architectures, exploring their adaptation for event detection remains a crucial and promising direction for future research. These adaptations can further help develop more robust NLU systems as well.

Limitations

Error and difficulty analysis. A breakdown by context length and type rarity, showing where improvements occur, would give us a better idea of the function of the models.

Prompt technique exploration. We did not investigate advanced prompt-based or instruction-based methods in depth, so their impact on macro-averaged metrics was not explored.

Hyperparameter tuning. Hyperparameter search was necessarily bounded; a wider sweep exploration might yield additional gains.

Testing on multiple splits. We used a single data split. Testing multiple random splits and seeds would provide a clearer assessment of variance and outcome consistency.

Comparing the results against the latest and best models. The results were not compared with the other latest and state-of-the-art models. As approaches and evaluation techniques differ for each model and framework, comparing the results of our model with the other ones requires reproducing the results in a unified framework. It was not done in this project.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney, Australia. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Nancy Chinchor and Elaine Marsh. 1998. Appendix D: MUC-7 information extraction task definition (version 5.1). In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*.

DeepSeek-AI, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. 2024. Deepseek llm: Scaling open-source language models with longtermism.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*.

David Dukić and Jan Snajder. 2024. Looking right is sometimes right: Investigating the capabilities of decoder-only LLMs for sequence labeling. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14168–14181, Bangkok, Thailand. 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.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern,

Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo,

Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models.

Ralph Grishman. 1997. Information extraction: Techniques and challenges. In *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*, pages 10–27, Berlin, Heidelberg. Springer Berlin Heidelberg.

Rujun Han, I-Hung Hsu, Jiao Sun, Julia Baylon, Qiang Ning, Dan Roth, and Nanyun Peng. 2021. ESTER: A machine reading comprehension dataset for reasoning about event semantic relations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7543–7559, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *ICLR*.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Hansi Hettiarachchi and Tharindu Ranasinghe. 2023. Explainable event detection with event trigger identification as rationale extraction. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 507–518, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Kuan-Hao Huang, I-Hung Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, and Heng Ji. 2024. TextEE: Benchmark, reevaluation, reflections, and future challenges in event extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12804–12825, Bangkok, Thailand. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

Zixian Huang, Xinwei Huang, Ao Wu, Xiaxia Wang, and Gong Cheng. 2025. Transforming decoder-only models into encoder-only models with improved understanding capabilities.

Yinhan Liu et al. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. 2017. Film: Visual reasoning with a general conditioning layer. volume abs/1709.07871.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report.

Colin Raffel et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Dan Saattrup Nielsen, Kenneth Enevoldsen, and Peter Schneider-Kamp. 2025. Encoder vs decoder: Comparative analysis of encoder and decoder language models on multilingual NLU tasks. In *Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025)*, pages 561–572, Tallinn, Estonia. University of Tartu Library.

Étienne Simon, Helene Olsen, Huiling You, Samia Touileb, Lilja Øvrelid, and Erik Velldal. 2024. Generative approaches to event extraction: Survey and outlook. In *Proceedings of the Workshop on the Future of Event Detection (FuturED)*, pages 73–86, Miami, Florida, USA. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J. Carey, Ilhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.

Zitao Wang, Xinyi Wang, and Wei Hu. 2023. Continual event extraction with semantic confusion rectification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11945–11955, Singapore. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, et al. 2020. Transformers: State-of-the-art natural language processing. *EMNLP Demos*.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. 2020. Aser: A large-scale eventuality knowledge graph. In *Proceedings of The Web Conference 2020*, WWW '20, page 201–211, New York, NY, USA. Association for Computing Machinery.

A Additional evaluation Metrics

Here are Micro-Precision and Micro-Recall scores of our fine-tuned models in table 6

B Default Hyperparameters

The default hyperparameters of the LLM-as-embedding model (Section 3.3) are shown in Table 8.

C Hyperparameters for Prompting Techniques

To comprehensively assess few-shot performance under limited-resource settings, we experimented with four configurations: 1% shots with 1-min count, 1% shots with 3-min count, 3% shots with 1-min count, and 3% shots with 3-min count. These combinations were chosen to balance *prompt space constraints* and *event type coverage*. The 1% settings mimic extremely low-resource conditions, while 3% shots provide broader contextual diversity. Enforcing a minimum example count of 3 per event type ensures that even rare classes are represented, thereby testing model generalization to tail events. Conversely, a 1-min count allows for more event types to be included but may underrepresent rarer classes. This setup enables analysis of the trade-off between *event diversity* and *example richness per type*.

Moreover, a decoding temperature of 0.4 generally led to improved results compared to 0.0, likely due to increased output diversity and reduced repetition. These findings highlight the challenge of long-tailed event detection and demonstrate that even minimal prompting, when carefully structured, can substantially enhance performance.

D P values for t-test

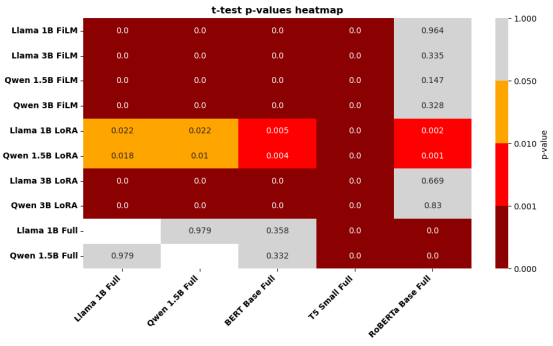

Figure 8: p values for t-test

E Micro-F1 scores for fine-tuning results

Figure 9 captures the overall performance by computing global precision and recall across all event types. It is particularly useful when evaluating models on imbalanced datasets, as it favors frequently occurring classes.

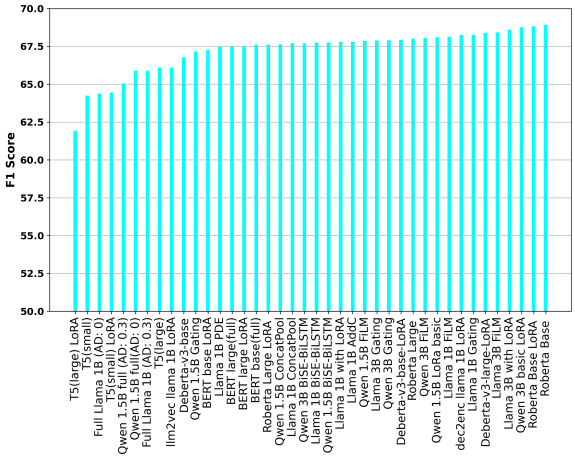

Figure 9: Micro-F1 scores of all fine-tuned models on MAVEN

F Recall for fine-tuning results

Figure 10 aggregates true positives across all classes, emphasizing overall sensitivity to detecting events. In contrast, Figure 11 treats all classes equally, offering a balanced view of recall performance, especially highlighting how well rare event types are recovered.

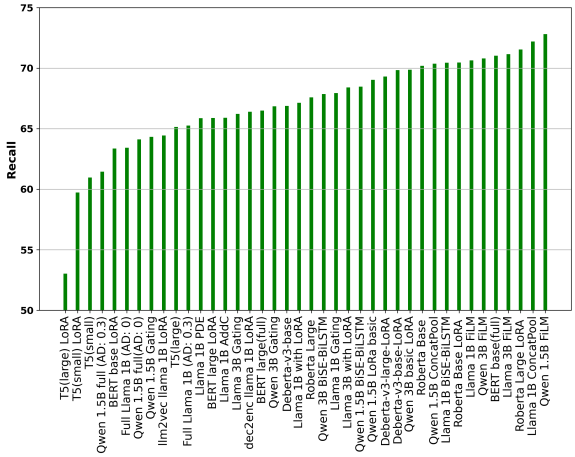

Figure 10: Micro-Recall scores of all fine-tuned models on MAVEN.

Model	Variation	Size	LoRA	Micro-P(%)	Micro-R(%)
Llama	BaseTE	1B	Yes	68.47	67.13
	Gating	1B	Yes	68.58	67.94
	FiLM	1B	Yes	65.80	70.63
	BiSE-BiLSTM	1B	Yes	65.26	70.45
	ConcatPool	1B	Yes	63.73	72.20
	BaseTE	3B	Yes	68.81	68.40
	Gating	3B	Yes	69.68	66.22
	FiLM	3B	Yes	65.15	71.41
	Dropout: 0	1B	No	65.35	63.42
	Dropout: 0.3	1B	No	66.54	65.25
DEC2ENC Llama		1B	Yes	70.19	66.39
LLM2VEC Llama		1B	Yes	67.85	64.44
BERT	_	base	No	64.51	71.03
		base	Yes	68.91	63.36
		large	No	68.51	66.49
		large	Yes	69.26	65.88
T5	_	large	No	67.10	65.14
		large	Yes	74.44	53.03
		small	No	67.93	60.96
		small	Yes	70.00	59.73
Roberta	_	base	No	67.74	70.19
		large	No	68.45	67.59
		base	Yes	67.28	70.46
		large	Yes	64.10	71.53
Qwen	BaseTE	1.5B	Yes	67.22	69.03
	Gating	1.5B	Yes	70.25	64.33
	FiLM	1.5B	Yes	63.55	72.81
	BiSE-BiLSTM	1.5B	Yes	67.05	68.47
	ConcatPool	1.5B	Yes	65.11	70.37
	Dropout: 0	1.5B	No	67.77	64.12
	Dropout: 0.3	1.5B	No	69.09	61.45
	BaseTE	3B	Yes	67.71	69.87
	FiLM	3B	Yes	65.53	70.79
	Gating	3B	Yes	69.02	66.84
	BiSE-BiLSTM	3B	Yes	67.58	67.85
DeBERTa-v3	_	base	No	66.71	66.87
		base	Yes	66.13	69.84
		large	No	67.02	69.42
		large	Yes	67.47	69.30

Table 6: Micro-Precision and Micro-Recall scores for MAVEN dataset.

G Precision for fine-tuning results

Figure 12 reflects the model's ability to avoid false positives across the dataset, while Figure 13 averages performance over all event types, revealing how precisely both common and rare events are identified

H Testing With Different Splits

We experimented with the five splits that were used in the TextEE paper (Table 9). Also, the average scores are very close to the scores of the split used for all experiments in our paper. These experiments were on the MAVEN dataset.

I Metrics According to Sample Frequency

Figures 14-16 show distributions across Top-k event types and provide insight into how model performance scales with class frequency. This highlights whether models generalize well to both high frequency classes and tail classes.

J U-shaped Pattern

Figures 14, 15, and 16 show a U-shaped pattern. In addition to event frequency, other factors may influence class performance. Among the 40 most frequent event types, some may be closely related to others, which can lead to misclassification. For instance, "Motion", "Motion-directional", and "Self-motion" can be very difficult to distinguish from each other. All of these 3 are among the top 40 most frequent event types. Similarly, Attack and hostile encounter can be confused between each other. They belong to top 10 most frequent event types. These confusions might be a reason for lower F1 scores on those quantiles.

K Impact of LoRA Rank on the Models

We experimented with multiple ranks of LoRA on the BERT and Llama 1B model. What can be observed from the tables is that LoRA ranks that are higher than rank 8 do not yield additional

Model	Size	Temp	Shots	Min Count	Shot Type	Micro-P(%)	Micro-R(%)	Micro-F1(%)	Macro-P(%)	Macro-R(%)	Macro-F1(%)
LLaMA 3	8B	0.0	1%	1	Few-shot	73.84	50.67	60.10	51.36	46.19	48.67
		0.4	1%	1		74.45	51.38	60.80	51.58	46.52	48.96
		0.4	1%	3		73.81	50.95	60.28	50.69	45.61	48.02
		0.0	3%	3		75.06	51.90	61.37	54.17	48.61	51.28
		0.4	3%	3		77.94	54.21	**63.95**	56.82	51.72	**54.13**
		0.4	3%	1		77.39	53.62	63.35	56.12	51.64	53.79
Qwen 2.5	7B	0.0	1%	1	Few-shot	74.29	50.24	59.94	50.11	45.15	47.50
		0.4	1%	1		74.02	50.50	60.04	51.69	46.76	49.11
		0.0	3%	3		74.42	51.42	60.82	52.64	47.38	49.87
		0.4	3%	3		75.34	52.48	61.87	55.51	50.68	52.99
		0.4	1%	3		74.35	50.37	60.05	50.23	45.82	47.94
		0.4	3%	1		75.39	52.15	61.65	55.37	50.34	52.74
Gemma	8B	0.0	1%	1	Few-shot	74.75	51.16	60.78	51.24	46.29	48.64
		0.4	1%	1		74.51	51.43	60.85	51.98	46.77	49.21
		0.0	3%	3		74.18	51.61	60.87	51.96	47.13	49.42
		0.4	3%	3		75.40	52.66	62.01	54.03	48.81	51.29
		0.4	1%	3		74.72	51.30	60.83	51.54	46.49	48.87
		0.4	3%	1		74.33	52.17	61.31	54.02	49.21	51.49
DeepSeek	7B	0.4	1%	3	Few-shot	74.21	51.80	61.01	53.13	48.14	50.53
			3%	1		73.81	50.95	60.29	51.86	46.73	49.15
LLaMA 3	8B	0.4	0%	-	Zero-shot	44.61	30.32	36.10	51.36	46.19	24.11
Qwen 2.5	7B			-		41.75	27.60	33.23	50.11	45.15	22.75
Gemma	8B			-		43.51	28.94	34.76	51.24	46.29	22.95

Table 7: Few-shot and Zero-shot Event Detection Metrics.

Hyperparameter	Value
Hidden dimension	100
Last layer dropout	0.2
Optimizer	AdamW
Learning rate (LLaMA, Qwen, RoBERTa, T5, DeBERTa)[1]	5×10^{-6}
Learning rate (BERT)	5×10^{-5}
Warmup proportion	0.1
Weight decay	0.01

Table 8: Default hyperparameters used for LLM-as-embedding models.

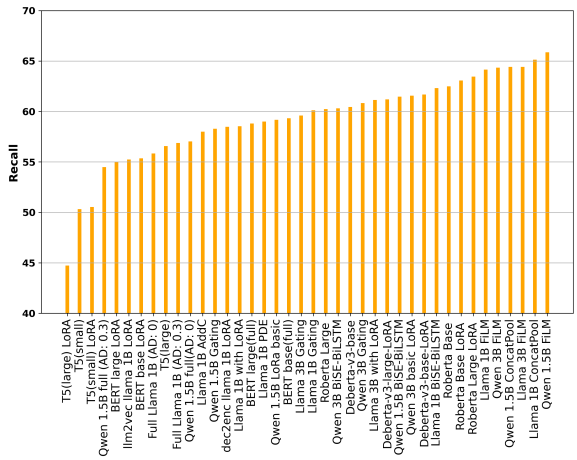

Figure 11: Macro-Recall scores of all fine-tuned models on MAVEN.

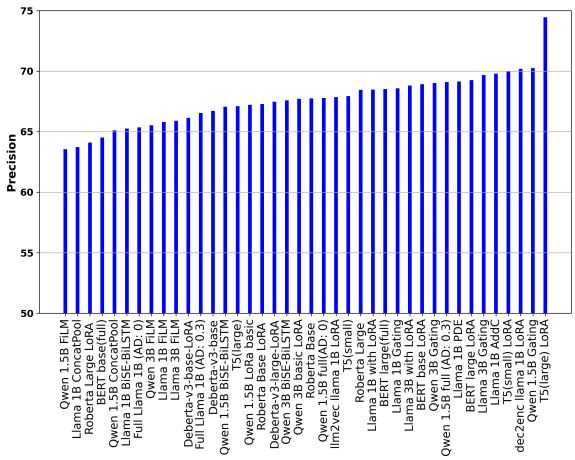

Figure 12: Micro-Precision scores of all fine-tuned models on MAVEN

performance improvements. Experiments on the MAVEN dataset are shown in table 10.

[1]For DeBERTa-large full fine-tuning, LR $= 1 \times 10^{-6}$.

L Performance Changes Across Quantiles Induced by LoRA

After further analyzing the impact of LoRA, we find the following comparison between Llama 1B full finetuning and Llama 1B LoRA finetuning

Model	Micro-F1 (%)
Llama full finetuning (our model)	64.32
Llama FiLM (our model)	68.04

Table 9: Micro-F1 performance comparison of Llama-based models.

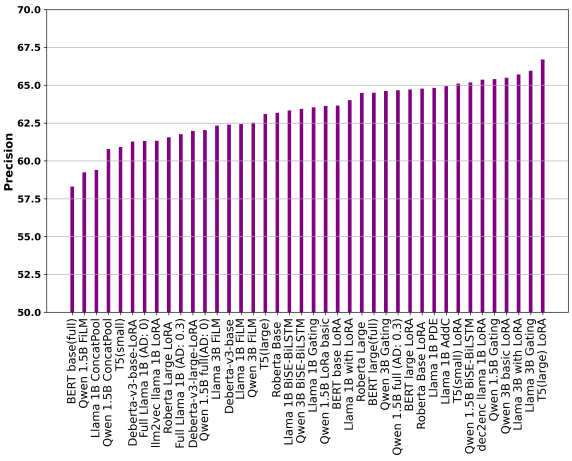

Figure 13: Macro-Precision performance of all fine-tuned models on MAVEN.

Model	LoRA Rank	Micro-F1	Macro-F1
BERT small	4	66.14	55.78
BERT small	8	67.26	57.26
BERT small	16	67.18	55.01
BERT small	64	56.17	66.53
LLaMA 1B	2	57.89	66.83
LLaMA 1B	4	58.92	67.46
LLaMA 1B	8	59.17	67.58
LLaMA 1B	16	58.13	66.75

Table 10: Effect of LoRA rank on Micro-F1 and Macro-F1 for BERT-small and LLaMA-1B models.

from each quantile. The experiment is done on the MAVEN dataset. The table 11 shows Macro-F1 (%) improvement from Llama full finetuning to Llama LoRA finetuning. We subtracted the two f1 scores for each quantile.

As we can see, LoRA increases the F1 scores of lesser frequent events more than the frequent ones.

M Effect of Upsampling

Naive upsampling does not improve performance as shown in Table 12. In fact, both macro and micro F1 scores slightly decrease when upsampling is applied.

N Impact of Input Format on Llama

The table 13 shows that when a preceding space is not used for each word token, the performance of Llama drops significantly. This is because Llama's tokenizer requires a leading space before every word in order to correctly identify word boundaries during tokenization.

O Computational Resource and Time

The GPU configuration is as follows:
GPU Model: NVIDIA RTX A6000
Memory (VRAM): 48 GB GDDR6
Memory (RAM): 256 GB (4x 64GB DIMM DDR4 Synchronous 3200 MHz; Samsung M393A8G40AB2)
CPU: AMD Ryzen Threadripper PRO 5955WX 16-Cores

The table 14 shows the approximate average time for experiments with and without LoRA. LoRA saves both memory and time while finetuning a model.

P Hallucination of Out-of-List Event Types

During evaluation, large language models occasionally hallucinated event types that were absent from the predefined ontology, generating labels not included in the allowed event list. Such out-of-list predictions indicate the model's tendency to rely on *semantic intuition* rather than strict schema constraints, highlighting a key limitation of open-ended generation. Although these hallucinations were relatively infrequent, they inflated false positives and distorted macro-level metrics. To mitigate this, all outputs were post-validated against the canonical event list, replacing invalid types with a placeholder label such as `NONE`.

Q Overview of Our Event Detection Approach

The figure 17 illustrates a general overview of our event detection framework.

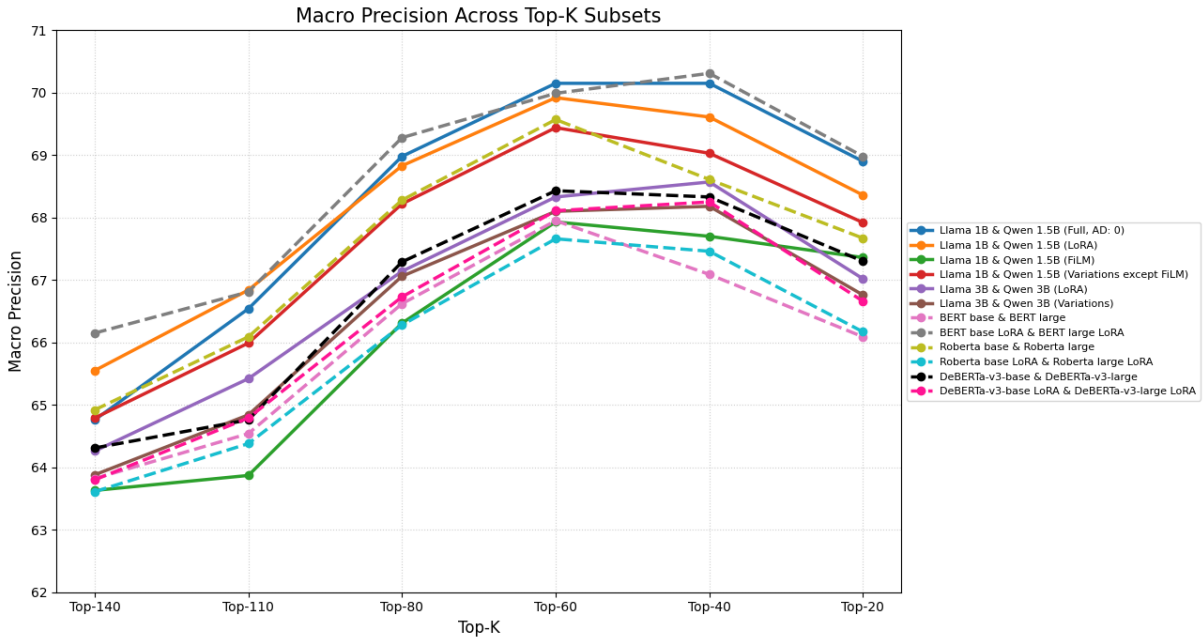

Figure 14: Macro-Precision score distribution across top-k event types

Metric	Top-140	Top-110	Top-80	Top-60	Top-40	Top-20
Performance Change	+3.02	+3.16	+3.15	+2.94	+2.98	+2.66

Table 11: Macro-F1 score (%) changes across different top-k quantiles after adding LoRA (Llama 1B model with vs without LoRA).

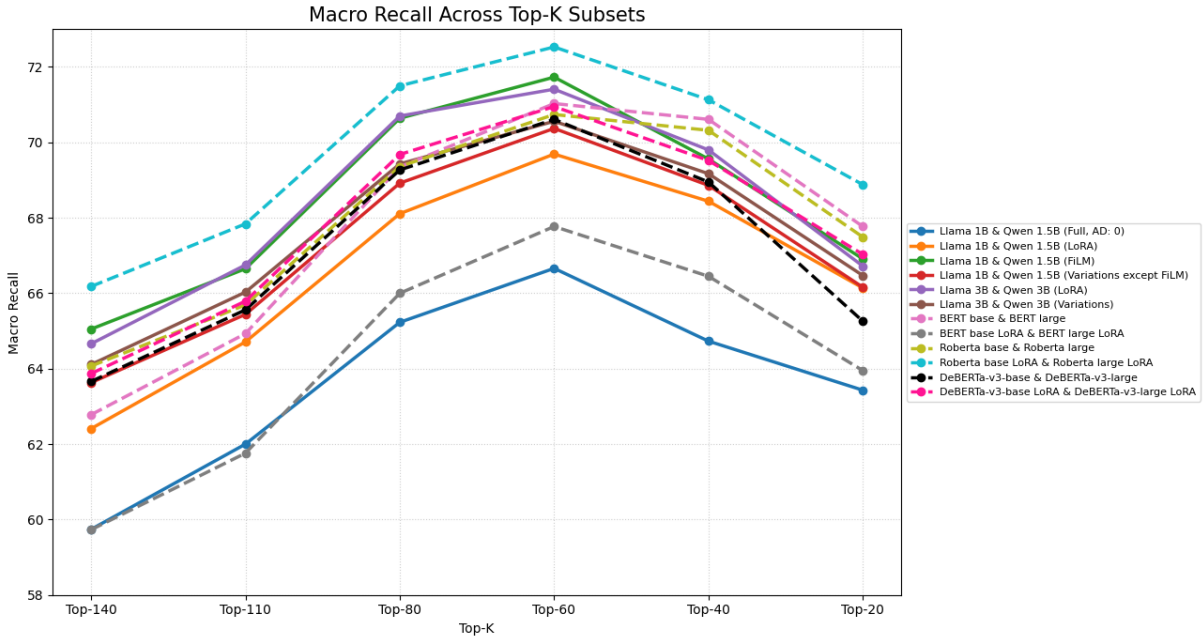

Figure 15: Macro-Recall score distribution across top-k event types

Model	Macro-F1	Micro-F1
BERT (LoRA rank-16) with upsampled dataset	48.19	65.07
BERT (LoRA rank-16) no upsample	**52.99**	**66.12**

Table 12: Effect of naive upsampling on BERT rank-16.

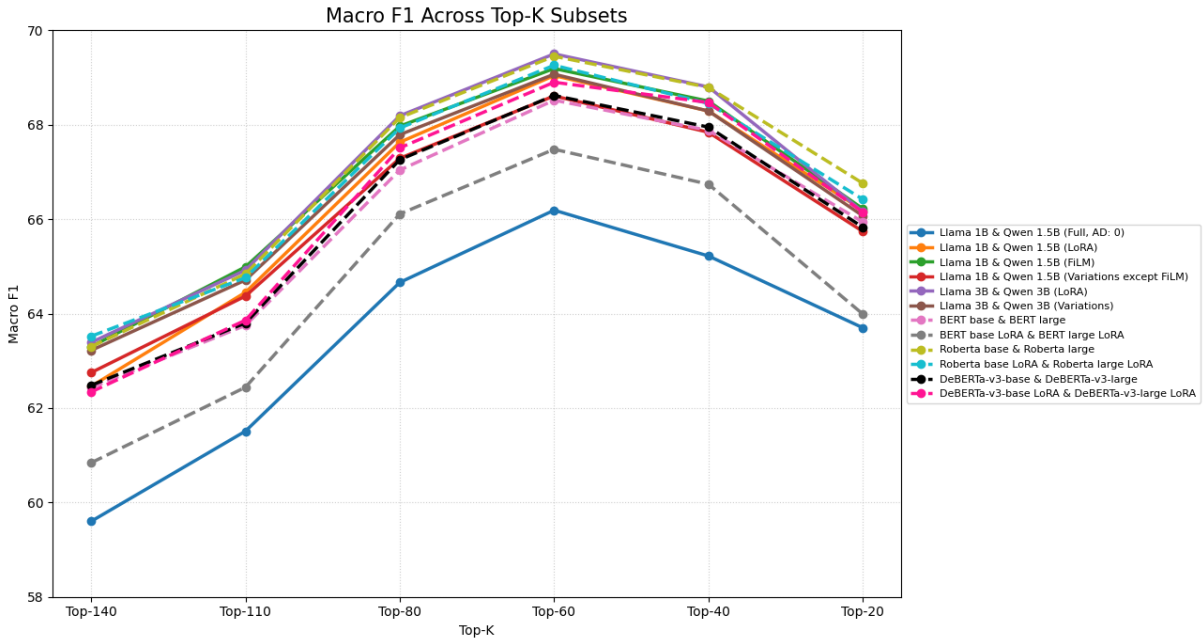

Figure 16: Macro-F1 score distribution across top-k event types.

Input Format (Leading Space Tokens)	LoRA	Micro-F1	Macro-F1
No	No	55.15	39.08
No	Yes	60.23	48.26
Yes	No	64.37	57.23
Yes	Yes	67.79	59.31

Table 13: Impact of input formatting and LoRA on model performance.

Model	LoRA (Rank 8)	Approx. Avg. Time (minutes)
LLaMA 1B	Yes	600
LLaMA 1B	No	900
BERT Small	Yes	300
BERT Small	No	420

Table 14: Approximate average training time with and without LoRA (rank 8).

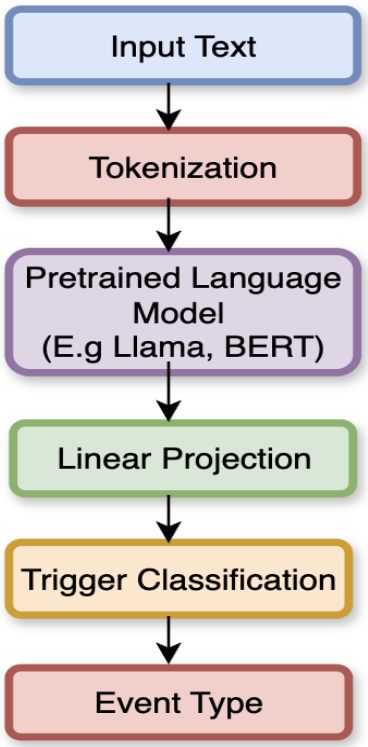

Figure 17: Overview of the Event Detection Framework based on Wang et al. (2023).