ForestCast: Open-Ended Event Forecasting Based on Semantic News Forest

Anonymous ACL submission

Abstract

Open-ended Event Forecasting (OEEF) is vital in various real-world applications. However, it faces challenges, including limited availability of datasets that enhance LLM's predictive capabilities and crude methods of organizing forecast-related information. In this work, we construct a large-scale dataset **NewsForest** that contains 12,406 prediction chains reflecting the drivers of event development. To effectively extract information from the prediction background, we propose a prediction method, **ForestCast**. ForestCast organizes all relevant news into a story tree and predicts each branch based on the story tree. ForestCast has five main steps: (1) collecting and cleaning news, (2) clustering news into event nodes, (3) constructing the news story tree, (4) mining the semantic structure of the news story tree, (5) predicting the next node and evaluating the quality of the predictions. Experiments demonstrate that the NewsForest dataset enhances the model's ability to predict these structures. The ForestCast method improves the accuracy and quality of predictions.

1 Introduction

Event prediction shows great application potential in various fields such as policy making, risk management, and financial markets (Zhao, 2022). By analyzing historical trends and current dynamics, accurately mining evolutionary structures of events and predicting future events can effectively help decision-makers anticipate and respond to possible challenges and opportunities (Zhao, 2022).

Traditional forecasting tasks can be categorized into script event prediction (Chambers and Jurafsky, 2008) and temporal knowledge graph completion (TKGC) (Leblay and Chekol, 2018). These tasks are limited to predicting specific attributes and can only select answers from a finite range (Lin et al., 2022; Ma et al., 2024; Shi et al., 2023; Xu et al., 2023). Real-world developments are often not limited to a specific scope, and critical information is not always a defined attribute. For example, when predicting the trend of US-China tariffs, a specific tariff rate is not the sole focus of the prediction. It is also difficult to define the scope of the future tariff. Open-ended Event Forecasting (OEEF) is proposed to address these critical and diverse forecasting tasks (Wang et al., 2025).

The OEEF task faces significant challenges. First, existing datasets (Li et al., 2021; Caselli and Vossen, 2017) focus solely on data crawling and attribute extraction, overlooking the developmental logic of events and topics' true meaning. This leads to an inability to improve the prediction ability of LLMs, and even a steep drop in accuracy for open-ended tasks. Second, existing prediction methods deal with the background information of the prediction by simple clustering and summarization (Guan et al., 2024; Ma et al., 2024; Wang et al., 2025), ignoring the complex relationships between events in the background information.

To address the above challenges, this paper proposes a dataset **NewsForest** and a prediction method **ForestCast**. The NewsForest dataset selects topics that reflect the drivers of event development and uncovers the logical progression relationships within these topics. It enables LLMs to learn hidden event development drivers and enhance their forecasting capabilities. The ForestCast method aims to mine the evolutionary structures of events from a prediction background and provides predictions for multiple evolutionary directions of events. We have implemented the ForestCast method online. It can be accessed at http://newsinsight.cn [1] and the associated datasets and code are posted at https://anonymous.4open.science/r/Newsforest.

The main contributions of this research include the following two aspects:

[1] We provide a temporary visitor account for reviewers to use. Account: visitor, password: 1234

- We construct **NewsForest**, a large OEEF dataset that can be used to enhance the event prediction capabilities of LLMs. The dataset contains 12,406 prediction chains covering the four most significant domains.
- We develop **ForestCast**, an OEEF method. This method can organize massive news into a news story tree and use a fine-tuned model to predict the future development of the story tree.

2 Related Work

2.1 News Story Tree Construction

The Topic Detection and Tracking(TDT) (Allan et al., 1998) helps users quickly extract key information from massive news by thematically clustering and continuously tracking news events. However, the TDT task ignores potential dependency relationships between events. Researchers propose various methods to capture the structural features of event evolution. Nallapati et al. (2004) quantifies the dependency between two events based on temporal relationships and TF-IDF vector similarity. Further, Yang et al. (2009) introduces the concept of event graphs to describe the relationship between events.

However, these studies only focus on pairwise event relationships and fail to fully reflect the overall event evolutionary structures. Shahaf et al. (2012) proposes "metro maps" to describe event evolutionary structures. Liu et al. (2018, 2020) proposes a structure more aligned with event development patterns and user cognition—the news story tree, which constructs dependencies between two events through a keyword map of the text. However, these methods are still limited to pruning operations on the graph structure and fail to effectively reflect the internal cohesion within the same branch and the distinctiveness between different branches.

Moreover, existing methods for capturing event evolutionary structures mostly rely on low-level text feature analysis, such as keyword graphs (Liu et al., 2020), keyword reoccurrence rate (Shahaf et al., 2012), and TF-IDF vectors (Nallapati et al., 2004). In contrast, the method proposed in this paper innovatively introduces LLMs and pre-trained sentence encoders, enabling analysis at a higher semantic level.

2.2 Event Prediction

Script event prediction (Chambers and Jurafsky, 2008) requires selecting the most likely subsequent event from a candidate list given an event context. In recent years, several studies have predicted the most probable outcomes by constructing event chain (Wang et al., 2017, 2024), event evolution graph (Ding et al., 2019; Li et al., 2018; Du et al., 2022), or event graph deformation structure (Zhou et al., 2021; Ma et al., 2023; Granroth-Wilding and Clark, 2016) as event evolutionary structures. On the other hand, Temporal Knowledge Graph Completion (TKGC) (Xia et al., 2024; Deng et al., 2020; Rong et al., 2025; Zhang et al., 2024a,b) addresses incomplete temporal knowledge graphs by learning representations of entities, relations, and times to predict missing information.

However, script event prediction and TKGC are limited to predicting specific attributes and can only select results within predefined scopes. When predicting real-world events, these methods offer insufficient guidance. Therefore, Guan et al. (2024) first proposes the task definition of OEEF, characterized by: (1) diverse predictive questions covering different stages of event development and viewpoints, promoting comprehensive analysis; (2) flexible prediction results with no restrictions on scope, format, or length, allowing for semantically complete detailed responses.

Guan et al. (2024) proposes a prediction method. However, this method only clusters and summarises the background news, ignoring the dependency relationship between related events. In the dataset construction of OEEF, Guan et al. (2024) provides a small manual dataset for testing purposes only. Wang et al. (2025) further proposes a large OEEF dataset; however, performance decreases after fine-tuning models with this dataset. We hypothesise that the dataset organizes topics by simply listing decades of history for a place or people, making it difficult to capture the hidden factors driving event development and the underlying logic of events.

3 NewsForest

We construct a large dataset containing 12,406 prediction chains. The dataset can be used directly for model training or evaluation. We use news from 2024-10-01 to 2025-04-20 to avoid the knowledge leakage problem of LLM as much as possible.

3.1 Data Selection and Cleaning

The topics we select are typically event-oriented and strategic, better reflecting the hidden drivers

Header	Content
Instruction	Task: Please predict the next node in the chain based on the background.
Input	**Background**: In New York, a federal judge kept a lawsuit against Trump alive, while Trump's legal team pushed for dismissals. Judges delayed or recused from Trump-related cases. Democrats confirmed judges pre-inauguration, and Republicans tried to stall Trump trials. **Chain**: 1. A federal judge denied throwing out a baseless lawsuit filed against Trump, maintaining its validity in court proceedings. 2. In Colorado, Republican judges dismissed Trump's election interference and secret documents cases, potentially setting Aileen Cannon as a future AG; these rulings could sway the outcome of Trump supporter candidate selection in court appointments. 3. Judges continue to cancel Trump's 2020 election case court deadlines post-presidential win, with a judge shutting down GOP challenge months after Trump's victory. 4. The judge paused Trump-related January 6 cases, citing special counsel's request and concerns about timing after Trump's election win and before his inauguration.
Chosen	After Joe Biden's victory in the US presidential election, a judge paused Donald Trump's legal challenges to the 2020 election results. This decision came shortly after the formal certification of the election outcome. Judge Jack Smith put the federal election interference case against Trump on hold.
Reject	Rioters sought charges dismissal unless Trump was also convicted, but lost in court, and were sentenced to prison following the Capitol riot.

Table 1: Example of masked DPO training data

of event development. We focus on the four most instructive domains: economics, politics, military, and social events (Table 2).

Domain	Examples
Economics	(tariff, China), (stock, market)
Politics	(elect, republican), (trump, China)
Military	(NATO, Ukraine), (America, Iran)
Social Events	(Florida, hurricane), (police, racism)

Table 2: Sample topics from different domains in the NewsForest dataset

We select keyword groups for popular topics from the past six months. We query all articles with keywords in their titles from our GDELT-based (Leetaru and Schrodt, 2013) database. To ensure topic coherence, we semantically encode the article headlines using all-MiniLM-L6-v2. We use the HDBSCAN algorithm (Rahman et al., 2016) to cluster the articles and retain only the largest cluster as the final data source. For each topic, to ensure coverage of both long-term and short-term prediction tasks (Wang et al., 2025), we set news search time lengths of 20, 30, 40, 60, and 80 days. Within the determined time length, we randomly set the period.

3.2 Dataset Main Construction

After determining the keywords and period, we construct the news story tree. Constructing the news story tree involves three steps: clustering news into event nodes, constructing the news story tree, and mining the semantic structure of the news story tree. Since the process of constructing the story tree in this step is identical to the method used in the prediction approach, we provide a detailed explanation of the story tree construction steps in Section 4.2, 4.3, and 4.4.

3.3 Dataset Masking and Specifics

After constructing the news story tree using the ForestCast method, we treat the path from a leaf node to the root node as a news development chain. The root node's branch summary serves as background. We need to mask each news development chain further to enable the LLMs to learn hidden event evolutionary structures from the dataset. For each chain, we mask one or more nodes at the end of the chain. If the chain length is greater than 6, we mask the last third of the nodes in the chain. The masked nodes serve as the correct answers for the chain prediction. Thus, long chains can generate two or more prediction chains. If the chain length is less than 6, only the last node is masked. Additionally, we use the DPO method to train the model, so we select the node with the lowest attachment score as the rejection answer. An example is shown in Table 1. In the end, we get the dataset with 12,406 prediction chains. We also present more information about the dataset in Appendix A.2.1.

3.4 Dataset Quality Assessment

We found a lack of quality assessment for event evolutionary structures in TDT tasks. Therefore, referencing the evaluation methods in Guan et al. (2024), we propose six evaluation metrics, detailed in Table 3. Atomicity evaluates the node itself. Validity and relevance assess the relationship between the current node and its preceding nodes. Chain consistency and causal insight evaluate the entire chain. Branch rationality assesses the whole tree. We provide scoring examples using LLM to rate six attributes on a 1-5 score.

Evaluation results (Figure 2) show average scores from 2.57 to 3.74. Chain consistency and branch rationality score 3.66 and 3.70, indicating good structural integrity. We also present more as-

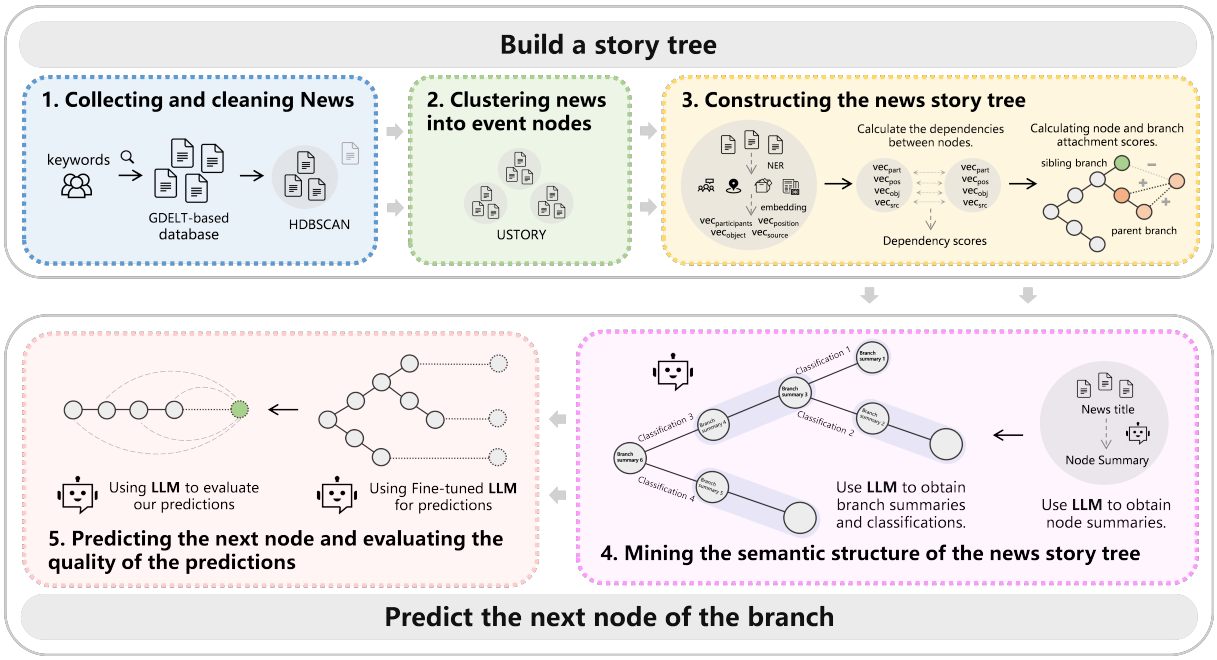


Figure 1: The pipeline of the ForestCast method.

Metric	Description
Atomicity	Is the event description specific?
Validity	Does the event provide new information?
Relevance	Are the main participants consistent consecutively?
Chain Consistency	Is logic consistent throughout the chain?
Causal Insight	Are hidden relationships captured?
Branch Rationality	Are branches distinct/non-mergeable?

Table 3: Explanation of evaluation metrics for news story trees

sessment information about the NewsForest dataset in Appendix A.2.2 and A.2.3.

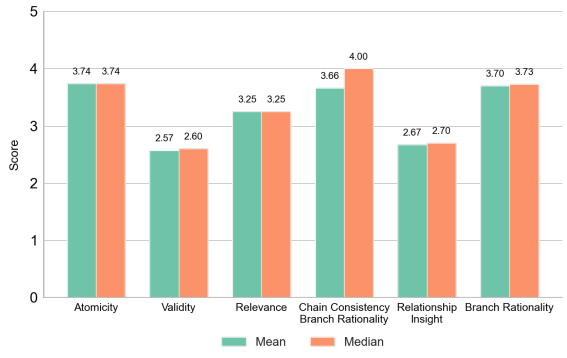


Figure 2: Evaluation results of the NewsForest dataset.

4 ForestCast

Existing work on processing predictive information remains focused on extracting attributes between events and linking them based on those attributes. These works overlook the developmental logic and underlying drivers of the information. We propose a method that further predicts events based on the organization of event evolutionary structures. Our method requires users to provide topic-related keywords. Then we can automatically complete news collection and analysis, returning the story tree and its predictions within 5 minutes. In the Appendix A.1.4, we provide the time complexity of the method and its efficiency in real situations. Figure 3 shows the main interface of the website. We detail other functions in the Appendix A.1.1. As shown in Figure 1, our ForestCast method is divided into five modules: (1) collecting and cleaning news; (2) clustering news into event nodes; (3) constructing the news story tree; (4) mining the semantic structure of the news story tree; (5) predicting the next node and evaluating the quality of the predictions.

4.1 Collecting and Cleaning News

Our data is sourced from GDELT to ensure comprehensive and credible news sources. We obtain news headlines, links, publication times, and media sources from the GDELT project. We then retrieve the full text of the news based on the links. We use keyword searches for news. After obtaining relevant news, we deduplicate the articles.

4.2 Clustering News into Event Nodes

After obtaining news, we need to cluster news articles narrating the same event. These clusters serve as event nodes in the news story tree. Considering clustering speed and accuracy, we use the semantics-based USTORY (Yoon et al., 2023)for clustering.

4.3 Constructing the news story tree

In the current work on the TDT task, the story tree is the visual form that best matches the user's cognitive habits and event evolutionary structures. Therefore, we construct story trees to organize relevant news.

4.3.1 Calculating Dependency Degree Between Two Nodes

Inspired by Yang et al. (2009), we regard the hidden dependency relationship between events is determined by the participants, positions, objects, and media sources. Therefore, we use en_core_web_sm text processing model to extract important terms from articles, deduplicate and disambiguate them to obtain four sets of important terms: $set_{\text{participants}}$, set_{position}, set_{object}, set_{source}. Given the varying importance of terms, we calculate their frequency of occurrence and assign weights accordingly. After reordering terms by weight, we obtain weighted lists of important terms: $list_{\text{participants}}$, $list_{\text{position}}$, $list_{\text{object}}$, $list_{\text{source}}$. We also assume that the four types of important terms have different influences on the dependency relationship, setting different weights $\alpha_{\text{participants}}$, α_{position}, α_{object}, α_{source}. We aim for the entire construction process to be based on the semantic level, thus using the pre-trained word encoder GloVe (Pennington et al., 2014) to semantically encode the lists of important terms, obtaining weighted vector lists: $vec_{\text{participants}}$, vec_{position}, vec_{object}, vec_{source}. Now, we can calculate the dependency score between two nodes by computing the weighted sum of the cosine vector similarities. All the details of the hyperparameter(α, μ, λ) settings are provided in the Appendix A.1.3.

$$\text{DepScore}_{i,j} = \sum_{k \in \{\text{part,pos,obj,src}\}} \alpha_k \cdot \text{sim}(v_{k,i}, v_{k,j})$$

4.3.2 Calculating Node and Branch Attachment Scores

To ensure that the event evolutionary structures form a cohesive whole, we must design the construction method to enhance the distinctiveness between different tree branches while reinforcing the cohesion within the same branches. Our design idea is that when attaching a candidate node v to the tree at potential attachment node u, we should consider the dependency score $dep(v, u)$ as well as the dependency scores between v and the nodes in the parent branch (P_u) and sibling branches (S_u) of u. The dependency score of the potential attachment node accounts for a proportion μ. The scope of parent and sibling branch nodes is defined as tracing back to the first parent node with multiple children of the potential attachment node. Within the parent branch, node weights decay as the distance to the potential attachment node increases. Within sibling branches, all nodes are assigned the same weight. We want the dependency score between v and P_u to be large, and the dependency score between v and S_u to be small, thus setting a penalty coefficient λ.

$$\begin{aligned}\text{AttachScore}_{v,u} =& \mu \cdot \text{dep}(v, u) \\ &+ (1-\mu) \cdot \sum_{p \in P_u} w_p \cdot \text{dep}(v, p) \\ &- \lambda \cdot \frac{1}{|S_u|} \sum_{s \in S_u} \text{dep}(v, s)\end{aligned}$$

where:

- P_u: Set of parent branch nodes from u to, but not including, the first multi-child parent. $w_p = \exp(-\beta \cdot d(p, u))$, with $d(p, u)$ as the distance and $\beta > 0$.
- S_u: Set of sibling branch nodes sharing u's multi-child parent. $|S_u|$ is the number of siblings.

4.3.3 Constructing the news story tree

After determining the attachment method, we can construct a tree from scratch. Initially, we define the time of an event node as the average time of the news articles associated with it. We reorder the nodes chronologically to obtain an event node sequence. To better initialise the news story tree, we add a pre-construction phase to filter out nodes at the beginning of the sequence that are irrelevant to the topic. We select the first five nodes of the sequence to generate an initial tree. We then remove nodes from the event node sequence whose attachment scores fall below the initial attachment threshold and add them to a queue of detached

nodes. The remaining nodes are re-added to the sequence. For the formal construction of the news story tree, we sequentially take nodes from the sequence as candidate nodes to be attached. All nodes already attached to the tree serve as potential attachment nodes. We calculate the candidate node's attachment score to all potential attachment nodes and attach the candidate node to the node with the highest score. Similarly, during the formal construction phase, if a node's highest attachment score is below the attachment threshold, it is added to the queue of detached nodes.

4.4 Mining the Semantic Structure of the News Story Tree

Previous works typically attach nodes after extracting low-level textual features, neglecting semantic-level information. Benefiting from LLMs' powerful text generation capabilities, we mine the semantic structure of the news story tree after it is constructed. This allows users to grasp the hidden event evolutionary structure better. In particular, we execute the following tasks: (1) use an LLM to obtain event node summaries; (2) use an LLM to obtain the branching rationale and branch summaries for the news story tree; (3) adjust the branches based on their summary similarity. All prompts are provided in the Appendix A.1.2.

4.4.1 Obtaining Event Node Summaries Using LLM

We pass all news headlines under an event node to LLM to obtain a node summary.

4.4.2 Obtaining Branching Basis Using LLM

LLMs exhibit limited proficiency in handling long texts and tree-structured data. Therefore, we employ a leaf-root data processing method. After obtaining node summaries of all nodes, we start from the leaf nodes. We conduct a branch summarization for the branch containing a leaf node. The scope of this branch extends to the first multi-child parent node, excluding the multi-child parent node itself. We pass the node summaries of all nodes in this branch to LLM to get a branch summary. After obtaining all branch summaries for a multi-child parent node through post-order traversal, we pass these branch summaries and the node summary to LLM. We instruct the LLM to generate the branching rationale for distinct branches and to synthesize all branch summaries along with the node summary of the multi-child parent. This summary becomes the branch summary for the multi-child parent node. When a node has a branch summary, we prioritise using the branch summary over the node summary. This way, we can aggregate all branch information layer by layer. By traversing the entire news story tree in this post-order manner, we can obtain the branching rationale and branch summary for each branch. Additionally, we obtain a summary of the entire tree, as the root node's branch summary aggregates information from all nodes.

4.4.3 Adjust the branches based on their summary similarity

We use a pre-trained sentence encoder, all-MiniLM-L6-v2 (Wang et al., 2021), to encode node summaries. We perform a post-order traversal of the tree. We sequentially calculate the cosine similarity between a node's summary and its parent node's summary. If the similarity is below a threshold, the node and its children are reattached to the root node. If a child of the root node falls below the threshold, this child and its descendants are removed from the news story tree and added to the queue of detached nodes. Eventually, the detached nodes are presented separately to the user.

4.5 Predicting the next node and evaluating the quality of the predictions

After obtaining the story tree's semantic structure, we use LLM to predict the semantic level. Specifically, we treat the path from a leaf node to the root node as a news development chain. We use the root node's branch summary as background context. After passing the background and the news development chain to LLM, we require the model to predict the next node of the chain. Finally, we attach the predictions to the end of the news story tree and assess their quality. Specific assessment methods refer to Section 5.2. Additionally, we use the prediction model Qwen2.5-7B, which has been fine-tuned with the NewsForest dataset. Section 5.1 shows that this fine-tuning process effectively improves the model's prediction accuracy and quality.

5 Experiments

As the OEEF task is relatively new, there is no benchmark in the OEEF task. We will demonstrate the effectiveness of our method and dataset directly through experiments. Our experiments aim to answer two questions: (1) Can our ForestCast method

assist humans or LLMs in capturing event evolutionary structures from complex news? (2) Does our NewsForest dataset capture hidden real-world relationships that LLMs can learn?

5.1 Experimental Method

Ye et al. (2024) propose MIRAI to evaluate a model's ability to predict international relations. However, it cannot evaluate the prediction ability of models under OEEF. Therefore, we design an evaluation method inspired by the MIRAI. The difficulty in evaluating OEEF results is that if the prediction is general, it is more likely to be true. Thus, we need to evaluate both the quality and accuracy of the prediction. Referencing Guan et al. (2024) for assessing the quality of predictions, we implement a dual evaluation of prediction accuracy and quality. We extract the prediction results from the forecasting model. We then search online for some latest related news and pass this information on to an evaluation LLM. The evaluation LLM determines if the prediction has already occurred. If it has, we continue to score the prediction result for atomicity, validity, relevance, causal insight, and chain consistency(Appendix A.3).

We use ForestCast to create an up-to-date news story tree dataset. Topics include recent hot topics (Russia-Ukraine negotiations) occurring between April 13 and 20, 2025, and long-term popular topics from the past six months (US tariff policy). For these topics, we use ForestCast to construct a test dataset with time lengths of 20, 30, 40, 60, and 80 days, all ending on April 20, 2025. We reserve ten days to allow all true predictions to occur. We complete the evaluation of all predictions between May 1 and 3, 2025. To demonstrate that ForestCast can assist humans or LLMs in capturing event evolutionary structures from a flood of news, we create the news timeline dataset based on the news story tree dataset. We replace the node summaries in the chains with all news headlines under the event nodes. These headlines are then reordered by their publication time to form a news timeline dataset.

To demonstrate that the NewsForest dataset captures hidden real-world relationships that LLMs can learn, we train the Qwen2.5-7b model with NewsForest. In our experiments, we locally deploy the Qwen2.5 series models (Qwen, 2024)for testing. We also test Deepseek-V3-671b (DeepSeek-AI, 2024), one of the leading models without deep reasoning, and Gemini-2.5-pro-0325 (Gemini Team, 2023), one of the leading models with deep reasoning(via API). We provide training details in the Appendix A.4.1.

5.2 Evaluating Story Trees for Event Evolutionary Structures

We use five models to make predictions on the story tree dataset and the timeline dataset. Results are in Table 4 and 5. We analyze the experimental results in terms of both quality and accuracy, with specific conclusions as follows:

(1) Pass@1 accuracy is critical in real-world scenarios. And the news story tree dataset generally performs better on pass@1. Multiple rounds of generation show marginal gains. This may be due to the task's high demand for logical consistency in generation, resulting in multiple rounds of generation predicting in the same direction. In the real world, the predictions of each branch should also have unique directions.

(2) Deepseek-V3-671b and Gemini-2.5-pro-0325 partially perform better on the news timeline dataset. First, this is because the news timeline dataset has more raw data. This suggests that more capable models can capture more information from the raw data in the news timeline dataset, thus improving the prediction accuracy. However, DeepSeek and Gemini have significantly lower prediction quality than the other models. This suggests that with the emergence of multiple responses, the more capable models show a tendency to explore different directions of temporal development, similar to the different branches of our story tree. They also perform lower on atomicity, proving that the model improves the prediction correctness by giving fuzzy predictions. This proves that fine-tuning of small models as predictive models is necessary.

(3) The news story tree has higher scores and a wider range, indicating that the average performance and potential of the dataset are both greater.

(4) The largest gaps between the two datasets are in the causal and relevance metrics. This means that the news story tree is better able to grasp the event evolutionary structures.

(5) Additionally, the news story tree dataset performs more consistently on the long and short-term prediction tasks, as detailed in the Appendix A.4.2.

5.3 Evaluating the Enhancement of Prediction Capability with Fine-Tuned Models

To demonstrate that NewsForest captures learn progression drivers learnable by models, we use the

Model	News Timeline Dataset			News Story Tree Dataset		
	pass@1	pass@3	pass@5	pass@1	pass@3	pass@5
Qwen2.5-7b	37.11%	71.78%	86.22%	43.11%	78.22%	91.56%
Qwen2.5-7b-lora-dpo	38.66%	73.11%	86.89%	46.22%	79.77%	91.56%
Qwen2.5-14b	45.33%	78.67%	92.00%	48.67%	82.22%	92.67%
Deepseek-V3-671b	44.44%	78.66%	91.33%	47.78%	79.33%	90.67%
Gemini-2.5-pro-0325	48.44%	85.33%	94.44%	52.00%	81.33%	91.33%

Table 4: Prediction accuracy of different models on news timeline dataset and story tree dataset

Model	News Timeline Dataset					News Story Tree Dataset				
	Atom.	Rel.	Valid.	Causal.	Consist.	Atom.	Rel.	Valid.	Causal.	Consist.
Qwen2.5-7b	2.99	2.33	2.70	2.66	3.78	3.46	3.07	2.86	3.27	3.72
Qwen2.5-7b-lora-dpo	2.97	2.26	2.58	2.61	3.51	3.50	3.06	2.75	3.29	3.77
Qwen2.5-14b	3.01	2.44	2.29	2.71	3.55	3.48	3.19	2.91	3.40	3.77
Deepseek-V3-671b	3.02	2.47	2.12	2.56	3.23	3.41	3.16	2.56	3.02	3.72
Gemini-2.5-pro-0325	2.89	2.37	2.07	2.41	3.06	2.88	2.90	2.15	2.74	3.56

Table 5: Prediction quality scores(average of five responses) for different models on the news timeline dataset and news story tree dataset

NewsForest dataset to train Qwen2.5-7b. Comparing the prediction results of Qwen2.5-7b-lora-dpo and Qwen2.5-7b in Table 4 and 5, we find that:

(1) After training, our prediction accuracy improves significantly on two datasets.

(2) The quality of the news timeline dataset decreases, and its accuracy increases. We find that Gemini exhibits lower quality but higher accuracy on the news timeline dataset. Therefore, we hypothesize that the quality decline in the fine-tuned model is because it begins to explore different developmental directions on the news event timeline dataset, resulting in reduced quality but increased accuracy.

This indicates that NewsForest contains progression drivers that the LLM can capture.

6 Discussion

This research advances Open-Ended Event Forecasting (OEEF) through two major contributions: the NewsForest dataset and the ForestCast method. NewsForest is designed to enhance LLMs by teaching them event development logic. ForestCast organizes news into semantic story trees, enabling predictions along multiple event trajectories.

Experiments confirm that NewsForest and ForestCast significantly enhance LLM performance. Story trees significantly improve models' ability to understand complex event dynamics compared to linear timelines. Fine-tuning LLMs with NewsForest further enhances performance. On the news timeline dataset, the prediction accuracy of models increases, but the quality drops. This suggests that NewsForest's training encourages models to explore diverse event paths.

In addition, we made an important discovery. By comparing the performance of different models, we can prove that more powerful LLMs cannot replace fine-tuning models and organizing prediction background information. Larger models like Deepseek and Gemini sometimes perform better on the timeline dataset containing more information, but the prediction quality is significantly lower. However, NewsForest-trained smaller models better balance accuracy and quality.

This work addresses persistent OEEF issues. Existing datasets(Wang et al., 2025; Ma et al., 2024) often lack development logic, hindering LLM predictive power and even degrading performance. NewsForest counters this with rich event logic, enabling models to learn progression drivers. Traditional methods like basic clustering(Guan et al., 2024) fail to capture event dependencies, whereas ForestCast's story trees offer a sophisticated, semantically rich framework for forecasting.

Future research could explore the application of ForestCast to more diverse data sources, optimise the algorithm for building news story trees, and continue to improve the performance of LLMs under the OEEF task.

Limitations

First, our data source is singular, including only news. In contemporary society, social media also provides much critical information about event developments. Second, automatic evaluation methods based on LLMs may differ from human evaluation methods. Future work will aim to bridge the gap between these two approaches.

Ethics Statement

In our study, ForestCast is developed using open-source projects, including GDELT. These resources have been widely employed in other studies, ensuring that no ethical standards are compromised. Regarding compatibility with original access conditions, the GDELT data is publicly accessible for research, and our derivative dataset (NewsForest) is used solely within this context, ensuring compliance with GDELT's terms.

References

James Allan, Ron Papka, and Victor Lavrenko. 1998. On-line new event detection and tracking. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, pages 37–45. ACM.

Tommaso Caselli and Piek Vossen. 2017. The event StoryLine corpus: A new benchmark for causal and temporal relation extraction. In *Proceedings of the Events and Stories in the News Workshop*, pages 77–86, Vancouver, Canada. Association for Computational Linguistics.

Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised learning of narrative event chains. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 789–797. The Association for Computer Linguistics.

A. Feng B. Xue B.-Wang B. Wu B. Lu C. Zhao C. Deng C. Zhang C. Ruan C. Dai D. Guo D. Yang D. Chen D. Ji D. Li E. Lin F. Dai F. DeepSeek-AI, Liu. 2024. DeepSeek-V3 technical report. *CoRR*, abs/2412.19437.

Songgaojun Deng, Huzefa Rangwala, and Yue Ning. 2020. Dynamic knowledge graph based multi-event forecasting. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1585–1595. ACM.

Xiao Ding, Zhongyang Li, Ting Liu, and Kuo Liao. 2019. ELG: an event logic graph. *CoRR*, abs/1907.08015.

Li Du, Xiao Ding, Yue Zhang, Ting Liu, and Bing Qin. 2022. A graph enhanced BERT model for event prediction.

R. Borgeaud S. Alayrac J.-B. Yu J. Soricut R. Schalkwyk J. Dai A. M. Hauth A. Millican K. Silver D. Johnson M. Antonoglou I. Schrittwieser J. Glaese A. Chen J. Pitler E. Lillicrap T. Lazaridou A. Gemini Team, Anil. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. pages 2727–2733.

Yong Guan, Hao Peng, Xiaozhi Wang, Lei Hou, and Juanzi Li. 2024. Openep: Open-ended future event prediction. *CoRR*, abs/2408.06578.

Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, pages 1771–1776. ACM.

Kalev Leetaru and Philip A. Schrodt. 2013. Gdelt: Global data on events, location, and tone. *ISA Annual Convention*.

Manling Li, Sha Li, Zhenhailong Wang, Lifu Huang, Kyunghyun Cho, Heng Ji, Jiawei Han, and Clare R. Voss. 2021. The future is not one-dimensional: Complex event schema induction by graph modeling for event prediction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5203–5215. Association for Computational Linguistics.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing narrative event evolutionary graph for script event prediction. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4201–4207. ijcai.org.

Li Lin, Yixin Cao, Lifu Huang, Shu'ang Li, Xuming Hu, Lijie Wen, and Jianmin Wang. 2022. What makes the story forward?: Inferring commonsense explanations as prompts for future event generation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 1098–1109. ACM.

Bang Liu, Fred X. Han, Di Niu, Linglong Kong, Kunfeng Lai, and Yu Xu. 2020. Story forest: Extracting events and telling stories from breaking news. *ACM Trans. Knowl. Discov. Data*, 14(3):31:1–31:28.

Bang Liu, Di Niu, Kunfeng Lai, Linglong Kong, and Yu Xu. 2018. Growing story forest online from massive breaking news. *CoRR*, abs/1803.00189.

Yunshan Ma, Chenchen Ye, Zijian Wu, Xiang Wang, Yixin Cao, and Tat-Seng Chua. 2023. Context-aware event forecasting via graph disentanglement. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pages 1643–1652. ACM.

Yunshan Ma, Chenchen Ye, Zijian Wu, Xiang Wang, Yixin Cao, Liang Pang, and Tat-Seng Chua. 2024. SCTc-TE: A comprehensive formulation and benchmark for temporal event forecasting.

Ramesh Nallapati, Ao Feng, Fuchun Peng, and James Allan. 2004. Event threading within news topics. In *Proceedings of the 2004 ACM CIKM International Conference on Information and Knowledge Management, Washington, DC, USA, November 8-13, 2004*, pages 446–453. ACM.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. The Association for Computer Linguistics.

A. Yang B. Zhang B. Hui-B. Zheng B. Yu B. Li C. Liu D. Huang F. Wei H. Lin H. Yang J. Tu J. Zhang J. Yang J. Yang J. Zhou J. Lin J. Qwen, Yang. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Md Farhadur Rahman, Weimo Liu, Saad Bin Suhaim, Saravanan Thirumuruganathan, Nan Zhang, and Gautam Das. 2016. HDBSCAN: density based clustering over location based services. *CoRR*, abs/1602.03730.

Huan Rong, Zhongfeng Chen, Zhenyu Lu, Xiao-ke Xu, Kai Huang, and Victor S. Sheng. 2025. Pred-id: Future event prediction based on event type schema mining by graph induction and deduction. *Inf. Fusion*, 117:102819.

Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. 2012. Trains of thought: generating information maps. In *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pages 899–908. ACM.

Xiaoming Shi, Siqiao Xue, Kangrui Wang, Fan Zhou, James Y. Zhang, Jun Zhou, Chenhao Tan, and Hongyuan Mei. 2023. Language models can improve event prediction by few-shot abductive reasoning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2140–2151. Association for Computational Linguistics.

Zhen Wang, Xi Zhou, Yating Yang, Bo Ma, Lei Wang, Rui Dong, and Azmat Anwar. 2025. Openforecast: A large-scale open-ended event forecasting dataset. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 5273–5294. Association for Computational Linguistics.

Zhongqing Wang, Yue Zhang, and Ching-Yun Chang. 2017. Integrating order information and event relation for script event prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 57–67. Association for Computational Linguistics.

Zikang Wang, Linjing Li, and Daniel Zeng. 2024. Integrating relational knowledge with text sequences for script event prediction. *IEEE Trans. Neural Networks Learn. Syst.*, 35(7):9443–9454.

Yuwei Xia, Ding Wang, Qiang Liu, Liang Wang, Shu Wu, and Xiaoyu Zhang. 2024. Chain-of-history reasoning for temporal knowledge graph forecasting. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 16144–16159. Association for Computational Linguistics.

Wenjie Xu, Ben Liu, Miao Peng, Xu Jia, and Min Peng. 2023. Pre-trained language model with prompts for temporal knowledge graph completion. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 7790–7803. Association for Computational Linguistics.

Christopher C. Yang, Xiaodong Shi, and Chih-Ping Wei. 2009. Discovering event evolution graphs from news corpora. *IEEE Trans. Syst. Man Cybern. Part A*, 39(4):850–863.

Chenchen Ye, Ziniu Hu, Yihe Deng, Zijie Huang, Mingyu Derek Ma, Yanqiao Zhu, and Wei Wang. 2024. MIRAI: evaluating LLM agents for event forecasting. *CoRR*, abs/2407.01231.

Susik Yoon, Dongha Lee, Yunyi Zhang, and Jiawei Han. 2023. Unsupervised story discovery from continuous news streams via scalable thematic embedding. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 802–811. ACM.

Jinchuan Zhang, Bei Hui, Chong Mu, Ming Sun, and Ling Tian. 2024a. Historically relevant event structuring for temporal knowledge graph reasoning. *CoRR*, abs/2405.10621.

Jinchuan Zhang, Ming Sun, Qian Huang, and Ling Tian. 2024b. PLEASING: exploring the historical and potential events for temporal knowledge graph reasoning. *Neural Networks*, 179:106516.

Liang Zhao. 2022. Event prediction in the big data era: A systematic survey. *ACM Comput. Surv.*, 54(5):94:1–94:37.

Yucheng Zhou, Xiubo Geng, Tao Shen, Jian Pei, Wenqiang Zhang, and Daxin Jiang. 2021. Modeling event-pair relations in external knowledge graphs for script reasoning. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4586–4596. Association for Computational Linguistics.

A Appendix

A.1 ForestCast

Our method has been deployed live.Figure 3 is a screenshot of the main interface. In this section, we specify the additional functionality, method specifics, and time complexity of our implementation.

A.1.1 Function Demonstration

Keyword Extraction To enable users to grasp the development structure of the story tree quickly, we extract keywords for each node. The keyword extraction follows a rule prioritizing high recurrence rates within the same branch and low recurrence rates across different branches. Our keywords are derived from a deduplicated list of key entities associated with the node. An example is shown in Figure 4.

Tree Folding and Unfolding When there are too many news articles, it results in an excessive number of nodes. The interface cannot display the complete story tree. Therefore, we implement node folding and unfolding functionalities for the story tree. This allows users to focus on specific branches while maintaining an overview of the tree. An example is shown in Figure 5.

News Data Analysis To provide users with both a global and detailed understanding of news related to a topic, we analyze the raw news data. In the sidebar, we display the publication patterns of different news outlets across various periods. The colors of the nodes represent different media sources. Upon clicking a specific node, the sidebar displays the publication distribution of news linked to that node. An example is shown in Figure 3.

Branch and Node Information Display We present extracted semantic information to help users understand the event evolutionary structure. When hovering over a branch, we display the branching rationale; when hovering over a node, we show a node summary. Upon clicking a node, the original information is displayed, with news titles shown in the bottom-left interface. Clicking a title displays the corresponding full news article in the bottom-right interface. An example is shown in Figure 5.

A.1.2 Method Prompt

The prompt used for method implementation is as follows:

Branch Summary

If you are a journalist and you are given a chain of multiple news stories, please give this news chain a 100-word summary in English. There can only be a summary in the answer, and no extra words are allowed. The chain of news is as follows:

Branch Summary and branching rationale

If you're a journalism person, I'm going to give you multiple follow-up stories and a central story, and the follow-up news describes different aspects of the central news. Your two tasks are:
1. Please give all follow-up news a differentiating 2-8 word English phrase to summarise the dependency relationship between follow-up news and central news, focusing on discovering the difference in their dependency relationship and the main subject in the news.
2. Please make a coherent English summary of the follow-up news and the central news in 80 words. The summary should include time, place, person, cause, process, and result as much as possible.
The answer template is as follows (the number varies according to the actual number):
The Relationship between Follow-up News 1 and Central News: Specifics
The Relationship between Follow-up News 2 and Central News: Specifics
Summary: Summary in 80 words
Follow-up news is as follows: Follow-up news
Central News is as follows: Central News

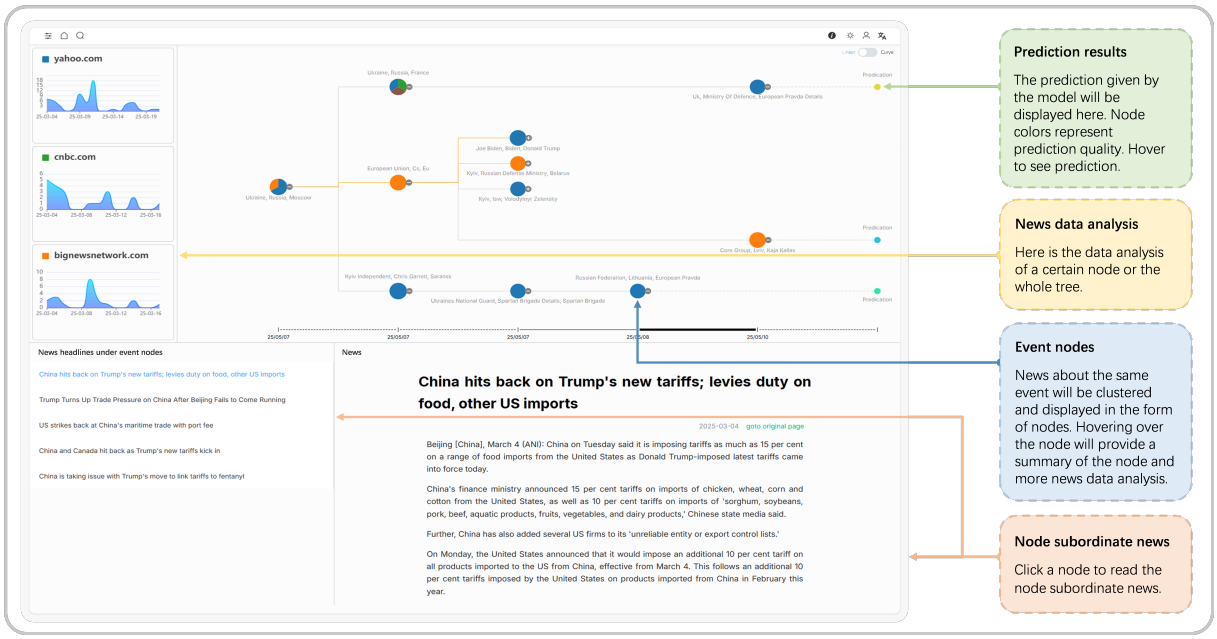


Figure 3: Website screenshot. The right side shows a description of each part of the website.

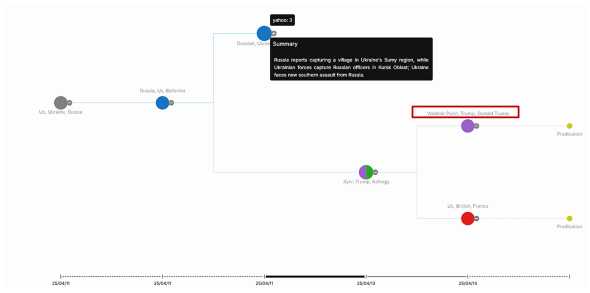


Figure 4: Node summary and display of keywords for event nodes in the user interface.

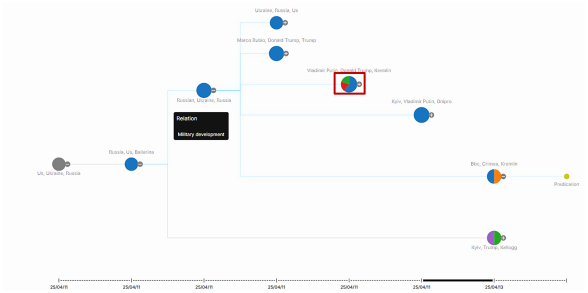


Figure 5: Demonstration of event node collapsing functionality and branch branching rationale in the user interface.

Node summary

Suppose you are a journalist and you need to help a stranger sort out the development of an event and write a coherent summary of the event. In that case, your task is to read the following news headline(s) and summarize the events described in the news in a concise sentence of 80 words or fewer.
1. Pay special attention to the changes and development of the core entity. Strictly cover all header elements to ensure that the logical chain is complete, the dynamic process is clear, and the data is not lost.
2. The summary should include the cause, process, result, time, place, and people as much as possible.

A.1.3 Setting of method hyperparameters

$\alpha_{\text{participants}} = 0.6$, $\alpha_{\text{position}} = 0.2$, $\alpha_{\text{object}} = 0.1$, $\alpha_{\text{source}} = 0.1$, $\lambda = 0.2$, $\mu = 0.9$

A.1.4 Time Complexity Analysis of Story Tree Construction

The time complexity of constructing the news story tree in the ForestCast is determined by the computational costs of its key steps. Let m be the number of event nodes, which are obtained from clustering news articles, and d be the dimension of the vector representations used for similarity computations.

Computing Dependency Degrees For each pair of event nodes, a dependency score (DepScore) is calculated based on the similarities of their important terms. This involves computing vector similarities in d-dimensional space. Since there are four

lists, each similarity computation takes $O(d)$ time. Therefore, the total time complexity for this step is:

$$O(4 \cdot d)$$

Attaching Nodes to the Tree Nodes are attached to the tree sequentially. When attaching a new node v, an attachment score (AttachScore) is computed with each existing node u in the tree. We roughly assume that there are m nodes in the sibling branch and the parent branch. Since there are up to m potential attachment points for each of the m nodes, the total time complexity for this step is:

$$O(m^2 \cdot d)$$

Overall Time Complexity The overall time complexity of the story tree construction is:

$$O(m^2 \cdot d)$$

where m is the number of event nodes and d is the dimension of the term vectors.

The Efficiency of its Operation in Real Situations Our method is laid out on 2*A6000. When the server is not congested, it takes about 5 minutes to complete the whole process for 800 articles, 3 minutes for 500 articles, and 1 minute for 100 articles.

A.2 NewsForest Dataset

A.2.1 Dataset Overview

Our dataset comprises global news data collected over six months. Table 7 shows the data distribution across different domains. The overall dataset statistics are presented in Table 8.

A.2.2 Dataset Evaluation Protocol

We show a detailed description of all indicators in Table 3. The evaluation prompt case is as Table 6.

A.2.3 Dataset Evaluation Results

We conduct further analysis of the dataset and found that various metrics remain stable across different node counts and chain lengths. The results are presented in Figure 6 and Figure 7.

A.3 Evaluation Method

For searching for recent news, we use Tavily search api, which is set to search for the ten most relevant news items, and the search time is within 10 days. We then pass the information to the evaluation model Qwen2.5-7b to determine if the prediction occurred. For the quality assessment part, we reuse some of the story tree evaluation metrics. We assess the first five indicators in the Table 3.

A.4 Experiment Details

A.4.1 Training Parameters

The training protocol we use is LLama-Factory. We train on 4*A6000 with the following training parameters.

Method Parameters

```
stage: dpo
do_train: true
finetuning_type: lora
lora_rank: 32
lora_target: all
pref_beta: 0.1
pref_loss: sigmoid
```

Training Parameters

```
per_device_train_batch_size: 1
gradient_accumulation_steps: 16
learning_rate: 5.0e-6
num_train_epochs: 6.0
lr_scheduler_type: cosine
warmup_ratio: 0.15
bf16: true
ddp_timeout: 180000000
resume_from_checkpoint: null
```

Dataset Parameters

```
dataset: data
template: qwen
overwrite_cache: true
preprocessing_num_workers: 16
dataloader_num_workers: 4
```

Evaluation Parameters

```
val_size: 0.1
per_device_eval_batch_size: 1
eval_strategy: steps
eval_steps: 100
```

A.4.2 Model Evaluation resluts

We also analyze accuracy over different time lengths. Because the quality metrics are complex, we analyze only the accuracy to focus on the im-

Score	Scoring Criteria	Example
1 (vague)	Mentions general idea only; lacks specifics (actions, people, results). Uses general words (e.g., "attention").	Chang'e 6 attracted widespread international attention.
2 (somewhat specific)	Little specific info, mostly vague. You may mention the event type/reaction, not the specific participants/actions/results.	After Chang'e 6 completed its important mission, it received some international feedback.
3 (moderately specific)	Some key details (core content), giving a general idea; may lack specifics (participants, results, background).	Chang'e 6 successfully collected and returned samples... attracting international attention.
4 (more specific)	Most key info (event, people/institutions, specific results/reactions). Core elements are relatively clear.	Chang'e 6 successfully brought back samples... space agencies from many countries expressed congratulations.
5 (very specific)	Clearly describes main events, identifies key participants, specific actions, results/reactions. Provides verifiable details.	Chang'e 6 successfully brought back samples... Russia sent congratulations to China.

Table 6: Scoring standard for atomicity provided to an LLM

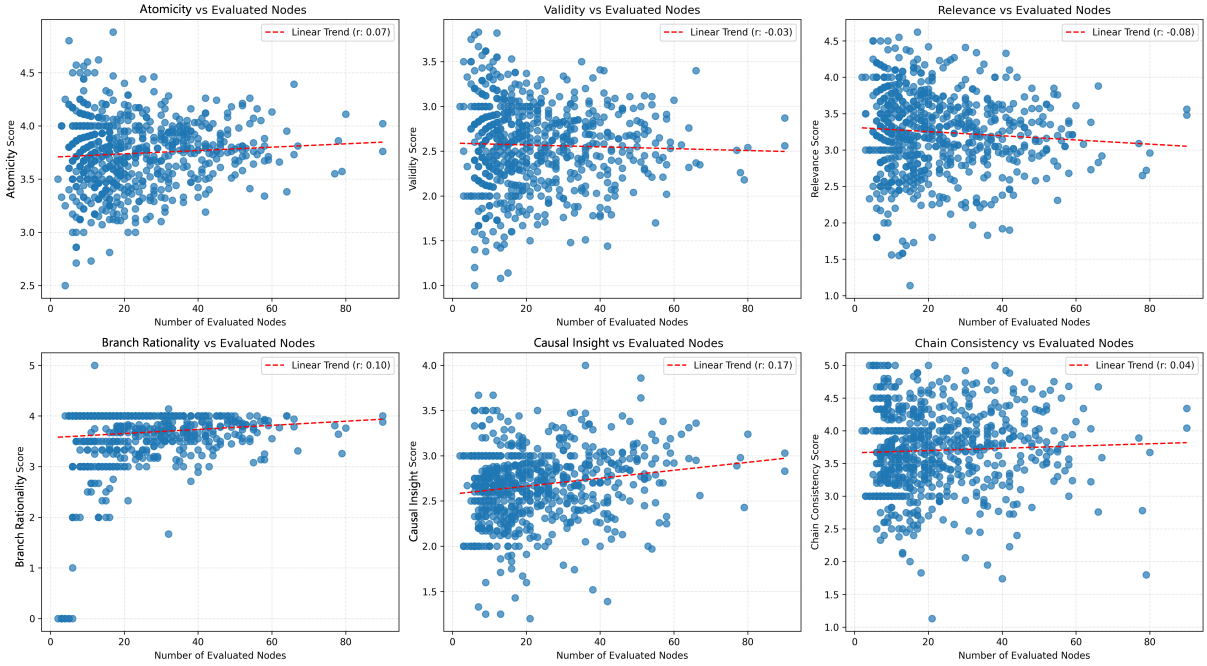


Figure 6: Evaluation results for the dataset: Metric trends as the number of nodes changes.

Attribute	Story Chain	Prediction Chain
Average Length	4.39	2.95
Max Length	14	9
Min Length	2	1

Table 7: Chain length data before and after dataset processing

Domain	# Trees	# Story Chain	# Prediction chain
Economics	157	1726	2396
Politics	284	3613	5326
Military	217	2801	3942
Social Events	36	461	742
Total	**694**	**8601**	**12406**

Table 8: Data volume per domain

pact. We analyze the prediction accuracies over different time lengths on the story tree dataset and the news timeline dataset(Figure 8a and Figure 8b).

The highest prediction accuracy is found at 30 days. We hypothesize that this is because the predictions at 20 days may not have happened yet to be confirmed. Beyond 30 days, information becomes more complex, prediction becomes naturally more difficult, and prediction accuracy decreases. Comparing the performance of the different models, we can see that Gemini and DeepSeek perform better in long-term prediction. Qwen-14b performs stably in short-time and long-time prediction. Comparing

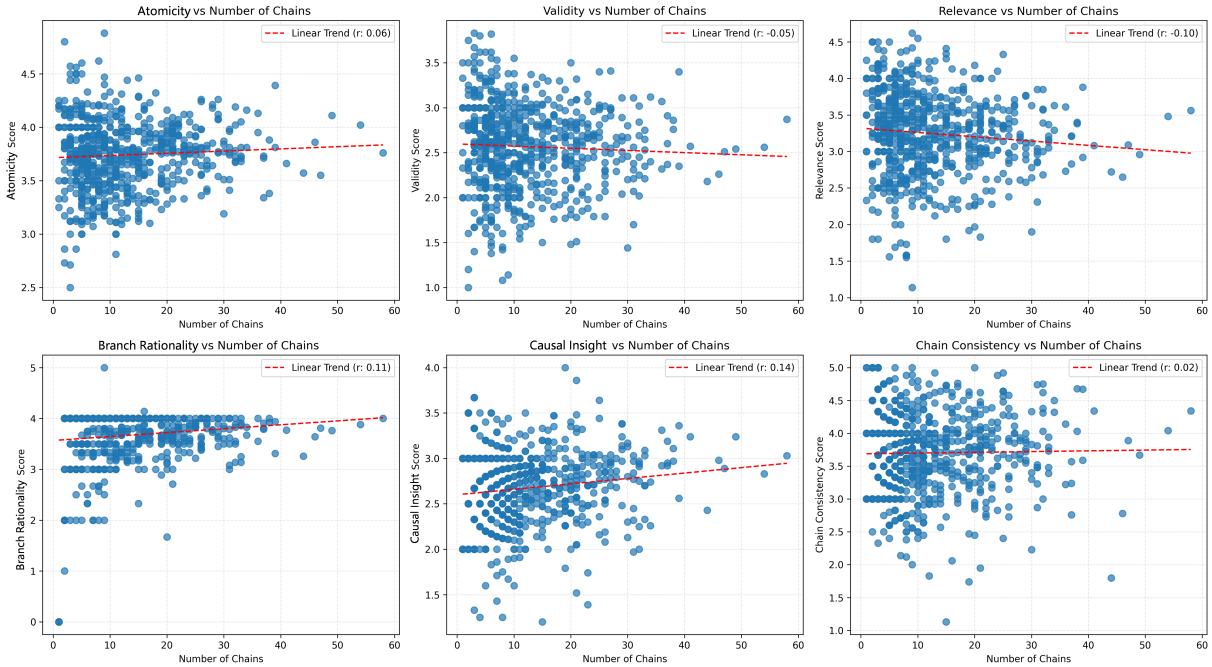
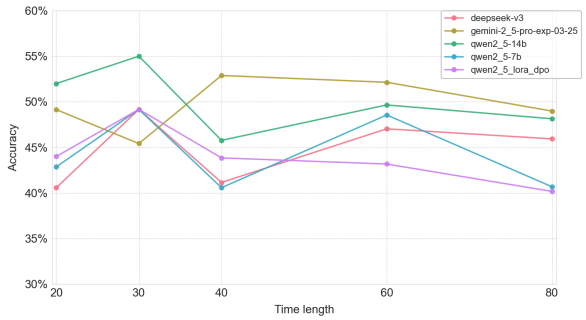
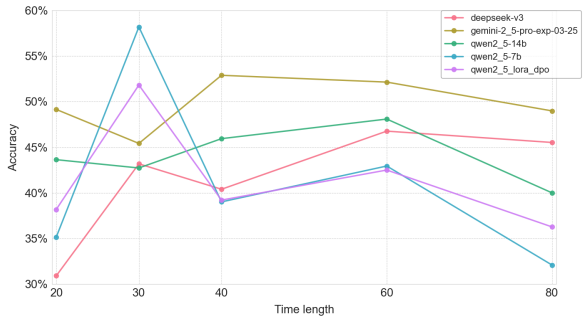


Figure 7: Evaluation results for the dataset: Metric trends as the average chain length changes.

different datasets, we find that the models perform more consistently on the story tree dataset.



(a) Prediction accuracy curves on news story tree dataset.



(b) Prediction accuracy curves on news timeline dataset.

Figure 8: Prediction accuracy curves across time lengths