Retrieval of Temporal Event Sequences from Textual Descriptions

Zefang Liu*
Georgia Institute of Technology
Atlanta, GA 30332, USA
liuzefang@gatech.edu

Yinzhu Quan*
Georgia Institute of Technology
Atlanta, GA 30332, USA
yquan9@gatech.edu

Abstract

Retrieving temporal event sequences from textual descriptions is crucial for applications such as analyzing e-commerce behavior, monitoring social media activities, and tracking criminal incidents. To advance this task, we introduce TESRBench, a comprehensive benchmark for temporal event sequence retrieval (TESR) from textual descriptions. TESRBench includes diverse real-world datasets with synthesized and reviewed textual descriptions, providing a strong foundation for evaluating retrieval performance and addressing challenges in this domain. Building on this benchmark, we propose TPP-Embedding, a novel model for embedding and retrieving event sequences. The model leverages the TPP-LLM framework, integrating large language models (LLMs) with temporal point processes (TPPs) to encode both event texts and times. By pooling representations and applying a contrastive loss, it unifies temporal dynamics and event semantics in a shared embedding space, aligning sequence-level embeddings of event sequences and their descriptions. TPP-Embedding demonstrates superior performance over baseline models across TESRBench datasets, establishing it as a powerful solution for the temporal event sequence retrieval task.

1 Introduction

Temporal event sequence retrieval (Gupta et al., 2022) plays a crucial role in various applications, such as e-commerce user activity analysis, social media monitoring, and crime tracking. These sequences combine temporal information with event types, making them more complex than traditional text data. Effective retrieval requires models capable of capturing both time-sensitive dynamics and structured relationships within the sequences. While traditional language models perform well for general text retrieval (Kashyap et al., 2024), they often struggle to handle the unique temporal and structural complexities of event sequences.

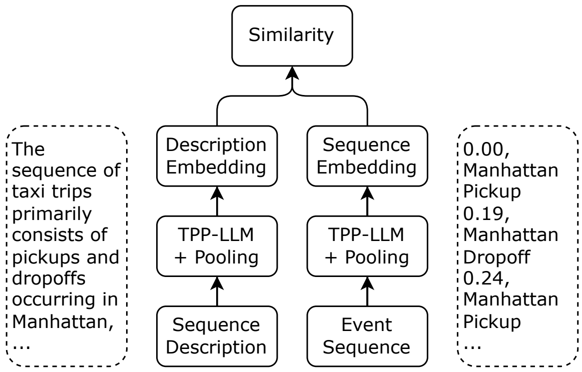


Figure 1: TPP-Embedding framework with one TESRBench example, where the model embeds both textual descriptions and temporal event sequences using a shared TPP-LLM framework, applies pooling to generate fixed-length representations, and uses contrastive learning with similarity scores to align matching pairs for effective event sequence retrieval.

To address these challenges, we introduce TESRBench[1], a comprehensive benchmark for evaluating temporal event sequence retrieval (TESR) from textual descriptions. TESRBench comprises diverse real-world event sequence datasets with synthesized and reviewed textual descriptions, offering a strong foundation for benchmarking retrieval models. It highlights the complexities of aligning event sequences with textual descriptions and provides a standardized platform for evaluating model performance, uncovering key challenges, and identifying opportunities for improvement in temporal and contextual modeling.

Building on this benchmark, we propose TPP-Embedding[2], a novel framework for temporal event sequence retrieval that extends the TPP-LLM

*These authors contributed equally to this work.

[1]Benchmark available on https://huggingface.co/tppllm.

[2]GitHub repository available on https://github.com/zefang-liu/TPP-Embedding.

model (Liu and Quan, 2024). TPP-LLM integrates temporal encoding for event times and textual embeddings for event types within a large language model (LLM) backbone to model temporal point processes (TPPs). Extending this framework, TPP-Embedding aligns sequence-level representations of event sequences and their textual descriptions in a shared embedding space. By modeling the interdependencies between events and their temporal context, TPP-Embedding generates richer, contextually informed embeddings optimized for retrieval tasks. Evaluated across TESRBench datasets, TPP-Embedding demonstrates superior performance over text-based baselines and generalizes effectively across different event domains.

In this paper, our key contributions are: (1) Introducing TESRBench, a benchmark for evaluating TESR models with diverse datasets; (2) Proposing TPP-Embedding, which integrates temporal and event-type information for accurate event sequence retrieval from descriptions; and (3) Showcasing the scalability and flexibility of our approach through multi-domain experiments.

2 Related Work

Recent developments in sentence representation models, such as Sentence-BERT (Reimers and Gurevych, 2019), have significantly improved retrieval tasks by enabling efficient semantic similarity searches using transformer-based embeddings (Vaswani et al., 2017). While these models perform well in standard text retrieval tasks (Lin et al., 2022), they struggle with the temporal and event-specific complexities of event sequence data. To address these challenges, temporal point process (TPP) models (Mei and Eisner, 2017; Shchur et al., 2021; Xue et al., 2023) have been adapted for retrieval tasks. NeuroSeqRet (Gupta et al., 2022, 2023) introduces a neural framework for continuous-time event sequence retrieval by leveraging marked TPPs to model temporal dynamics and using a trainable unwarping function, neural relevance models, and hashing techniques to optimize retrieval efficiency. However, despite these advancements, existing models either treat event types as categorical inputs, limiting their ability to capture rich event semantics, or treat entire sequences as text, ignoring their temporal dependencies.

Recently, Liu and Quan (2024) proposed TPP-LLM, a framework that integrates large language models (LLMs) with TPPs to capture event semantics and temporal dynamics for event sequence modeling and prediction. While TPP-LLM focuses on predicting future event types and times using both textual and temporal information, our proposed TPP-Embedding extends this framework to the task of retrieving temporal event sequences from textual descriptions. By introducing a shared embedding space for sequences and descriptions and employing contrastive learning, our model effectively aligns sequence-level representations with natural language descriptions, enabling retrieval while maintaining temporal and semantic dependencies.

3 Benchmark

In this section, we present TESRBench, a comprehensive benchmark designed to evaluate temporal event sequence retrieval (TESR) from textual descriptions. We provide an overview of its key components, including detailed dataset summaries, the methodology for generating event sequence descriptions, and the evaluation process used to assess the quality of these descriptions.

3.1 Dataset Summaries

TESRBench is built on five real-world datasets from diverse domains: Stack Overflow, Chicago Crime, NYC Taxi Trip, U.S. Earthquake, and Amazon Review. Each dataset captures sequences of event-based information within specific time periods but lacks accompanying textual sequence descriptions. To address this, we generate textual descriptions for these event sequences using GPT-4o-mini (Achiam et al., 2023), creating objective summaries that emphasize the order and timing of events while preserving their essential structure. Details of the description generation and evaluation processes are provided in subsequent subsections. Examples of the data from TESRBench are included in Appendix A for further reference.

The datasets in TESRBench span various domains and offer rich opportunities for analysis. Table 1 presents an overview of their key statistics, using the same train/validation/test splits as Liu and Quan (2024), which are detailed in Table 2. The **Stack Overflow** (Stack Exchange, Inc., 2024) dataset tracks non-tag-related badges earned between January 2022 and December 2023, comprising 3,336 sequences across 25 event types. The **Chicago Crime** (Chicago Police Department,

Dataset	Domain	# of Types	# of Events	# of Seq.	Avg. Seq. Length	Time Unit
Stack Overflow	Social Networks	25	187,836	3,336	56.31	Month
Chicago Crime	Urban Dynamics	20	202,333	4,033	50.17	Month
NYC Taxi Trip	Transportation	8	362,374	2,957	122.55	Hour
U.S. Earthquake	Natural Disasters	3	29,521	3,009	9.81	Day
Amazon Review	E-Commerce	18	127,054	2,245	56.59	Week

Table 1: Dataset statistics overview of event sequences in TESRBench. (# = Number.)

Dataset	Seq.	Train	Val.	Test
Stack Overflow	3,336	2,668	334	334
Chicago Crime	4,033	3,226	403	404
NYC Taxi Trip	2,957	2,365	296	296
U.S. Earthquake	3,009	2,407	301	301
Amazon Review	2,245	1,796	224	225

Table 2: Numbers of sequences in train, validation, and test sets of TESRBench datasets.

2024) dataset focuses on the top 20 crime types and blocks with 30-120 incidents during the same time period, yielding 4,033 sequences across 20 crime categories. The **NYC Taxi Trip** (Monroy-Hernandez, 2014) dataset captures trips from May 1-7, 2013, excluding Staten Island, with 2,957 sequences across 8 location categories. The **U.S. Earthquake** (U.S. Geological Survey, 2024) dataset records 3,009 sequences of earthquake events from January 2020 to December 2023, categorized into 3 magnitude levels. Finally, the **Amazon Review** (Ni et al., 2019) dataset comprises 2,245 sequences of 40-200 reviews per user between January and June 2018, spanning 18 categories. Collectively, these datasets establish a robust foundation for evaluating models on diverse temporal event sequence retrieval tasks.

3.2 Description Generation

To create textual descriptions for the event sequences in TESRBench, we employ a structured process using GPT-4o-mini (Achiam et al., 2023). The process begins with crafting a system message, as illustrated in Figure 2, which guides GPT-4o-mini to produce objective summaries that focus on the order and timing of events. The instructions explicitly avoid interpreting behaviors or including specific numbers or timestamps, ensuring consistency and objectivity in the generated summaries. For each dataset, specific prompts are designed to reflect the context of the event sequences, as detailed in Table 3. These prompts present sequences of events with timestamps and event types, formatted to highlight the unique characteristics of each dataset. GPT-4o-mini processes these prompts and generates concise textual descriptions that capture key patterns and trends, providing an accurate summary of how events unfold over time. This approach ensures that the generated descriptions are well-aligned with the underlying temporal and contextual dynamics of the event sequences.

System Message:
You are an expert in summarizing event sequences. Your task is to provide a 2-5 sentence objective summary of the sequence's key patterns and trends without interpreting any behaviors or motivations. Focus on the sequence's order and timing, emphasizing how the events unfold over time. Describe general trends such as whether certain event types occur earlier or later, or if events cluster in certain periods. Avoid including exact numbers or timestamps.

Figure 2: Instructions for generating objective summaries of event sequences, focusing on the order, timing, and general trends without including specific numbers or timestamps.

3.3 Description Evaluation

To evaluate the quality of the generated descriptions for temporal event sequences, we establish a set of assessment criteria and scoring scales. Leveraging LLMs as evaluators (Zheng et al., 2023), we assess the descriptions across five key dimensions: accuracy, coverage, fidelity, clarity, and conciseness. The definitions of these criteria, along with their respective scoring scales, are outlined below:

- **Accuracy**: Does the description correctly represent the sequence of events, focusing on the event types, their order, and timing? (1 = Completely inaccurate, 5 = Completely accurate)
- **Coverage**: Does the description include all significant events and key details of the se-

Dataset	Description
Stack Overflow	Here is a sequence of badges earned by a user on Stack Overflow, with relative timestamps (in months) and badge names. Please provide a summary that describes the timing and order of events: {event_sequence}
Chicago Crime	Here is a sequence of crime incidents reported at a block in Chicago, with relative timestamps (in months) and crime types. Please provide a summary that describes the timing and order of events: {event_sequence}
NYC Taxi Trip	Here is a sequence of taxi trips taken by a driver in New York City, with relative timestamps (in hours) and trip locations. Please provide a summary that describes the timing and order of events: {event_sequence}
U.S. Earthquake	Here is a sequence of earthquake events in the U.S., with relative timestamps (in days) and magnitude categories. Please provide a summary that describes the timing and order of events: {event_sequence}
Amazon Review	Here is a sequence of product reviews submitted by a user on Amazon, with relative timestamps (in weeks) and review categories. Please provide a summary that describes the timing and order of events: {event_sequence}

Table 3: Overview of dataset-specific prompts, describing event sequences from various domains.

quence, without omitting critical information? (1 = Very incomplete, 5 = Fully comprehensive)

- **Fidelity**: To what extent does the description capture and reflect the temporal relationships and patterns (e.g., clustering, trends, or intervals) in the event sequence? (1 = No temporal fidelity, 5 = High temporal fidelity)
- **Clarity**: Is the description easy to understand, with clear language and a logical structure that aids comprehension? (1 = Very unclear, 5 = Very clear)
- **Conciseness**: Does the description provide the necessary information in a succinct manner, avoiding unnecessary verbosity or redundancy? (1 = Overly verbose or incomplete, 5 = Very concise and complete)

The averaged evaluation scores across datasets are presented in Tables 4, 5, and 6, which report the evaluation of event sequence descriptions using three evaluators: GPT-4o, GPT-4o-mini, and Claude 3.5 Haiku. GPT-4o's evaluation scores highlight strong performance, particularly in clarity and conciseness, while showing slightly lower scores in accuracy, coverage, and fidelity compared to GPT-4o-mini's evaluation. GPT-4o-mini assigns consistently high scores across all dimensions, indicating a strong alignment with the generated descriptions. Meanwhile, Claude 3.5 Haiku presents a different evaluation pattern, demonstrating relatively strong clarity and fidelity scores but notably lower coverage ratings. The varying assessments from these evaluators provide complementary perspectives on the quality of the descriptions, reinforcing their effectiveness in summarizing event sequences while preserving key temporal and contextual relationships. These results further emphasize the robustness of the generated descriptions when assessed across different evaluation frameworks.

Dataset	Acc.	Cov.	Fid.	Cla.	Con.
StackOverflow	4.10	4.05	4.25	4.94	4.56
Crime	4.01	4.00	4.18	4.98	4.67
Taxi	4.44	4.03	4.46	4.89	4.36
Earthquake	4.36	4.31	4.42	4.96	4.95
Amazon	4.66	4.33	4.74	4.99	4.82

Table 4: Evaluation scores from GPT-4o for event sequence descriptions in TESRBench. (Acc. = Accuracy, Cov. = Coverage, Fid. = Fidelity, Cla. = Clarity, Con. = Conciseness.)

4 Methodology

In this section, we introduce TPP-Embedding, an extension of TPP-LLM (Liu and Quan, 2024), de-

Dataset	Acc.	Cov.	Fid.	Cla.	Con.
StackOverflow	5.00	5.00	5.00	5.00	5.00
Crime	5.00	4.87	4.96	5.00	4.86
Taxi	5.00	4.99	4.99	5.00	4.99
Earthquake	4.99	4.99	4.85	5.00	4.93
Amazon	5.00	5.00	4.98	5.00	5.00

Table 5: Evaluation scores from GPT-4o-mini for event sequence descriptions in TESRBench. (Acc. = Accuracy, Cov. = Coverage, Fid. = Fidelity, Cla. = Clarity, Con. = Conciseness.)

Dataset	Acc.	Cov.	Fid.	Cla.	Con.
StackOverflow	4.00	3.27	4.11	4.93	4.00
Crime	4.00	3.05	3.96	4.93	4.00
Taxi	4.00	3.00	3.84	4.84	4.00
Earthquake	4.11	3.32	4.22	4.95	4.00
Amazon	4.18	3.70	4.48	4.80	3.98

Table 6: Evaluation scores from Claude 3.5 Haiku for event sequence descriptions in TESRBench. (Acc. = Accuracy, Cov. = Coverage, Fid. = Fidelity, Cla. = Clarity, Con. = Conciseness.)

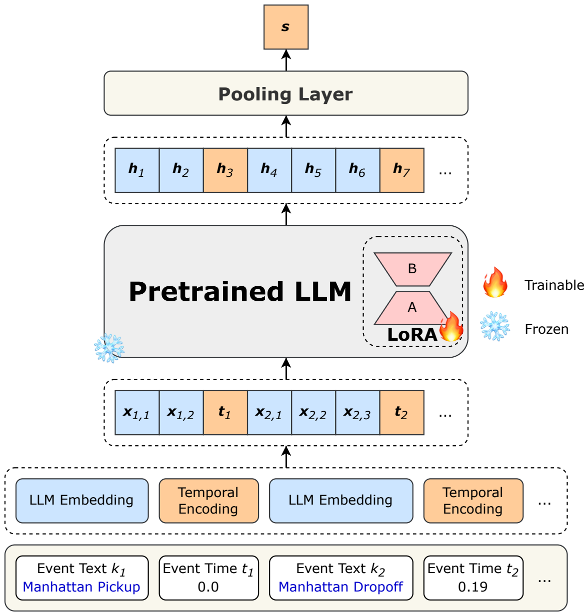


Figure 3: TPP-Embedding architecture, illustrating the embedding process for a event sequence through the integration of temporal and text representations, followed by processing with a large language model and a pooling layer to generate a fixed-length sequence representation.

signed to embed both event sequences and textual descriptions into a shared embedding space, enabling effective retrieval based on similarity.

4.1 Model Architecture

Given a set of textual descriptions $\mathcal{D} = \{d_1, d_2, \ldots, d_m\}$ and a set of temporal event sequences $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$, the task is to retrieve the most relevant sequence $s^* \in \mathcal{S}$ for a given description d_j. Each event sequence s_i consists of a series of events $\{e_{i,1}, e_{i,2}, \ldots, e_{i,n_i}\}$, where each event $e_{i,j}$ is represented by an event time $t_{i,j}$ and an event type $k_{i,j}$. Thus, the sequence can be written as $s_i = \{(t_{i,1}, k_{i,1}), (t_{i,2}, k_{i,2}), \ldots, (t_{i,n_i}, k_{i,n_i})\}$. The goal is to embed both descriptions d_j and event sequences s_i into a shared embedding space for effective retrieval.

Embedding Event Sequences. As illustrated by Figure 3, TPP-Embedding builds upon TPP-LLM (Liu and Quan, 2024) by embedding event sequences through the integration of temporal and event-type representations. For each event $e_{i,j}$, the temporal embedding is computed as $\boldsymbol{t}_{i,j} = f_t(t_{i,j})$, where f_t is a temporal encoding function (Zhang et al., 2020; Zuo et al., 2020). Each event type text $k_{i,j}$ is tokenized by the large language model (LLM) tokenizer and embedded using its embedding layer, resulting in $\boldsymbol{X}_{i,j} = [\boldsymbol{x}_{i,j,1}, \boldsymbol{x}_{i,j,2}, \ldots, \boldsymbol{x}_{i,j,n_j}]$. The temporal and type embeddings are concatenated to form the final event representation $\boldsymbol{E}_{i,j}$. These event embeddings are then passed through the LLM to obtain hidden states $\boldsymbol{H}_i = [\boldsymbol{h}_{i,1}, \boldsymbol{h}_{i,2}, \ldots, \boldsymbol{h}_{i,l_i}] = \mathrm{LLM}([\boldsymbol{E}_{i,1}, \boldsymbol{E}_{i,2}, \ldots, \boldsymbol{E}_{i,n_i}])$. Finally, a pooling operation (Reimers and Gurevych, 2019) is applied to produce a fixed-length representation of the sequence: $\boldsymbol{s}_i = \mathrm{Pool}(\boldsymbol{H}_i)$.

Embedding Descriptions. Textual descriptions d_j are embedded using the same LLM and tokenizer as the event sequences. The description is tokenized and passed through the LLM, resulting in hidden states. A pooling operation is then applied to obtain the final description embedding: $\boldsymbol{d}_j = \mathrm{Pool}(\mathrm{LLM}(d_j))$. By embedding descriptions and sequences in the same space, TPP-Embedding enables retrieval based on their similarity.

4.2 Training Objective

To align the embeddings of descriptions and their corresponding event sequences, we employ a contrastive learning framework. Positive pairs (d_i, s_i) consist of a description and its matching event sequence, while other sequences in the batch serve as negatives. The cosine similarity between description and sequence embeddings is computed as $\mathrm{sim}(d_i, s_j) = \frac{\boldsymbol{d}_i \cdot \boldsymbol{s}_j}{\|\boldsymbol{d}_i\| \|\boldsymbol{s}_j\|}$. The training objective uses a multiple negatives ranking loss (Henderson et al.,

2017) to maximize similarity for positive pairs and minimize it for negative pairs. The loss function is given by:

$$\mathcal{L} = -\log \frac{\exp(\mathrm{sim}(d_i, s_i))}{\sum_j \exp(\mathrm{sim}(d_i, s_j))}. \quad (1)$$

This encourages the model to rank the correct event sequence higher than incorrect ones for each description. To improve efficiency, we apply 4-bit precision quantization (Dettmers et al., 2024) to reduce memory usage and use low-rank adaptation (LoRA) (Hu et al., 2021) to fine-tune a small subset of parameters while keeping the rest frozen. These enhancements allow for efficient fine-tuning and deployment without compromising retrieval performance.

5 Experiments

In this section, we present a detailed overview of the baseline models used for comparison, the evaluation metrics employed, the experimental setup, the results obtained, and the ablation studies conducted.

5.1 Baselines

To enable evaluation with common embedding models, we transform temporal event sequences into a textual format by concatenating events within a sequence. Each event is represented by its relative timestamp followed by the corresponding event type text, separated by a comma. These events are concatenated with line breaks, resulting in a single textual representation for each event sequence. This approach ensures that the temporal and semantic information is preserved for text-based embeddings.

We compare TPP-Embedding against several widely used embedding models: All-MiniLM-L12-v2 (Wang et al., 2020), All-MPNet-Base-v2 (Song et al., 2020), BGE-Large-En-v1.5 (Xiao et al., 2023), MxbAI-Embed-Large-v1 (Li and Li, 2023; Lee et al., 2024), Multilingual-E5-Large-Instruct (Wang et al., 2024), and GTE-Qwen2-1.5B-instruct (Li et al., 2023). These models are designed for generating sentence embeddings and are adapted here for retrieving the most relevant event sequences based on descriptions.

To ensure a fair comparison, all baseline models are fine-tuned using a contrastive learning framework. Specifically, we employ the multiple negatives ranking loss (Henderson et al., 2017), which treats a description and its corresponding event sequence as a positive pair, while all other mismatched pairs within the batch are considered negatives. This fine-tuning process aligns the embeddings of matching descriptions and sequences while separating non-matching ones. In addition, Table 7 provides an overview of the total parameters and trainable parameters for each baseline model. While the baseline models (besides Qwen2-1.5B) require fine-tuning all parameters, TPP-Embedding models and Qwen2-1.5B leverage LoRA for efficient fine-tuning.

Model	Parameters	Trainable
MiniLM-L12	33.4M	33.4M
MPNet-Base	109M	109M
BGE-Large	335M	335M
MxbAI-Large	335M	335M
mE5-Large	560M	560M
Qwen2-1.5B	1.5B	4.4M
TPP-Llama	1.1B	4.5M
TPP-Llama-Chat	1.1B	4.5M

Table 7: Numbers of total and trainable model parameters. (M = Million, B = Billion.)

5.2 Evaluation Metrics

The temporal event sequence and description matching task is framed as a retrieval problem, where the model retrieves the correct event sequence for each description by ranking all event sequences based on their similarity to the description embeddings. We evaluate retrieval quality using two metrics: Mean Reciprocal Rank (MRR) and Recall@K. MRR measures the ranking position of the correct sequence, providing an average of reciprocal ranks across all queries, while Recall@K calculates the proportion of cases where the correct sequence is included in the top K results.

5.3 Experimental Setups

For the baseline models (besides Qwen2-1.5B), we use the AdamW optimizer (Loshchilov and Hutter, 2017), training for 15 epochs with a learning rate of 2e-5, a cosine scheduler, a warmup ratio of 0.1, and a batch size of 8. Qwen2-1.5B uses the same LoRA and training settings as the TPP-Embedding models described below.

TPP-Embedding integrates temporal positional encoding for event times (Zuo et al., 2020), with event type embeddings placed before the temporal embedding (Liu and Quan, 2024). Two foundation models are employed: TinyLlama-1.1B-

Intermediate-Step-1431k-3T (TPP-Llama) and TinyLlama-1.1B-Chat-v1.0 (Zhang et al., 2024) (TPP-Llama-Chat). We utilize all hidden states with mean pooling (Reimers and Gurevych, 2019) and apply 4-bit quantization (Dettmers et al., 2024). LoRA (Hu et al., 2021) is used with a rank of 16 and dropout of 0.05, targeting the attention projection matrices. The model is trained for 25 epochs with a learning rate of 4e-4, a cosine scheduler, a warmup ratio of 0.02, and a batch size of 8. All experiments are conducted five times, with average results reported. The experiments were run on NVIDIA A100 and H100 GPUs. Additional experimental setups are provided in Appendix B.

5.4 Experimental Results

The experimental results demonstrate the effectiveness of our proposed models compared to traditional text-based embedding models. As shown in Table 8, along with Figures 4 and 5, TPP-Llama and TPP-Llama-Chat consistently outperform the baselines across most datasets in terms of both MRR and Recall@5. TPP-Llama achieves the highest MRR and Recall@5 on Stack Overflow and remains competitive across the benchmark except for Amazon Review, while TPP-Llama-Chat attains the best Recall@5 on U.S. Earthquake and leads on Chicago Crime and NYC Taxi Trip in both metrics. While Qwen2-1.5B demonstrates strong performance on U.S. Earthquake and MPNet-Base achieves the highest MRR on Amazon Review, the TPP-based models exhibit superior generalization across the majority of datasets. These results highlight the advantage of the temporal and event-type-aware design of TPP-Embedding, which effectively captures the structure and dependencies within event sequences compared to traditional models.

5.5 Multi-Domain Results

In real-world applications, it is often necessary to retrieve event sequences that span different domains, requiring models to handle various event sequence types. Multi-domain retrieval refers to a model's ability to effectively process and retrieve information across diverse datasets or domains simultaneously, rather than being specialized for a single domain. To simulate such settings, we created a multi-domain dataset by combining 30% of the data from the five datasets. As shown in Table 9, Qwen2-1.5B achieves the highest MRR, while TPP-Llama-Chat attains the best Recall@5. Although Qwen2-1.5B performs strongly, TPP-Llama and TPP-Llama-Chat achieve competitive retrieval effectiveness, particularly excelling in Recall@5, which is crucial for practical multi-domain retrieval scenarios. These results highlight the robustness of TPP-Embedding in retrieving diverse event sequences and its ability to generalize effectively across multiple domains, making it a strong choice for real-world applications.

5.6 Ablation Studies

In this subsection, we perform ablation studies to evaluate the effects of various model configurations on event sequence retrieval performance.

5.6.1 Embedding Inclusions

We conduct an ablation study to assess the impact of using only temporal tokens or only type (textual) tokens on retrieval performance. As shown in Table 10, using only textual tokens achieves performance comparable to using all tokens on the Stack Overflow dataset. However, this approach leads to a significant performance drop on the U.S. Earthquake dataset, likely due to the nature of the datasets: Stack Overflow includes 25 event types, allowing the model to rely primarily on textual contents, whereas the U.S. Earthquake dataset contains only 3 event types, making temporal information essential for accurate retrieval.

5.6.2 Hidden State Selections

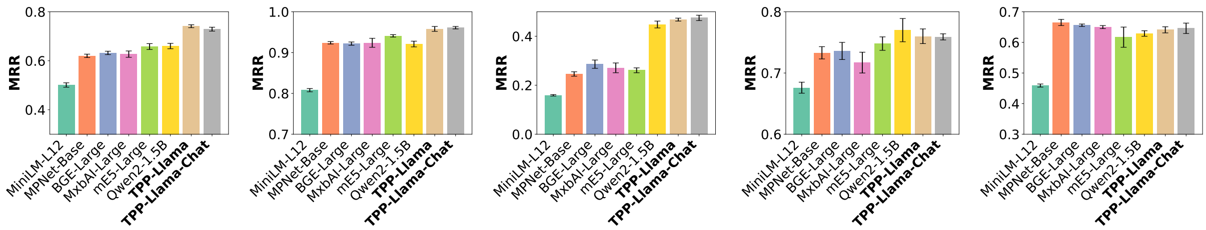
We evaluate the impact of different hidden state selections from the last hidden layer of the model for event sequences, specifically choosing only temporal tokens, a combination of temporal tokens and the last token of event type text tokens for each event, or all tokens. As shown in Table 11, using all tokens generally provides strong results, achieving the highest MRR on the StackOverflow dataset and the highest Recall@5 on the Earthquake dataset. While selecting temporal tokens and the last type tokens slightly improves MRR on the Earthquake dataset, using only temporal tokens lags behind both strategies on both datasets. Overall, choosing all tokens yields consistently good performance.

5.6.3 Pooling Modes

In experiments with different pooling modes as Table 12, we observe that the mean pooling method consistently performs well, achieving the highest MRR and Recall@5 on the StackOverflow dataset. However, for the Earthquake dataset, last token pooling (Muennighoff, 2022) slightly outperforms mean pooling. Max pooling shows competitive

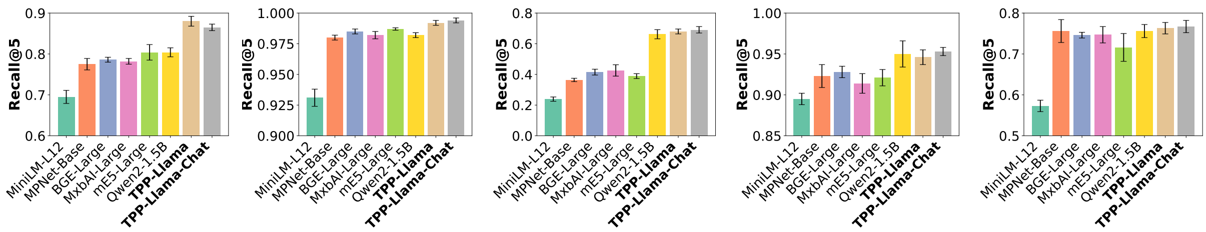
Model	StackOverflow	Crime	Taxi	Earthquake	Amazon
MiniLM-L12	0.501 / 0.695	0.808 / 0.931	0.159 / 0.239	0.676 / 0.895	0.459 / 0.573
MPNet-Base	0.620 / 0.775	0.924 / 0.980	0.246 / 0.364	0.733 / 0.923	**0.665** / 0.756
BGE-Large	0.632 / 0.786	0.922 / 0.985	0.286 / 0.415	0.736 / 0.928	0.656 / 0.746
MxbAI-Large	0.627 / 0.782	0.924 / 0.982	0.271 / 0.426	0.717 / 0.914	0.650 / 0.747
mE5-Large	0.658 / 0.804	0.941 / 0.987	0.261 / 0.389	0.748 / 0.921	0.617 / 0.716
Qwen2-1.5B	0.660 / 0.804	0.921 / 0.982	0.448 / 0.662	**0.770** / 0.950	0.629 / 0.756
TPP-Llama	**0.741 / 0.880**	0.958 / 0.992	0.468 / 0.680	0.760 / 0.946	0.641 / 0.763
TPP-Llama-Chat	0.729 / 0.865	**0.961 / 0.994**	**0.475 / 0.691**	0.759 / **0.953**	0.646 / **0.767**

Table 8: Comparison of average MRR and Recall@5 across TESRBench datasets in event sequence retrieval.



(a) Stack Overflow (b) Chicago Crime (c) NYC Taxi Trip (d) U.S. Earthquake (e) Amazon Review

Figure 4: Comparison of average MRRs with standard deviations on TESRBench in event sequence retrieval.



(a) Stack Overflow (b) Chicago Crime (c) NYC Taxi Trip (d) U.S. Earthquake (e) Amazon Review

Figure 5: Comparison of average Recall@5 with standard deviations on TESRBench in event sequence retrieval.

Model	MRR	Recall@5
MiniLM-L12	0.634 ± 0.007	0.795 ± 0.009
MPNet-Base	0.748 ± 0.003	0.889 ± 0.007
BGE-Large	0.744 ± 0.010	0.888 ± 0.010
MxbAI-Large	0.744 ± 0.006	0.876 ± 0.010
mE5-Large	0.749 ± 0.013	0.888 ± 0.012
Qwen2-1.5B	$\mathbf{0.783} \pm 0.005$	0.914 ± 0.013
TPP-Llama	0.772 ± 0.009	0.914 ± 0.008
TPP-Llama-Chat	0.770 ± 0.005	$\mathbf{0.919} \pm 0.009$

Table 9: Comparison of average MRRs and Recall@5 with standard deviations on the multi-domain dataset.

Embeddings	StackOverflow	Earthquake
Temporal Tokens	0.037 / 0.040	0.179 / 0.281
Textual Tokens	0.726 / **0.870**	0.675 / 0.890
All Tokens	**0.729** / 0.865	**0.759 / 0.953**

Table 10: Comparison of average MRRs and Recall@5 of TPP-Llama-Chat with different embedding inclusions.

Hidden States	StackOverflow	Earthquake
Temporal Tokens	0.718 / 0.862	0.754 / 0.939
+ Last Type Tokens	0.727 / **0.875**	**0.766 / 0.953**
All Tokens	**0.729** / 0.865	0.759 / **0.953**

Table 11: Comparison of average MRRs and Recall@5 of TPP-Llama-Chat with different hidden state selections.

performance on the StackOverflow dataset but performs considerably worse on the Earthquake dataset. Overall, mean pooling offers a balanced performance, making it a reliable choice.

Pooling	StackOverflow	Earthquake
Mean	**0.729 / 0.865**	0.759 / 0.953
Max	0.712 / 0.857	0.627 / 0.853
Last Token	0.728 / 0.848	**0.772 / 0.960**

Table 12: Comparison of average MRRs and Recall@5 of TPP-Llama-Chat with different pooling modes.

5.6.4 Loss Functions

To examine the impact of the loss function on retrieval performance, we replace the contrastive loss with a Mean Squared Error (MSE) loss, which optimizes cosine similarity to 1 for matched pairs. As shown in Table 13, this substitution leads to a pronounced decline in both metrics across all datasets, emphasizing the pivotal role of contrastive loss in capturing subtle relationships between closely related event sequences. These results highlight the effectiveness of contrastive learning in enhancing retrieval accuracy.

Loss	StackOverflow	Earthquake
MSE	0.020 / 0.016	0.020 / 0.015
Contrastive	**0.729 / 0.865**	**0.759 / 0.953**

Table 13: Comparison of average MRRs and Recall@5 of TPP-Llama-Chat with different loss functions.

6 Conclusion

In this paper, we introduce TESRBench, a comprehensive benchmark for evaluating temporal event sequence retrieval, alongside TPP-Embedding, a novel model designed to integrate temporal and event-type-aware representations. TESRBench provides a diverse set of datasets with synthesized textual descriptions, offering a robust foundation for benchmarking models in this domain. Our proposed TPP-Embedding model combines temporal encoding and event text embedding with a large language model backbone, enabling it to effectively capture the structure and dependencies of temporal event sequences. Extensive experiments conducted on TESRBench demonstrate its superior performance compared to traditional text-based baselines, particularly in handling temporally complex, multi-type event sequences. Furthermore, multi-domain experiments underscore the flexibility and adaptability of our approach across diverse event domains. Together, TESRBench and TPP-Embedding represent a significant step forward in advancing research on temporal event sequence retrieval.

Limitations

TESRBench, while providing a robust foundation for evaluating temporal event sequence retrieval, relies on synthesized textual descriptions generated by GPT-4o-mini, which may not fully capture the variability and complexity of real-world user-generated descriptions. A limitation of our TPP-Embedding model is its reliance on high-quality temporal and event-type data, which could pose challenges when dealing with noisy or incomplete event sequences encountered in real-world scenarios. Furthermore, while TPP-Embedding achieves strong retrieval performance, its dependence on large-scale language models can introduce computational latency on extremely large datasets, necessitating further optimization strategies. Finally, our current baselines are restricted to text-based methods, and future research could explore integrating recent time-context-aware sequential recommendation techniques (Li et al., 2020; Rashed et al., 2022; Tran et al., 2023; Liu et al., 2024) to further improve the retrieval of temporal event sequences from textual descriptions.

Ethical Considerations

In constructing TESRBench, we acknowledge potential ethical concerns related to the use of synthesized textual descriptions and real-world event data. While the textual descriptions are generated objectively, they may still inadvertently reflect biases or limitations inherent in the data sources. For TPP-Embedding, its ability to retrieve temporal event sequences could be misused in privacy-sensitive applications, such as personal activity tracking. It is crucial to ensure that all data used for training and retrieval is anonymized and managed responsibly. Additionally, biases in training data, such as uneven representation of event types or domains, could result in biased retrieval outcomes. Future work should emphasize dataset curation and the implementation of bias mitigation strategies to minimize potential harms.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Chicago Police Department. 2024. Crimes - 2001 to present.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Vinayak Gupta, Srikanta Bedathur, and Abir De. 2022. Learning temporal point processes for efficient retrieval of continuous time event sequences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4005–4013.

Vinayak Gupta, Srikanta Bedathur, and Abir De. 2023. Retrieving continuous time event sequences using neural temporal point processes with learnable hashing. *ACM Transactions on Intelligent Systems and Technology*.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Abhinav Ramesh Kashyap, Thanh-Tung Nguyen, Viktor Schlegel, Stefan Winkler, See Kiong Ng, and Soujanya Poria. 2024. A comprehensive survey of sentence representations: From the bert epoch to the chatgpt era and beyond. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1738–1751.

Sean Lee, Aamir Shakir, Darius Koenig, and Julius Lipp. 2024. Open source strikes bread - new fluffy embeddings model.

Quentin Lhoest, Albert Villanova Del Moral, Yacine Jernite, Abhishek Thakur, Patrick Von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, et al. 2021. Datasets: A community library for natural language processing. *arXiv preprint arXiv:2109.02846*.

Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*, pages 322–330.

Xianming Li and Jing Li. 2023. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2022. *Pretrained transformers for text ranking: Bert and beyond*. Springer Nature.

Yang Liu, Yitong Wang, and Chenyue Feng. 2024. Unirec: A dual enhancement of uniformity and frequency in sequential recommendations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 1483–1492.

Zefang Liu and Yinzhu Quan. 2024. Tpp-llm: Modeling temporal point processes by efficiently fine-tuning large language models. *arXiv preprint arXiv:2410.02062*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and B. Bossan. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods.

Hongyuan Mei and Jason M Eisner. 2017. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30.

Andres Monroy-Hernandez. 2014. NYC taxi trips.

Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Ahmed Rashed, Shereen Elsayed, and Lars Schmidt-Thieme. 2022. Context and attribute-aware sequential recommendation via cross-attention. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 71–80.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Oleksandr Shchur, Ali Caner Türkmen, Tim Januschowski, and Stephan Günnemann. 2021. Neural temporal point processes: A review. *arXiv preprint arXiv:2104.03528*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

Stack Exchange, Inc. 2024. Stack exchange data dump.

Viet Anh Tran, Guillaume Salha-Galvan, Bruno Sguerra, and Romain Hennequin. 2023. Attention mixtures for time-aware sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1821–1826.

U.S. Geological Survey. 2024. USGS earthquake catalog.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2023. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*.

Siqiao Xue, Xiaoming Shi, Zhixuan Chu, Yan Wang, Fan Zhou, Hongyan Hao, Caigao Jiang, Chen Pan, Yi Xu, James Y Zhang, et al. 2023. Easytpp: Towards open benchmarking the temporal point processes. *arXiv preprint arXiv:2307.08097*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. 2020. Self-attentive hawkes process. In *International conference on machine learning*, pages 11183–11193. PMLR.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. 2020. Transformer hawkes process. In *International conference on machine learning*, pages 11692–11702. PMLR.

A Data Examples

This appendix presents selected examples of event sequences from the validation sets in TESRBench, along with their corresponding descriptions, as shown in Table 14. These descriptions highlight key temporal patterns and provide context for the diversity of events and their occurrences across the benchmark's datasets.

B More Experimental Setup

Our experiments were conducted using several key Python libraries, including `pytorch` (Paszke et al., 2019) for deep learning, `transformers` (Wolf et al., 2020) for working with pre-trained language models, `sentence-transformers` (Reimers and Gurevych, 2019) for embedding and retrieval tasks, `datasets` (Lhoest et al., 2021) for data handling, `peft` (Mangrulkar et al., 2022) for parameter-efficient fine-tuning, and `bitsandbytes` (Dettmers et al., 2022) for model quantization.

C More Experimental Results

In this appendix, we provide additional experimental results to further analyze the performance of our models, both before and after fine-tuning.

C.1 Experimental Results before Fine-Tuning

The performance of all models before fine-tuning in Table 15 shows significantly lower retrieval accuracy, with the highest MRR achieved by BGE-Large across most datasets. TPP-based models perform poorly at this stage, due to unaligned temporal and textual spaces, indicating the necessity of fine-tuning to improve event sequence retrieval.

C.2 More Experimental Results after Fine-Tuning

The results after fine-tuning demonstrate a significant improvement across all models, as shown in Tables 16, 17, and 18. TPP-Llama and TPP-Llama-Chat consistently outperform baseline models in MRR, Recall@1, and Recall@5 across most datasets. These improvements highlight the importance of fine-tuning in enhancing their retrieval capabilities.

Dataset	Event Sequence	Description
Stack Overflow	0.00,Nice Question 0.57,Good Answer 0.66,Popular Question 0.83,Famous Question 0.89,Nice Question 2.25,Popular Question 2.32,Necromancer 2.43,Notable Question 3.00,Great Question 3.40,Good Question 3.64,Popular Question 3.69,Notable Question 4.00,Notable Question ... 23.41,Popular Question	The sequence of badges earned by the user on Stack Overflow shows a progression of achievements over time, with certain types of badges appearing more frequently at different intervals. Initially, badges such as "Nice Question" and "Good Answer" are earned in the first few months, followed by a cluster of "Popular Question" and "Notable Question" badges around the two to four-month mark. As time progresses, there is a notable increase in the frequency of "Famous Question" and "Popular Question" badges, particularly in the later months, indicating a trend toward these types of achievements. The sequence also includes several clusters of "Nice Question" badges, particularly around the seven-month mark and again towards the end of the timeline. Overall, the user's badge accumulation reflects a gradual increase in the variety and frequency of achievements over the course of the observed period.
Chicago Crime	0.00,Battery 0.26,Battery 0.52,Theft 0.69,Motor Vehicle Theft 0.74,Assault 0.99,Motor Vehicle Theft 0.99,Motor Vehicle Theft 1.08,Criminal Sexual Assault ... 23.72,Deceptive Practice	The sequence of crime incidents shows a notable clustering of certain crime types over time, particularly motor vehicle thefts, which appear frequently throughout the timeline, especially in the earlier months. Battery incidents are also prevalent, occurring multiple times in the first half of the sequence. Other offenses such as robbery and criminal damage emerge at various intervals, with some clustering in the middle to later months. Overall, there is a trend of increasing diversity in crime types as the timeline progresses, with a gradual rise in the frequency of theft-related incidents towards the end.
NYC Taxi Trip	0.00,Manhattan Pickup 0.19,Manhattan Dropoff 0.24,Manhattan Pickup 0.68,Manhattan Dropoff 0.73,Manhattan Pickup 0.99,Manhattan Dropoff 1.13,Manhattan Pickup 1.43,Manhattan Dropoff 1.45,Manhattan Pickup 1.54,Manhattan Dropoff ... 31.87,Brooklyn Dropoff	The sequence of taxi trips primarily consists of pickups and dropoffs occurring in Manhattan, with a notable concentration of events in the first few hours. Early in the sequence, the driver consistently alternates between pickups and dropoffs, with a high frequency of trips. As the sequence progresses, there are brief periods where trips shift to Queens and Brooklyn, particularly after a long duration of Manhattan trips. The latter part of the sequence shows a gradual transition to more pickups and dropoffs in Brooklyn, indicating a shift in location focus. Overall, the events are clustered closely together in time, with significant activity in the first half of the sequence before expanding to other boroughs.
U.S. Earthquake	0.00,Medium 0.66,Large 0.72,Large 0.99,Large 1.07,Large 1.08,Large 1.67,Large	The sequence of earthquake events begins with a medium magnitude event, followed closely by a series of large magnitude events occurring within a short time frame. The large events cluster together, with multiple occurrences happening within the first two days. This indicates a trend of increasing magnitude shortly after the initial medium event, with the majority of the large events occurring in rapid succession.
Amazon Review	0.00,Books 0.14,Sports and Outdoors 0.14,Books 0.29,Books 0.43,Books 0.57,Books 1.00,Books 1.14,Books ... 25.29,Books	The sequence of product reviews shows a predominant focus on the "Books" category, which appears consistently throughout the timeline, especially in the initial weeks. Other categories such as "Pet Supplies" and "Grocery and Gourmet Food" emerge intermittently, often clustering around specific weeks, particularly in the middle and later parts of the sequence. "Clothing Shoes and Jewelry" and "Movies and TV" also appear, but less frequently, with some clustering noted in the later weeks. Overall, there is a clear trend of sustained interest in "Books," with other categories appearing in a more sporadic manner.

Table 14: Event sequence examples with their descriptions from the validation sets of TESRBench.

Model (before FT)	StackOverflow	Crime	Taxi	Earthquake	Amazon	Multi-Domain
MiniLM-L12	0.091 / 0.123	0.071 / 0.111	0.028 / 0.024	0.037 / 0.043	0.142 / 0.200	0.154 / 0.208
MPNet-Base	0.068 / 0.087	0.027 / 0.020	0.022 / 0.017	0.031 / 0.027	0.068 / 0.071	0.102 / 0.127
BGE-Large	0.122 / 0.162	0.126 / 0.158	0.042 / 0.051	0.039 / 0.040	0.215 / 0.293	0.196 / 0.247
MxbAI-Large	0.085 / 0.102	0.091 / 0.134	0.039 / 0.037	0.038 / 0.043	0.174 / 0.227	0.170 / 0.221
mE5-Large	0.065 / 0.078	0.078 / 0.087	0.028 / 0.024	0.037 / 0.040	0.142 / 0.187	0.145 / 0.191
Qwen2-1.5B	0.047 / 0.054	0.032 / 0.032	0.025 / 0.020	0.027 / 0.027	0.109 / 0.116	0.095 / 0.114
TPP-Llama	0.022 / 0.021	0.019 / 0.020	0.020 / 0.014	0.022 / 0.020	0.033 / 0.027	0.025 / 0.030
TPP-Llama-Chat	0.020 / 0.015	0.018 / 0.012	0.019 / 0.014	0.023 / 0.020	0.033 / 0.031	0.021 / 0.017

Table 15: Comparison of MRRs and Recall@5 on TESRBench in event sequence retrieval before fine-tuning.

Model	StackOverflow	Crime	Taxi	Earthquake	Amazon
MiniLM-L12	0.501 ± 0.009	0.808 ± 0.004	0.159 ± 0.003	0.676 ± 0.009	0.459 ± 0.005
MPNet-Base	0.620 ± 0.007	0.924 ± 0.003	0.246 ± 0.009	0.733 ± 0.010	$\mathbf{0.665} \pm 0.010$
BGE-Large	0.632 ± 0.007	0.922 ± 0.004	0.286 ± 0.017	0.736 ± 0.014	$\underline{0.656} \pm 0.004$
MxbAI-Large	0.627 ± 0.013	0.924 ± 0.011	0.271 ± 0.020	0.717 ± 0.017	0.650 ± 0.005
mE5-Large	0.658 ± 0.012	0.941 ± 0.003	0.261 ± 0.010	0.748 ± 0.011	0.617 ± 0.033
Qwen2-1.5B	0.660 ± 0.011	0.921 ± 0.007	0.448 ± 0.014	$\underline{0.770} \pm 0.019$	0.629 ± 0.009
TPP-Llama	$\mathbf{0.741} \pm 0.006$	$\underline{0.958} \pm 0.006$	$\underline{0.468} \pm 0.006$	$\mathbf{0.760} \pm 0.012$	0.641 ± 0.010
TPP-Llama-Chat	$\underline{0.729} \pm 0.008$	$\mathbf{0.961} \pm 0.003$	$\mathbf{0.475} \pm 0.011$	0.759 ± 0.005	0.646 ± 0.017

Table 16: Comparison of average MRRs with standard deviations on TESRBench in event sequence retrieval.

Model	StackOverflow	Crime	Taxi	Earthquake	Amazon
MiniLM-L12	0.353 ± 0.006	0.711 ± 0.005	0.063 ± 0.007	0.513 ± 0.013	0.348 ± 0.007
MPNet-Base	0.497 ± 0.009	0.878 ± 0.006	0.123 ± 0.013	0.598 ± 0.015	$\mathbf{0.579} \pm 0.012$
BGE-Large	0.509 ± 0.011	0.875 ± 0.006	0.155 ± 0.023	0.595 ± 0.026	$\underline{0.569} \pm 0.011$
MxbAI-Large	0.502 ± 0.017	0.879 ± 0.019	0.130 ± 0.023	0.573 ± 0.019	0.564 ± 0.010
mE5-Large	0.540 ± 0.016	0.904 ± 0.006	0.132 ± 0.006	0.612 ± 0.013	0.520 ± 0.034
Qwen2-1.5B	0.541 ± 0.018	0.872 ± 0.012	0.284 ± 0.012	$\mathbf{0.638} \pm 0.031$	0.523 ± 0.009
TPP-Llama	$\mathbf{0.637} \pm 0.010$	$\underline{0.930} \pm 0.011$	$\underline{0.301} \pm 0.010$	$\underline{0.622} \pm 0.021$	0.538 ± 0.015
TPP-Llama-Chat	$\underline{0.620} \pm 0.012$	$\mathbf{0.936} \pm 0.003$	$\mathbf{0.305} \pm 0.014$	0.619 ± 0.013	0.546 ± 0.021

Table 17: Comparison of average Recall@1 with standard deviations on TESRBench in event sequence retrieval.

Model	StackOverflow	Crime	Taxi	Earthquake	Amazon
MiniLM-L12	0.695 ± 0.016	0.931 ± 0.007	0.239 ± 0.014	0.895 ± 0.007	0.573 ± 0.014
MPNet-Base	0.775 ± 0.014	0.980 ± 0.002	0.364 ± 0.011	0.923 ± 0.014	0.756 ± 0.028
BGE-Large	0.786 ± 0.006	0.985 ± 0.002	0.415 ± 0.019	0.928 ± 0.007	0.746 ± 0.007
MxbAI-Large	0.782 ± 0.007	0.982 ± 0.003	0.426 ± 0.037	0.914 ± 0.012	0.747 ± 0.020
mE5-Large	0.804 ± 0.019	0.987 ± 0.001	0.389 ± 0.016	0.921 ± 0.010	0.716 ± 0.034
Qwen2-1.5B	0.804 ± 0.011	0.982 ± 0.002	0.662 ± 0.030	$\underline{0.950} \pm 0.016$	0.756 ± 0.016
TPP-Llama	$\mathbf{0.880} \pm 0.012$	$\underline{0.992} \pm 0.002$	$\underline{0.680} \pm 0.016$	0.946 ± 0.009	$\underline{0.763} \pm 0.014$
TPP-Llama-Chat	$\underline{0.865} \pm 0.008$	$\mathbf{0.994} \pm 0.002$	$\mathbf{0.691} \pm 0.021$	$\mathbf{0.953} \pm 0.005$	$\mathbf{0.767} \pm 0.015$

Table 18: Comparison of average Recall@5 with standard deviations on TESRBench in event sequence retrieval.