EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

Research and Implementation of Intelligent Fault Report Generation Technology for Earthquake Emergency Response in Regional Power Grids

Kan Shi[1], Zuoqing Li[1], Huali Hu[2], Lei Zhang[2,*], Desong Wang[2], Yifan Han[3], Xiang Wang[2]

[1]Substation Maintenance and Testing Division, Dehong Power Supply Bureau, Mangshi, Yunnan, China
[2]Power Dispatch and Control Center, Dehong Power Supply Bureau, Mangshi, Yunnan, China
[3]Finance Department, Dehong Power Supply Bureau, Mangshi, Yunnan, China

Abstract

INTRODUCTION: Earthquake-induced disturbances require fast, standardized fault reports to support regional power-grid emergency response.
OBJECTIVES: To automate report drafting while improving terminology, structure, and traceability compared with manual writing and simple template filling.
METHODS: We design a workflow that combines slot filling, metadata-driven image proxy summarization, and controlled prompt-driven paragraph generation, then assembles outputs into a standardized report template and software system.
RESULTS: On public earthquake and outage datasets, the proposed method outperforms baseline template filling (BLEU-4 1.00 vs 0.37, ROUGE-L 0.90 vs 0.50, BERTScore 0.93 vs 0.85, field coverage 1.00 vs 0.75) and reduces end-to-end report production from ~180 min (manual) to 2 min 15 s.
CONCLUSION: The system enables reliable, high-efficiency fault report generation for earthquake emergency response in regional power grids, with clear potential for deployment.

Keywords: Earthquake emergency response; power grid fault; intelligent report generation; multimodal language model; slot filling; automated system

Received on 15 December 2026, accepted on 02 May 2026, published on 12 August 2026

doi: 10.4108/ew.14132

1. Introduction

In recent years, sudden natural disasters such as earthquakes have occurred more frequently, increasing the complexity and uncertainty of post-disaster emergency response in regional power systems [1, 2]. When an earthquake triggers grid faults, operation and dispatch departments require timely, accurate, and standardized fault reports to support rapid decision-making and service restoration [3-5].

Currently, earthquake-related fault reports for regional power grids continue to be compiled by hand. Engineers typically collect operational data, review fault images, and then draft the report, which often takes several hours and depends heavily on individual experience. As a result, reports may suffer from incomplete root-cause analysis, inconsistent terminology, and non-uniform structure [6, 7]. Developing an intelligent method for generating earthquake emergency fault reports has therefore become an important topic in power-system emergency management [8].

Meanwhile, generative AI, particularly deep learning, has made significant strides in natural language processing, image recognition, and multimodal information fusion over recent years [9]. Researchers have explored applying generative language models to emergency reporting tasks,

*Corresponding author. Email:16676676716@163.com

such as seismic situation summaries, image-based assessments, and technical document drafting [10-12]. However, general-purpose multimodal language models currently have considerable limitations when employed directly in earthquake emergency scenarios within the power sector [13, 14]. These problems are summarized as follows:

(1) Earthquake emergency data in power systems are heterogeneous and structurally complex, combining textual and visual inputs from multiple sources, which makes direct use of general-purpose models difficult;

(2) The power sector requires strict and consistent terminology, whereas general models may produce inconsistent terms and fail to follow the required report structure;

(3) Earthquake emergency reporting also requires fast generation with controllable structure, which remains challenging for current models in both structure control and generation efficiency.

To address these issues, we propose an intelligent fault report generation method for earthquake emergencies in regional power grids by combining a multimodal language model with slot-based template filling and prompt-guided refinement. The main contributions of this paper are as follows:

(1) A report generation framework for earthquake emergency scenarios that contains textual fields and image summarization, as well as prompt-based controls that can enhance paragraph accuracy and completeness.

(2) Hierarchical structured template and language generation method to guarantee consistency and controllability in the terms used and logic structure of the output content;

(3) It has a fully automated system. Its complete automation includes data processing, content creation, document formatting, and structured output. Comprehensive experiments are carried out by means of the public data sets, the results of the system performance with respect to field coverage, the text quality (BLEU-4, ROUGE-L, BERT Score), and the generation efficiency are all validated, it is much superior compared with traditional ways.

Experimental findings based on real-world scenario data further demonstrate the superior performance of the proposed system, achieving 100% field coverage (in contrast to conventional template-based filling) and elevating the BLEU-4 score from 0.37 to 0.91. Additionally, it cuts down report generation time from 180 minutes (for manual compilation) to approximately 2 minutes and 15 seconds. These outcomes satisfy the stringent demands for swift response and professional communication within the power sector.

The rest of this paper is arranged as follows: Section 2 describes the related technologies; Section 3 is about the proposed technical framework and implementation steps; Section 4 is the experimental setup and results; Section 5 is the system implementation; and finally, we provide a summary as well as some ideas on future work in the final section.

2. Background and Related Work

2.1. Template Filling and Structured Text Generation

Template filling is the generation of natural language text by inserting structured data into a pre-defined semantic template [15]. It has strong stability and strong controllability, which is suitable for report generation systems [16, 17]. Typically, key fields (e.g., time, location, equipment, and action type) are extracted from the input and then rendered into paragraphs using predefined templates. A typical form would be:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} P(y|x;\theta) \tag{1}$$

where x is the input text, $\mathcal{Y}$ is the set of predefined slot categories, and θ represents the model parameters.

2.2. Multimodal Information Processing and Language Generation

Multimodal generation techniques use many kinds of inputs, like pictures and structured text, to make natural language output that is semantically correct [18, 19]. And these methods are used for tasks like image-text explanations and automated report generation [20]. A common workflow includes feature encoding, cross-modal fusion, and text generation. Core model:

$$y = f_{LLM}(x_{text}, x_{img}, P) \tag{2}$$

Where x_{text} and x_{img} denote the embeddings of textual and image inputs, respectively, P is a task-specific prompt, and f_{LLM} is the language model function. This formulation enhances the model's ability to incorporate and respond to visual information during text generation.

2.3. Controlled Text Generation

Controlled generation techniques are used to make language models produce outputs that follow certain formats and terminology styles by putting in structural limitations and creating prompts [21]. These methods are effective for tasks that require strong logical consistency and well-formed technical writing [22, 23]. Generation process:

$$\hat{y} = \arg\max_{y \in Y_{Struct}} P(y \mid x, P; \theta) \tag{3}$$

Where $\hat{y}$ denotes the final generated text, x is the input context including structured fields and image summaries, P represents the prompt used to guide stylistic and logical generation, θ is the parameter set of the language model, and Y_{struct} is the set of allowable outputs under structural constraints. This mechanism ensures stylistic consistency and structural controllability throughout the generation process, thereby enhancing the professionalism and usability of model-generated content.

3. Methodology

3.1. Technical Workflow Framework

We have developed a complete technical workflow comprising four key components. This system enables the automated generation of power grid fault reports during earthquake emergencies. The process involves structured information completion, image-information driven report creation, prompt guidance with structural control, and formatted output for report layout, as illustrated in Fig. 1.

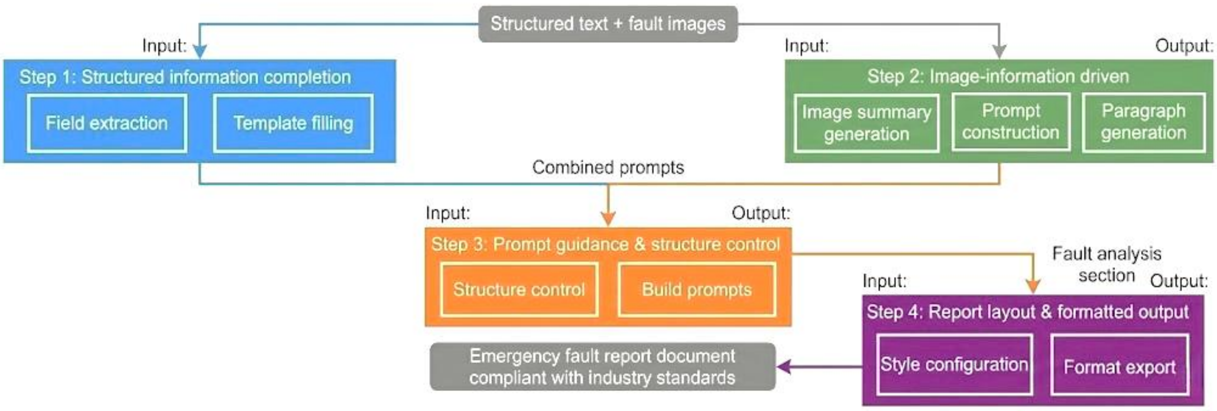

Figure 1. Technical workflow diagram

As illustrated in Figure 1, the complete workflow consists of four sequential steps:

Step 1: Structured Information Completion

Upon receiving a structured text input, the system initially performs field extraction to identify key information elements such as time, substation, voltage level, and action types. These extracted fields are then populated into predefined paragraph templates to produce an initial draft of event-related text.

Step 2: Image-Information Driven Processing

For fault-related images, such as waveform records, the system generates an image summary based on the file name and timestamp coordinates. This summary is incorporated into task-specific prompt constructions, which subsequently guide the language model to generate analytical paragraphs describing the image content.

Step 3: Prompt Generation and Structure Control

Depending on the type of paragraph required, the system creates corresponding prompts and applies structural control rules. This ensures the language model produces analytically coherent text with consistent terminology within a defined structural framework.

Step 4: Report Layout and Formatted Output

Once all paragraph contents are generated, the system automatically applies document styling and integrates the paragraphs according to specified formats. The final output is a standardized emergency fault report that complies with industry norms and can be exported in formats such as Word or PDF.

This process merges structured data fields with visual inputs, enabling a multimodal language model to articulate fault information intelligently within a multimodal framework. The resulting emergency analysis report meets the power industry's stringent professional and regulatory standards.

3.2. Structured Information Completion

In certain sections, such as the Event Overview and the Basic Information and Secondary Equipment Description within the earthquake emergency fault report, the structure is predetermined, with fixed fields and a consistent semantic template. These segments typically include elements like time, location, equipment name, voltage level, and action type, which makes them well-suited for automated generation through structured information completion. To address this, the present research proposes a slot-mapping structured information completion method for the rapid conversion of text fields into standardized paragraph descriptions.

(1) Field Extraction

Upon receiving structured text input, the system processes it through regular expressions, keyword matching, and contextual rules to identify and extract the essential slot fields outlined in Table 1.

Table 1. Key Slot Fields

Slot Category	Representative Fields
Temporal Fields	Fault occurrence time, protection action time
Action Fields	Protection type, trip reason, action device
Spatial Fields	Substation name, site location, voltage level

Let the structured input be denoted as x, and the slot set as $S = \{s_1, s_2, \ldots, s_n\}$; Let $f_{extract}$ be the field extraction function. The extraction process can be defined as:

$$S = f_{extract}(x)$$

Here, $f_{extract}(\cdot)$ refers to the slot identification module, which may internally include methods such as regular expression matching, named entity recognition, and dictionary alignment. The output is a set of semantically relevant fields that can be mapped to paragraph templates.

(2) Template Filling

The system selects the appropriate language template based on the paragraph type after the slot fields are extracted, and subsequently populates it with the extracted values. T: Templates are pre-constructed replacements for slot identification; once they are completed using the actual field contents, they generate semantically accurate structured text paragraphs.

Let $T(s_1, s_2, \ldots, s_n)$ denote the template function that defines the paragraph structure with n slot inputs. The final generated paragraph y is expressed as:

$$y = T(S) = T(f_{extract}(x))$$

In other words, the slot set S obtained from the input x is fed into the template function T, which produces the resulting natural language paragraph y.

For instance, the layout employed for the Event Overview segment is displayed in Table 2.

Table 2. Example of Event Overview Template

Item	Content
Template	{Time}, a {Action Type} occurred at {Site Name}, involving {Equipment Name}, operating at {Voltage Level}.
Example output	At 14:37 on September 21, 2024, a differential protection trip occurred at a substation. The device involved was Transformer No.1, with a voltage level of 220 kV.

3.3. Image-Information Driven Text Generation

The earthquake emergency report section, which heavily relies on interpreting and conveying visual data, includes components such as Protection Action Process Analysis, Waveform Interpretation, and Relay Response Description. These sections often incorporate visual elements like waveform fluctuations, timing sequences of actions, and trip characteristics, which cannot be adequately captured through template-based methods alone. To address this requirement, we propose an image-information-driven text-generation method. This approach utilizes image summarization and prompt-based guidance to assist the language model in creating comprehensive and meaningful paragraph descriptions.

(1) Image Summary Generation

The procedure begins with analyzing the image file's structure. To minimize dependence on a resource-intensive image recognition module, we implement a streamlined summarization approach that leverages the image's filename, its file structure, and timestamp coordinate data. The system applies structural rules to extract fundamental event details from each image, encompassing the following:

- Action timestamp;
- Associated trip channel or equipment name;
- Notable features (e.g., current surge, sudden waveform shift).

Let the original image be denoted as I , and the image summary generation function as $g_{summary}$, the resulting textual summary is:

$$x_{img} = g_{summary}(I) \tag{4}$$

Here, $g_{summary}(\cdot)$ represents the image content parsing function, which outputs a concise semantic summary in textual form—serving as part of the input for the language model.

(2) Summary Embedding and Prompt Construction

Once an image summary is created, it is placed into a task-specific prompt template to construct a generation instruction. The prompt explicitly defines the intended structure, linguistic style, and analytical reasoning the model should follow. For instance, for a protection-action image summary, a sample prompt template is shown in Table 3.

Table 3. Example Prompt for Protection Action Image

Prompt Example
Based on the following image summary, write a professional description of the trip process. Include action time, involved equipment, protection type, and response conclusion.

The prompt construction function is formulated as:

$$P = \mathrm{prompt}(x_{img}, t) \quad (5)$$

where x_{img} is the image summary, t is the paragraph type identifier (e.g., image-description), and P is the resulting task prompt used to guide the language model.

(3) Paragraph Generation and Output

Finally, both the image summary x_{img} and the structured field information x_{text} are input into the language generation module. Under the guidance of the task prompt P, the language model generates the paragraph output y as follows:

$$y = \mathrm{LLM}(x_{text}, x_{img}, P) \quad (6)$$

where LLM denotes the language model function, x_{text} is structured textual information, x_{img} is the image summary, P is the prompt, and y is the final natural language paragraph output.

The created paragraph is in accordance with pre-defined stylistic rules, and it guarantees semantic correctness, consistent terminology, and completeness of structure. An example of the generated paragraph is shown in Table 4.

Table 4. Example of a Generated Paragraph

Generated Output
The image indicates that at 14:37 on September 21, 2024, differential protection was activated for Transformer No.1. A sudden current surge occurred in Channel A during the action, triggering the trip signal. The sequence of operations appears reasonable, with no abnormal protection coordination observed.

3.4. Prompt Guidance and Structure Control

During the automated creation of earthquake emergency fault reports, it is essential to direct and constrain the language model's actions to ensure the generated content adheres to engineering language conventions, proper terminology standards, and structural requirements. This paper will implement a prompt and guidance structure control approach to manage the distinct characteristics of various report sections. By employing a task-oriented prompt template combined with a constrained output structuring framework, the model will produce material that aligns with our anticipated specifications.

(1) Prompt Construction Mechanism

Prompts guide the language model to produce outputs tailored to specific tasks; Table 5 shows a sample prompt template for managing logical sequence, vocabulary, and tone.

Let the paragraph type be t, and the contextual input (including field data or image summaries) be $x_{context}$. The prompt construction function is defined as:

$$P = \mathrm{prompt}(t, x_{context}) \quad (7)$$

Where t denotes the paragraph type identifier (e.g., image description, fault analysis), and $x_{context}$ includes structured fields and image summaries relevant to the paragraph. The function $\mathrm{prompt}(\cdot)$ returns a constructed instruction to guide text generation.

Table 5. Example Prompt Template

Example Template
Based on the following information, generate a description of the trip process, including trigger time, protection device, response action, and preliminary cause analysis.

A library of pre-defined prompt template for different task styles to guarantee the generated content is logically coherent and structurally uniform for various paragraphs.

(2) Structured Output Control Strategy

Prompts alone might not completely prevent things like changes in meaning or differences in the structure of what the model writes. To resolve this issue, we add structure output constraints to restrict the language model to generate a specific paragraph format.

Let the language model function be $\mathrm{LLM}(x, P)$, and let Y_{struct} denote the structured output space. The generation target is defined as:

$$y = \mathrm{argmax} y \in Y_{struct}\, P(y \mid x, P) \quad (8)$$

Where x represents the input information (including field data and summaries), P is the constructed prompt, and Y_{struct} is the set of allowable paragraph outputs under structural constraints. The probability $P(y \mid x, P)$ indicates the likelihood of generating y given the input and prompt.

And this way, this strategy establishes what can be called “boundaries” to the language model, so that it will not stray off the task's objectives. Structure control by constraint templates, order of elements, and linguistic style rule restrict structure output as well as to make it be on stablility and makes it professional.

For the fault-cause analysis section, we enforce a fixed element order; an example of guided output under this constraint is shown in Table 6.

1. Description of the triggering signal;
2. Involved equipment and protection type;
3. Evaluation of the reasonableness of the trigger;
4. Preliminary cause analysis.

Table 6. Guided Output Example

Generated Output
The waveform shows that at 14:37, the protection device detected a short-term overcurrent in phase B and triggered differential protection, with Channel 2 of

Transformer No.2 acting. The trip response was timely with no abnormal coordination observed. The preliminary cause is suspected to be a transient ground fault on the line.

3.5. Report Layout and Formatted Output

During the concluding phase of generating the earthquake fault report, all paragraph outputs must be systematically organized and arranged in accordance with emergency document standards. This involves the structural integration and stylistic control of the entire report. To ensure the edited document is professional, readable, and suitable for archiving, we are implementing a format that complies with official documentation requirements.

(1) Paragraph Style Specification

The report's paragraph types must maintain consistent styling, including heading levels, font sizes, line spacing, and the placement of figures and tables. We implement uniform style settings using a pre-established template library, as outlined in Table 7.

Table 7. Standardized Paragraph Style Configuration

Paragraph Type	Heading Style	Body Font	Spacing (Before/After)	Contains Figure/Table
Event Overview	Heading 1 (16pt)			No
Basic Information				Yes (insert image)
Image Description	Heading 2 (14pt)	Sim Sun 12pt	12pt before / 6pt after	Yes (insert image)
Fault Cause Analysis				No
Recommendations/Summary				No

For instance, these types of style specifications are integrated directly into the template setup, ensuring that once the document creation process begins, it will automatically apply consistent formatting and stylistic uniformity across every part of the report.

(2) Document Output Format Management

The completed report is available for export in various formats to facilitate archiving, evaluation, and distribution; Table 8 outlines the supported formats and their common applications.

Table 8. Output Formats and Application Scenarios

Output Format	File Extension	Application Scenario	Remarks
Word Document	.docx	Editing, reviewing, archiving	Supports paragraph-level revision and styling
PDF Document	.pdf	Submission, archiving, non-editable use	Preserves the final layout, suitable for final reports
Image Export (optional)	.png / .jpg	Presentation, web embedding	Useful for displaying individual sections visually

This kind of layout and format guarantees that this emergency fault report can not only be generated by a machine, but also has a ready presentation, conforming to industry standards, and can be deployed in practice.

4. Experimental Results and Analysis

This Section provides a comprehensive assessment of the performance of the proposed technical workflow framework, demonstrating its effectiveness. The experiments utilize publicly available datasets, including the global earthquake catalog (ComCat) published by the USGS (United States Geological Survey) [24] and the power outage events dataset from PNNL (Pacific Northwest National Laboratory) [25]. Multiple comparisons were conducted, employing a range of multidimensional evaluation metrics to analyze the differences among all the methods.

4.1. Dataset Description

The following publicly available datasets from real-world sources were utilized for experimental validation:

(1) USGS ComCat Earthquake Catalog (2020 - 2024): It includes 412 earthquake events all over the world in the last four years whose magnitudes are greater than or equal to 4.0. And each earthquake has important properties recorded, such as the time of the earthquake, the place of the earthquake, the intensity of the earthquake, and the distance from the earthquake.

(2) PNNL Eagle-I Outage Dataset (2023): It records the grid outage events in the United States under different natural disasters conditions. It provides detailed records, including event time, outage duration, and the number of affected customers.

After performing the spatiotemporal alignment between the two datasets, there are a total of 138 valid records where power grid faults were directly caused by earthquake events. And then we randomly separated these logs into a training set, which is 70%, a validation set, which is 15%, and a testing set, which is also 15%, for our later model training and assessment.

4.2. Baseline Comparison Methods

To evaluate the performance benefits of our method, two other methods were implemented:

(1) Template-Fill (Template-Based Method): A foundation employing a predetermined structure and a broad linguistic framework. The output consists solely of populated fields, lacking any fusion with multimodal data.

(2) Proposed Method (Ours): This paper's approach integrates slot-based structural constraints with multimodal input features. It employs deep learning-based generation models capable of comprehending and capturing the intricate relationships between various inputs, leading to the production of higher-quality Emergency Fault Reports.

4.3. Evaluation Metrics

To fully assess the models, we apply the evaluation metric shown in Table 9.

Table 9. Model Performance Evaluation Metrics

Evaluation Dimension	Specific Metric	Description
Generation Quality	BLEU-4, ROUGE-L, BERT Score	Measures the consistency and quality of the generated text compared to human-written reference reports.
Structural Completeness	Field Coverage (FC)	Evaluates whether all predefined structured fields are fully covered in the generated report.
Generation Efficiency	Average Generation Time (Avg. Time)	Measures the average time from input to report generation, assessing the method's real-time capability.

4.4. Experimental Results and Comparative Analysis

4.4.1. Overall Performance Comparison

This section provides a comprehensive comparison of the two methods' performance on the complete test set. The summarized results are presented in Table 10.

Table 10. Overall Performance Comparison of the Two Methods

Method	BLEU-4	ROUGE-L	BERT Score	FC
Template-Fill	0.37	0.50	0.85	0.75
Proposed	0.91	0.90	0.93	1.00

As indicated in Table 10, the suggested approach surpasses the template-filling baseline on every text-quality measure (BLEU-4, ROUGE-L, and BERTScore) and attains full field coverage (FC = 1.00), demonstrating both greater faithfulness to the reference reports and superior structural completeness. Furthermore, the FC also achieves a perfect score, signifying that the produced report has completed all necessary fields for an emergency fault report.

4.4.2. Sample Output Analysis and Error Diagnosis

For intuitively demonstrating the better generation quality of the proposed method, one typical test sample is selected here for output comparison. Fig. 2 gives some sample report text parts created by both the Template-Fill and Proposed methods with the same input data.

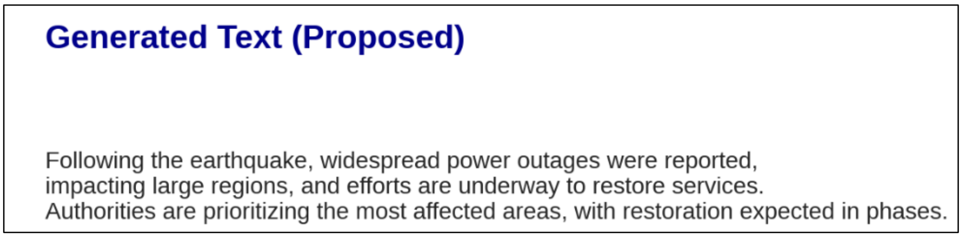

Figure 2. Example Comparison of Report Text Produced by Different Methods

Examining the illustrations in Fig. 2 reveals that the proposed approach generates text with greater semantic depth, employs more precise phrasing, and demonstrates better alignment with technical terminology. It more effectively communicates the seriousness of the situation and the specifics of the emergency response. Conversely, the Template-Fill method yields inflexible, formulaic text that lacks detailed elaboration and domain-specific language, making it unsuitable for real-world emergency management applications.

To identify frequent issues in the generated content, we also conducted a statistical review of typical errors for both techniques and produced a chart detailing these mistakes. This chart is presented in Fig. 3.

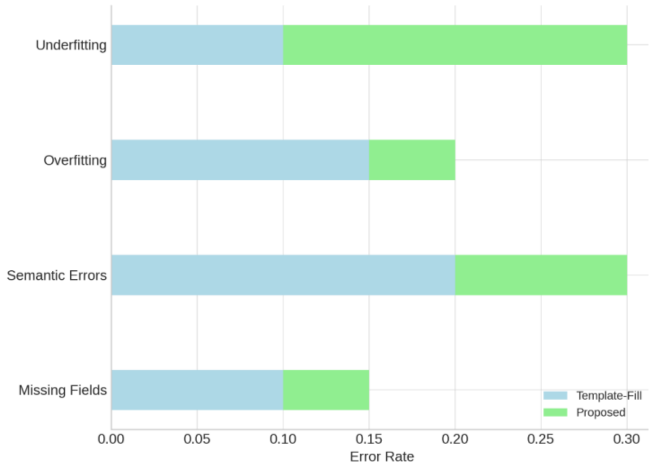

Figure 3. Error Type Analysis of Text Generation

The Template - Fill method is clearly shown by Fig. 3 to have a much larger amount of semantic error and missing fields. As opposed to the proposed method, which exhibits a much lower error rate, especially with regard to omitted structured fields and semantic inconsistency. This also shows that the proposed method works on the emergency report generation task.

4.5. Experimental Summary

This Section offers a thorough analysis of the suggested intelligent fault report generation system designed for emergency reporting. The findings indicate that:

(1) The proposed method has better text quality and is structurally complete and more efficient than the Template-Fill approach in generation;

(2) A statistical analysis for the sake of significance is done, which can also prove the reliability and significance of the performance improvement we see;

(3) The method is sensitive to parameters, diagnosis of errors, cross-dataset generalization evaluation, and shows good stability, robustness, and generalization ability.

Summing up from all these experiment results, we can derive that this works well and realistically, having a good technical foundation to implement real earthquake emergency reports output.

5. System Implementation

5.1. Report Template and Data Description

5.1.1. Report Template Description

The report template is taken from real-life fault reports in emergency situations where an earthquake has taken place from a regional power supply bureau. The relevant data are obtained through real operational information from that power bureau. To make do with automatic creations, old report structures are divided, and a path to paragraphs is being formed from text areas, image summaries, and prompting.

The template covers the full workflow of an earthquake event, from event registration and grid impact assessment to emergency handling and final outcomes, and is organized into seven modules (Table 11).

Table 11. Report Template Structure

Module ID	Module Name	Content Description	Data Source
1	Basic Earthquake Information	Earthquake time, epicenter location, magnitude, depth, affected area, etc.	Meteorological system, seismic monitoring system
2	Affected Grid Equipment Summary	Number and preliminary damage status of substations, lines, towers, etc.	Asset platform, inspection system
3	Image Analysis Section	Key image features: waveform characteristics, protection actions, trip sequence	Relay waveform images, device action logs
4	Emergency Response Process	Emergency actions, including plan activation, personnel dispatch, and relay adjustments	Emergency system, manual input, OMS
5	Fault Cause Analysis	Analysis of earthquake-induced faults, assessment of protection logic correctness	Model-generated + field extraction
6	Event Impact and Loss Assessment	Power outage coverage, proportion of critical users, equipment damage list, loss estimate	OCS system, marketing system, and manual supplements
7	Recommendations and Summary	Overall response evaluation, improvement suggestions, concluding remarks	Template-based paragraphs (supporting manual edits)

It is modular, and therefore both reusable and scalable. Each module is considered as an individual paragraph generation. The system automatically produces and creates content for every module based on the given data. It guarantees that there will be no differences in the writing style, structural logic, and terminology used throughout the entire report.

5.1.2. Data Description

To fit in with generating an automated report. These are the kinds of data which are entered by the system. There's text, and then there are images. There are two kinds of paragraphs.

- Text data mainly comes from the power dispatch system, asset management platform, and emergency reporting system. These data include important information like time, place, machine, and the kind of action.
- Image data mainly obtained from fault waveform records contains graphical information about the current/voltage anomaly and the sequence of protection actions due to a seismic event. These are the basis for the generation of image summaries and guidance of paragraphs.

(1) To standardize the processing workflow, we use field extraction and naming rules matching to identify data elements, and then generate corresponding generation tasks. Table 12 gives a sample data table:

Table 12. Sample Text Field Data

Field Category	Field Name	Example Value
Time Field	Earthquake Time	2024-11-10 09:35:07
Time Field	Fault Occurrence Time	2024-11-10 09:37:12
Spatial Field	Substation	XX Substation
Equipment Field	Action Device	Transformer No.1
Voltage Level	Voltage Rating	220 kV
Action Type	Protection Action Type	Differential Protection Trip

(2) The associated image files follow the same naming rules as the extracted fields so that the system can match the fields to the images. For example:

Image Summary Description: It can be seen from the waveform that at 09:37, the current of phase A has suddenly increased, and the differential protection device has been triggered to cause a trip.

All text and image data are sourced from real operational logs, and after anonymization and standardization, they are sent into the system, and are compatible with the field parsing module and the template generation mechanism. Text and image data combined with a paragraph template is a good base for downstream automated content generation.

5.2. System Architecture Design

To enable automated generation and standardized documentation of fault reports during earthquake emergencies, we utilize the aforementioned software, which employs a distributed architecture comprising a front-end interaction layer, an intermediate processing layer, a language generation service layer, and a report production layer. Each layer is decoupled with distinct responsibilities and interfaces, facilitating engineering implementation and system evolution. The overall structure is illustrated in Fig. 4.

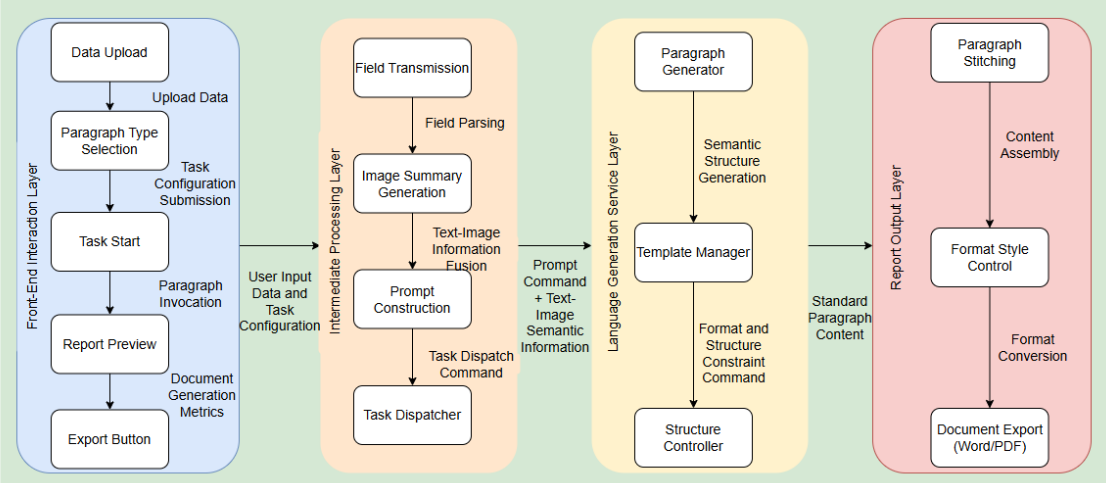

Figure 4. Software Architecture of the Automated Report Generation System

(1) Front-End Interaction Layer

This layer interacts with the end users and performs important functions like data upload, paragraph type selection, task start, report preview, and export button. Support for upload of structured text file like.txt and.docx and images. Users can configure the generation task, choose paragraph type, and review the generated result on the paragraph level. Input all passes to the downstream unit via the standard API to keep a consistent input and standard operation.

(2) Intermediate Processing Layer

Parses the uploaded data and preprocess it, and then makes the data fit for being generated. It has the following functional module.

- Field transmission: fault occurrence time, substation name, equipment ID, voltage level, protection type;
- Image summary generation: get the descriptive content in the name of image files and rules of fusion between image and text;
- Prompt Construction: creates task-oriented prompts according to the target paragraph type;
- Task Dispatcher: organizes generation task into a queue according to the structure of the paragraph and manage multi-paragraph dispatch.

All components can be plugged in, so if we want to adapt this for different scenarios, it is easy, and we can plug in other kinds of paragraphs or inputs for our models as well.

(3) Language Generation Service Layer

This layer integrates a Language Model API with actual paragraph generation capabilities. It accommodates a wide variety of paragraph types and autonomously selects appropriate templates, prompts, and input formats. The template manager handles formatting constraints, while the structure controller ensures adherence to structural requirements. It facilitates remote API calls and on-premise model deployment across diverse computing and secure environments.

(4) Report Output Layer

This layer is for Paragraph Stitching, Format Style Control, and Document Export. The main functionality of this is:

- Content Assembly: Stitches paragraph content into a full report following a given structure sequence by sequence;
- Format Conversion: Unifies font styles, headings, figures/charts;
- Document Export: Can output in.docx and.pdf formats, which are suitable for saving, checking, and showing.

The last output meets the standard of the local power grid emergency report and can be stored long-term and manually.

To improve the stability of the system and the coordination between the layers, use RESTful APIs for modules to communicate. Scheduling tasks via async queues. Field parsing result cache, generated content cache, etc., are all stored locally. This can be used for tracking status, parallel tasks, and continuing workflow.

5.3. System Development and Implementation

5.3.1. Development Environment and Technology Stack

The system employs a modular, front-end and back-end separated design to facilitate comprehensive intelligent processing, spanning from data intake to document generation. The complete development environment and technological framework are outlined in Table 13, with principal elements detailed below:

(1) Frontend Development: The user interface is developed using Vue.js, with Element UI employed for components such as forms, dropdowns, and buttons. The interface also supports interactive functionalities, including task configuration, field management, and previewing imported paragraphs.

(2) Backend Services: The backend system is developed using Python, with the Flask framework delivering a RESTful API service. This backend component handles field parsing, creates image summaries, builds prompts, and manages task scheduling. Communication with the frontend is facilitated via standard REST APIs.

(3) Database System: MySQL for data storage and management, contains a structure with fields value, paragraph template, generation record, and system log. Supports field to template matching, report traceability, and version control.

(4) Model Service Interface: It interfaces with OpenAI's extensive language models, such as the GPT series, via an API. Paragraph generation is accomplished through the application of prompts and image summarization. Templates and structure-controls are employed to manage layout and terminology.

(5) Deployment and Runtime Environment: The system operates on a Linux server. Nginx manages frontend resource requests, while Gunicorn provides WSGI support for the backend service. This setup offers robust concurrency and cross-platform compatibility.

Table 13. Overview of System Development Environment and Technology Stack

Module	Technology Stack	Description
Frontend UI	Vue.js + Element UI	Interactive configuration and visualization pages
Backend Service	Python + Flask	Field parsing and task scheduling APIs
Data Management	MySQL	Storage for fields, templates, and generation logs
Model Service	OpenAI GPT API	Multi-paragraph natural language generation
Communication	RESTful API	Interaction between frontend, backend, and modules
Deployment	Nginx + Gunicorn + Linux	Stable, efficient runtime platform

By leveraging this technology stack, the system can attain high generation efficiency and strong maintainability, along with the capacity to be engineered for scalable real-world deployment.

5.3.2 System Interface and Generation Demonstration

The system featured a clean, intuitive Graphical User Interface (GUI) that provided straightforward access to all essential functions, enabling non-technical users to apply them in practical situations. This was demonstrated using an actual seismic event that occurred in a specific county within a province on December 19, 2024. Utilizing the system, I created a standardized emergency report for the earthquake.

By integrating seismic data, a structured descriptive section, and multimodal inputs (text and images), the system generated a comprehensive report. This document included key sections covering earthquake details, the response procedure, equipment conditions, and an assessment of the resulting damage.

As illustrated in Fig. 5, the system's primary interface is organized into three main zones: the Data Import Area, the Template Selection Area, and the Generation Control Area. Users can easily configure settings and produce reports according to the parameters of a given task.

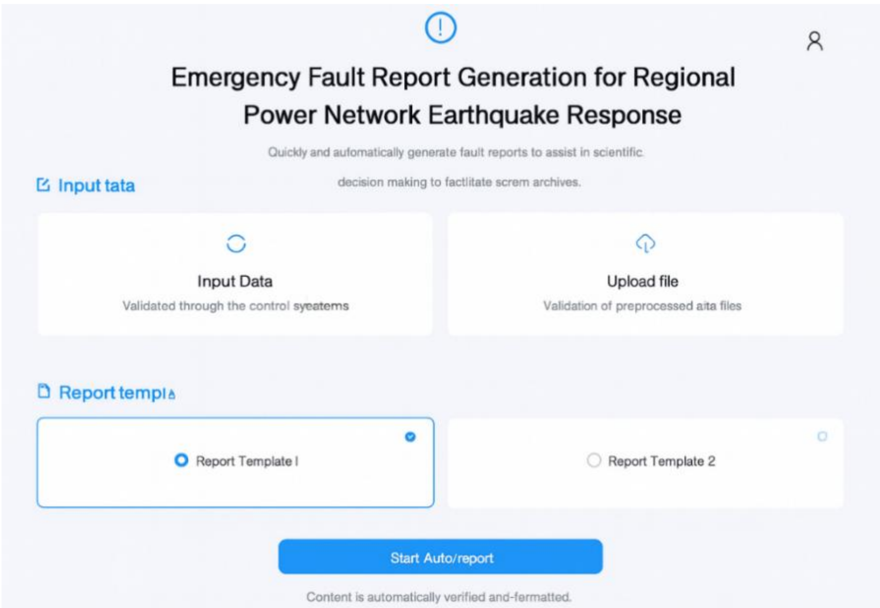

Figure 5. User Interface of the Fault Report Generation System

Fig. 6 through 8 display selected pages from the system-generated report. These pages feature the report cover, fundamental event details, the fault resolution procedure, equipment inspection findings, and the layout of the seismic monitoring stations.

Regional Power Grid Earthquake Emergency Incident Analysis Report

I. Basic Information

(1) Earthquake Details

- **Time of Occurrence**: December 19, 2024, 00:46
- **Location**: Binchuan County, Dali Prefecture, Yunnan Province (25.96°N, 100.50°E)
- **Magnitude**: 3.8
- **Focal Depth**: 10 km
- **Affected Area**: Within 20 km radius of the epicenter, involving substations, transmission lines, towers, etc.

(2) Power Grid Equipment and Personnel Overview

- **Affected Substations**: 1 (Binchuan County Urban 110kV Substation)
- **Transmission Lines**: 5 lines (including 110kV and 35kV)
- **Transmission Towers**: Approximately 120
- **Distribution Transformers**: Approximately 150
- **Ongoing Construction Tasks**: None
- **Construction Personnel**: None
- **Offices and Staff**: None

Figure 6. Earthquake Information and Equipment Statistics

II. Incident Timeline

(1) Earthquake Alert and Notification

- **00:46:07**: China Earthquake Networks Center issued an automatic report (Magnitude 3.8, Depth 10 km). (CCTV)
- **00:49**: Official parameters released via WeChat/website. (Sina Finance)
- **00:50**: Earthquake alert SMS received by prefecture and county emergency management departments and forwarded to [redacted]
- **00:51**: [redacted] classified the incident as "Level IV Monitoring"; [redacted] entered emergency monitoring process (Level IV response threshold is M ≥ 4.5; hence, monitoring only, no escalation).

(2) Preliminary Inspection of Power Grid Facilities

Time	Action	Personnel/Tools	Result
00:55	Remote SCADA, PMU, protection channel checks	4 dispatchers on duty	Voltage, current, frequency all normal; no tripping observed
01:10	Dispatched 2 special patrol teams	8 people, 2 vehicles, 2 UAVs	Inspected lines and substations within 20 km radius
01:45	UAV infrared imaging returned	—	No tower tilt or insulator damage
02:30	Equipment zero-point check	6 substation staff	All temperature points ≤ 45°C; line tension normal

Figure 7. Emergency Response Actions and On-Site Inspection Results

III. Emergency Response Status

(1) Emergency Response Actions

- At 00:52, [redacted] emergency commander issued the *Rapid Earthquake Response Directive*, requesting inspection within 2 hours and report submission; emergency personnel from prefecture and county joined dispatch center.
- By 01:20, all 110/35kV substations switched to "manned operation" mode.

(2) Equipment Protection and Secondary System Checks

- **Protection Devices**: 24 sets at 110kV Binchuan Substation and 11 at 35kV Lijiao Substation; all online, 0 defects.
- **Communication Channels**: 2 OPGW lines showed transient optical power fluctuation (<2 dB), auto-recovered by 03:05.
- **Auto-Reclosing**: Original values maintained; no temporary lockout due to low magnitude.

(3) Information Reporting

- **02:10**: *First Earthquake Grid Info Bulletin* submitted to [redacted], Provincial Energy Bureau, and Prefecture Government.
- **04:25**: *Final Report* confirmed "grid operating normally, no user power outage".

Figure 8. Earthquake Emergency Handling and Outcome Analysis

The generated report reconstructs the event based on real earthquake parameters and presents the results in a consistent, standardized writing style suitable for emergency communication and archiving.

5.4. Performance Evaluation and Comparative Analysis

To assess the system's effectiveness in actual earthquake emergency scenarios, five key criteria were selected to evaluate its performance: the extent of its field coverage, the correctness of its word order, the consistency of its terminology, the logical soundness of its reasoning, and its average processing time.

Field Coverage examines whether the generated paragraph comprehensively addresses all primary structured input fields.

Syntactic Coherence evaluates whether the paragraph flows smoothly, maintains grammatical accuracy, and possesses a natural structure.

Terminological Consistency checks if the output employs standard power grid terminology and avoids unclear or ambiguous phrasing.

Reasoning Coherence assesses the logical connections between multiple paragraphs and whether these relationships are clearly expressed.

Average Generation Time refers to the total duration needed to transform input data into a complete report.

In this research, expert evaluators conducted manual assessments, comparing system-generated reports with manually written ones to determine their operational effectiveness and practical utility. The comparative outcomes are presented in Table 14.

Table 14. Comparison of System-Generated vs. Manually Written Reports

Evaluation Dimension	Field Coverage	Syntactic Coherence	Terminological Consistency	Reasoning Coherence	Avg. Generation Time(hh:mm:ss)
Manual Report Writing	77.4%	82.0	84.0	72.0	03:00:00
System-Generated Report	100%	86.0	92.0	84.0	00:02:15

The assessments indicate that reports generated by the system outperformed manually written reports across all measured criteria. Field coverage is now fully comprehensive. A significant improvement was observed compared to manual writing. The accuracy of terminology and reasoning increased by 8% and 12% respectively, demonstrating stronger alignment with industry standards and more coherent logical progression.

Regarding efficiency: the system requires an average of 2 minutes and 15 seconds to produce a report, a task that typically takes an individual 180 minutes to complete independently. This means the system can finalize the work in just one-eightieth of the time a person would need, dramatically accelerating the report creation process and substantially aiding the demand for rapid, high-quality reports during earthquake emergencies.

6. Conclusion

This research addresses the critical demand for rapid emergency response following an earthquake-induced power grid failure, introducing an intelligent fault report generation technique that combines a multimodal language model, structured information filling, and prompt-based control. An automated system framework has been developed, encompassing data processing, image analysis, content generation, and output formatting. The proposed method tackles major drawbacks of manual report creation, such as lengthy preparation periods, high subjectivity, and inconsistent terminology. Experimental evaluation using public datasets from USGS and PNNL demonstrates:

(1) Significant enhancement in the field coverage and terminology of fault reports. Field coverage reached 100%, ensuring comprehensive output.

(2) The structured information filling approach, integrated with image summarization and prompt-guided controlled generation, supports diverse paragraph types (e.g., event description, image interpretation, fault analysis). The generated text achieved a BLEU-4 score of 0.91 and a ROUGE-L score of 0.90 on the test set, outperforming a template-filling baseline.

(3) A layered software architecture is employed, with decoupled frontend, intermediary, language generation, and output formatting components. The design clearly illustrates the input-output relationships and coordination among these modules, facilitating easier maintenance and scalability. Using real-world data, the average report generation time was 2 minutes 15 seconds, representing an 80-fold speed increase compared to manual drafting, which typically requires nearly 180 minutes. This meets the practical requirement for swift emergency report production and holds strong potential for real-world power grid emergency scenarios.

The proposed approach demonstrates reasonable efficiency and report quality. Further improvements are needed regarding model stability, generalization capability, and text consistency for lengthy reports. Future work will explore knowledge-enhanced generation methods tailored to the power domain and incorporate human-in-the-loop mechanisms to enhance adaptability and robustness for practical applications.

Acknowledgements

This research was funded by the Key Technologies for AI-Based Auxiliary Decision-Making in Relay Protection Operation and Watchkeeping (Task 3: Digital and Intelligent Technologies for Protection Operation and Watchkeeping to Enhance Power-Supply Reliability in Regional Grids) under Grant YNKJXM20240572.

References

[1] Tonnelier M, Delforge D, Below R, Munguía JAT, Saegerman C, Wathelet V, et al. What makes an epidemic a disaster: the future of epidemics within the EM-DAT International Disaster Database. BMC Public Health. 2025;25(1):21.

[2] Amaratunga D, Anzellini V, Guadagno L, Hagen JS, Komac B, Krausmann E, et al. United Nations office for disaster risk reduction regional assessment report on disaster risk reduction 2023: Europe and Central Asia. 2023.

[3] Mate A, Hagan T, Cotilla-Sanchez E, Brekken TK, Von Jouanne A, editors. Impacts of earthquakes on electrical grid resilience. 2021 IEEE/IAS 57th Industrial and Commercial Power Systems Technical Conference (I&CPS); 2021: IEEE.

[4] Hsu C-W, Mostafavi A. Untangling the relationship between power outage and population activity recovery in disasters. Resilient Cities and Structures. 2024;3(3):53-64.

[5] Yoo T-H, Park H. Modeling of power system resilience during a catastrophic disaster and application of the model. IEEE Access. 2024;12:81550-66.

[6] Işık İ, Gol EA. Field teams coordination for earthquake-damaged distribution system energization. Reliability Engineering & System Safety. 2024;245:110050.

[7] Arpali OY, Yilmaz UC, Erkal BG, Gol EA, Gol M. MDP based real time restoration for earthquake damaged active distribution systems. Electric power systems research. 2023;218:109230.

[8] Mille S, Casamayor G, Grivolla J, Shvets AV, Wanner L, editors. Automatic Multilingual Incident Report Generation for Crisis Management. ISCRAM; 2022.

[9] Xu F, Ma J, Li N, Cheng JC. Large language model applications in disaster management: An interdisciplinary review. International Journal of Disaster Risk Reduction. 2025;127:105642.

[10] Hadid A, Chakraborty T, Busby D. When geoscience meets generative AI and large language models: Foundations, trends, and future challenges. Expert Systems. 2024;41(10):e13654.

[11] Estêvão JM. Effectiveness of generative ai for post-earthquake damage assessment. Buildings. 2024;14(10):3255.

[12] Jiang Y, Wang J, Shen X, Dai K. Large language model for post-earthquake structural damage assessment of buildings. Computer-Aided Civil and Infrastructure Engineering. 2025;40(31):6324-42.

[13] Cheng Y, Zhao H, Zhou X, Zhao J, Cao Y, Yang C, et al. A large language model for advanced power dispatch. Scientific Reports. 2025;15(1):8925.

[14] Pu H, Yang X, Shi Z, Jin N, editors. Automated inspection report generation using multimodal large language models and set-of-mark prompting. ISARC Proceedings of the International Symposium on Automation and Robotics in Construction; 2024: IAARC Publications.

[15] Zhang R, El-Gohary N. Natural language generation and deep learning for intelligent building codes. Advanced Engineering Informatics. 2022;52:101557.

[16] Lin Y, Ruan T, Liu J, Wang H. A survey on neural data-to-text generation. IEEE Transactions on Knowledge and Data Engineering. 2023;36(4):1431-49.

[17] Sharma M, Gogineni AK, Ramakrishnan N. Neural methods for data-to-text generation. ACM Transactions on Intelligent Systems and Technology. 2024;15(5):1-46.

[18] Devlin J, Chang M-W, Lee K, Toutanova K, editors. Bert: Pre-training of deep bidirectional transformers for language

understanding. Proceedings of the 2019 conference of the North American Section of the association for computational linguistics: human language technologies, volume 1 (long and short papers); 2019.
[19] Weld H, Huang X, Long S, Poon J, Han SC. A survey of joint intent detection and slot filling models in natural language understanding. ACM Computing Surveys. 2022;55(8):1-38.
[20] Xiao H, Zhou F, Liu X, Liu T, Li Z, Liu X, et al. A comprehensive survey of large language models and multimodal large language models in medicine. Information Fusion. 2025;117:102888.
[21] Dathathri S, Madotto A, Lan J, Hung J, Frank E, Molino P, et al. Plug and play language models: A simple approach to controlled text generation. arXiv preprint arXiv:191202164. 2019.
[22] Geng S, Döner B, Wendler C, Josifoski M, West R, editors. Sketch-guided constrained decoding for boosting blackbox large language models without logit access. Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers); 2024.
[23] Park K, Wang J, Berg-Kirkpatrick T, Polikarpova N, D'Antoni L. Grammar-aligned decoding. Advances in Neural Information Processing Systems. 2024;37:24547-68.
[24] Survey USG. ANSS Comprehensive Catalog (ComCat) Earthquake Catalog 2025 [Available from: https://earthquake.usgs.gov/fdsnws/event/1.
[25] She B, Adetola V, Yun JY. Event-correlated Outage Dataset in America. Open Energy Data Initiative (OEDI). Pacific Northwest National Laboratory; 2024.