AI - Enhanced System for Template Management and Generation

Naveen Koka

Email: *na.koka[at]outlook. com*

Abstract: ***Our advanced report generation tool is designed to transform complex and unstructured log data into comprehensive and actionable reports, leveraging cutting - edge technologies like Large Language Models (LLMs), vector databases, and HTML - based WYSIWYG editors. This solution addresses the challenges of traditional report generation platforms by automating data extraction, standardization, and integration into customizable templates. The tool facilitates the creation of dynamic, visually appealing reports that meet specific business needs, significantly enhancing the efficiency and accuracy of the reporting process. During setup, the tool embeds a WYSIWYG editor, allowing users to design report templates with ease. These templates incorporate placeholder tags representing data attributes, which the LLMs will later populate with actual values. The configuration metadata, including template names, descriptions, and data sources, is stored in a relational database. As log data is generated, it is segmented into smaller records, with each record's semantic meaning captured through LLM - generated embeddings stored in a vector database. This structure ensures efficient data management and retrieval. At runtime, an event triggers the report generation process, selecting the appropriate template based on predefined criteria. The system retrieves the relevant metadata, performs a vector database search to obtain the necessary records, and uses the LLM to extract and populate the template tags with the latest data. The result is a high - quality, data - driven report ready for conversion to PDF and distribution. This robust and flexible approach enables industries like healthcare and manufacturing to harness their vast log data, generating insightful reports that drive better decision - making and operational improvements.***

Keywords: AI, LLM, Templating Tool, Lang chain, Data logs

1. Introduction

In today's data - driven world, the ability to generate insightful reports from extensive and complex data sources is crucial across various industries. Our advanced report generation tool harnesses the power of Large Language Models (LLMs), vector databases, and user - friendly WYSIWYG editors to streamline this process. Designed to handle gigabytes of log data efficiently, the tool transforms raw data into polished, customizable reports. Whether in the medical industry, where patient records and operational metrics need to be analyzed, or in manufacturing, where production logs and quality control data are critical, our solution provides a seamless way to extract, process, and present data. By automating these tasks, businesses can save time, reduce errors, and gain valuable insights to drive decision - making and operational excellence.

2. Problem Statement

Generating reports is crucial for understanding a business's current status and making informed decisions. Traditional monolithic report generation platforms handle billions of data points, producing comprehensive reports. However, these platforms tend to be complex and cumbersome, often requiring significant time and effort to manage and extract meaningful insights.

In addition to traditional reports, there are reports designed to document completed work for customers, such as invoices. These customer - facing reports are essential for maintaining transparency and ensuring that clients receive accurate records of the services provided. Generating such reports typically involves standardized data formats, making the process straightforward and consistent.

A major challenge arises when dealing with logs generated from systems that lack standardized data formats. Creating reports from these logs can be particularly difficult, often requiring substantial investment of time and resources. This process needs to be repeated for each type of log, complicating report generation and making it a resource - intensive task.

3. Solution

To address these challenges, we will leverage the capabilities of Large Language Models (LLMs) to create a versatile tool capable of handling diverse reporting needs. This tool aims to simplify the process of generating reports from complex and unstructured data sources. By utilizing LLMs, we can automate the extraction of data from logs, standardize it, and incorporate it into customizable report templates, thus reducing the time and resources required for report generation.

The setup will include WYSIWYG (What You See Is What You Get) tools for creating report templates. These user - friendly tools will allow users to design report layouts easily, specifying where and how the extracted data should be displayed. Once the template is created, LLMs will process the log data, extract relevant information, and populate the template accordingly. This approach ensures that even data from unstructured sources can be seamlessly integrated into professional - looking reports.

Finally, the tool will have the capability to convert the completed reports into PDF format, making it easy to share and distribute. By automating the entire process—from data extraction and template application to PDF conversion—this tool will streamline report generation, enhance accuracy, and significantly reduce the manual effort involved. This innovative solution will empower businesses to efficiently

DOI: https://dx.doi.org/10.21275/SR24707220200

generate various types of reports, regardless of the complexity of the underlying data.

3.1 Definitions

3.1.1 Editor

The WYSIWYG (What You See Is What You Get) tool is a crucial component of this solution, enabling users to design the report layout with ease and flexibility. We recommend using an HTML - based WYSIWYG editor, which allows for the creation of richly formatted and visually appealing reports. By leveraging HTML, users can design reports with free - flow formatting, incorporating various styles, fonts, and multimedia elements to enhance the overall presentation.

An HTML - based WYSIWYG tool provides a user - friendly interface where users can drag and drop elements, format text, and arrange data fields effortlessly. This approach not only simplifies the report design process but also ensures that the final output is both professional and customizable to meet specific business needs. The ability to create intricate and detailed report layouts will empower users to convey information more effectively, making the reports not only informative but also visually engaging.

3.1.2 Template Tags

Template tags play a vital role in our report generation process. These tags act as placeholders within the report template, marking the spots where actual data will be inserted. By embedding tags into the template, users can design the structure and layout of the report without worrying about the specific data content at that stage. During the data processing phase, these tags will be replaced with real values extracted from logs or other data sources, ensuring that the final report is both accurate and dynamic.

The use of template tags allows for a high degree of customization and flexibility. Users can place tags wherever data needs to appear, whether it's text, numbers, dates, or even more complex data structures. This approach makes it possible to create highly detailed and specific reports tailored to various business needs. By defining these tags clearly, the tool can seamlessly integrate the extracted data into the designed template, maintaining the intended format and layout.

To facilitate this process, our WYSIWYG tool will support the easy insertion and management of template tags. Users will be able to define tags with a simple interface, ensuring that each placeholder is correctly positioned and labeled. Once the template design is complete, the LLMs will take over, extracting the relevant data, matching it with the corresponding tags, and replacing the tags with actual values. The result is a polished, data - driven report that is ready for conversion to PDF and distribution, significantly enhancing the efficiency and effectiveness of the reporting process.

3.1.3 LLM

Choosing the right LLM library is crucial for the success of our report generation tool. By using a versatile library like Langchain, we gain the flexibility to easily switch between different LLM APIs as needed. This adaptability is key to maintaining the tool's relevance and efficiency as new and improved LLMs become available.

Langchain provides a unified framework for integrating various LLMs, allowing us to leverage the strengths of different models without being tied to a single provider. This means we can start with one LLM and, if necessary, transition to another with minimal disruption to our system. The ability to replace LLM APIs seamlessly ensures that our tool remains up - to - date with the latest advancements in language model technology, providing the best possible performance for data extraction and report generation.

Moreover, using Langchain allows us to optimize the tool for different use cases and data sources. Depending on the specific requirements of a report, we can select the most appropriate LLM for the task, ensuring high accuracy and efficiency. Langchain's modularity and compatibility with various LLMs make it an ideal choice for our needs, offering a robust foundation for building a flexible and powerful report generation solution.

By integrating Langchain, we can ensure that our tool remains adaptable and future - proof, capable of leveraging the best available LLMs for extracting and processing data. This strategic choice will enhance our ability to deliver high - quality, dynamic reports, meeting the evolving needs of businesses while maintaining the flexibility to incorporate new technologies as they emerge.

3.1.4 Vector Database

Incorporating a vector database into our report generation tool is a strategic decision that significantly enhances our data management and retrieval capabilities. A vector database stores data along with embeddings generated from a Large Language Model (LLM), which are essentially high - dimensional representations of the data's semantic meaning. This approach offers several key advantages, making it an integral part of our solution.

Firstly, using embeddings allows for more efficient and accurate data retrieval. Unlike traditional databases that rely on exact keyword matches, a vector database can perform searches based on the semantic similarity of data. This means that even if the query terms do not exactly match the stored data, the system can still retrieve relevant results based on their contextual meaning. This capability is particularly useful for handling unstructured data from logs, where the exact wording and format can vary widely.

4. Application Implementation

Below are the important steps to implement a template generation application.

4.1 Data preparation

Data preparation is a crucial step in ensuring the effectiveness of our report generation tool, especially when dealing with large volumes of log data. As soon as the log data is generated, it needs to be processed and stored efficiently in the vector database. Given the sheer size of the logs, often reaching gigabytes, we must convert these documents into multiple smaller records and generate embeddings for each row. This systematic approach ensures that the data is manageable, searchable, and ready for seamless report generation.

The first step involves breaking down the large log files into smaller, more granular records. This can be achieved by parsing the logs and extracting meaningful chunks of data, such as individual log entries or specific events. By segmenting the logs into smaller pieces, we can handle the data more effectively, making it easier to process and store in the vector database. Each record should represent a coherent unit of information that can be independently analyzed and used in the report generation process.

Once the data is segmented, the next step is to generate embeddings for each record using an LLM. These embeddings capture the semantic meaning of the data, transforming it into high - dimensional vectors that can be stored in the vector database. This process involves feeding each record into the LLM, which then produces a corresponding embedding that encapsulates the contextual information within the data. By doing so, we ensure that the vector database is populated with rich, meaningful representations of the log data.

Storing the embeddings in the vector database allows for efficient and effective data retrieval. When generating reports, the LLM can perform semantic searches to find relevant records based on their embeddings, even if the exact terms in the query do not match the data. This capability significantly enhances the flexibility and accuracy of our report generation tool, enabling it to produce comprehensive and insightful reports from large and complex log datasets.

4.2 Setup

The setup phase is critical for configuring and managing templates within our report generation tool. Here, we will save all relevant configurations, including metadata such as template name, description, and data source. This metadata will be stored in a standard relational database, ensuring easy access and management.

First, we embed the HTML - based WYSIWYG editor into the application, allowing users to design and customize their reports seamlessly. Users can create a report layout, incorporating various design elements and placeholders for data attributes. These attributes, represented as template tags (e. g., {{status}}), will be used to dynamically insert data into the report during runtime. The editor should provide an intuitive interface for users to easily insert and manage these template tags, ensuring that they can effectively design the report layout without needing advanced technical skills.

Next, the system will extract all template tags before saving the metadata. This extraction process involves scanning the designed template for placeholders and cataloging them. This ensures that all necessary tags are identified and recorded, allowing for efficient data binding during report generation. The extracted tags will be stored alongside the template metadata in the relational database, providing a comprehensive configuration that can be easily accessed and reused.

During runtime, the stored metadata and template tags will be used to generate the report. The LLM will retrieve the relevant data from the vector database, match it with the corresponding template tags, and populate the template accordingly. This approach ensures that the report is dynamically generated based on the latest data, maintaining accuracy and relevance.

4.3 Runtime

During runtime, the report generation process is initiated by an event trigger. This event can be based on specific criteria, determining which template to invoke based on the data source and the nature of the problem at hand. This dynamic selection ensures that the most appropriate template is used for each report generation task.

Once the correct template is identified, the system retrieves its associated metadata. This metadata is then passed to the vector database to perform a search, which filters and returns a definitive set of relevant records. These records contain the data needed for populating the report. The next step involves sending these records to the LLM, which will process the data and extract the required information.

To ensure the LLM provides the necessary output, we need to supply it with a clear and precise system prompt. This prompt should guide the LLM in extracting the template tags and generating the corresponding values. An example of an effective system prompt is as follows:

"Based on the data, extract all the template tags and create an output in the format below. If there are multiple values for the same tag, take the latest one. The input provided is in the format where the template tags are what we need to extract from the data. For instance, the input may look like this: {'template_tags': [tag1, tag2], 'data': [. . .]}. The expected output should be: {'<template tag>': 'actual value'}. "

This prompt ensures that the LLM understands the task and produces the desired output format. By clearly defining the input structure and the expected output, we can guide the LLM to accurately extract the values for each template tag from the data.

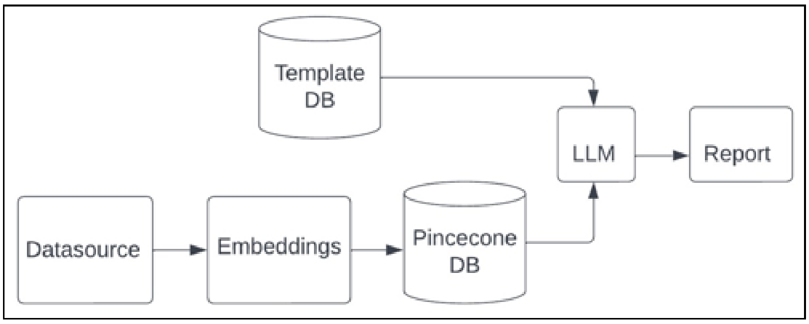

Figure 1: Template Generation

5. Summary

In summary, the runtime process involves the following steps:

1) **Event Trigger**: Initiate the report generation based on specific criteria.
2) **Template Selection**: Retrieve the appropriate template and its metadata based on the data source.
3) **Vector Database Search**: Use the metadata to perform a search in the vector database, retrieving a definitive set of records.
4) **Data Processing with LLM**: Send the records to the LLM with a clear system prompt to extract the template tags and generate the required values.
5) **Output Generation**: The LLM provides the extracted values, which are then used to populate the report template.

6. Uses

Logs are available across various industries, providing valuable insights and data that can be leveraged for report generation. Below are two critical industry spaces where our report generation tool can be particularly beneficial

6.1 Medical Industry

In healthcare, logs are generated from numerous sources, including patient records, medical devices, and hospital management systems. These logs contain vital information such as patient diagnoses, treatment histories, and operational metrics. Efficiently generating reports from this data can help in monitoring patient outcomes, ensuring compliance with regulatory standards, and improving overall hospital management. By utilizing our tool, healthcare providers can seamlessly convert complex log data into comprehensive reports, aiding in better patient care and informed decision - making.

6.2 Manufacturing Industry

In the manufacturing sector, logs are produced from production lines, machinery, and quality control systems. These logs track various metrics, including equipment performance, production rates, and defect occurrences. Generating detailed reports from this data is crucial for maintaining operational efficiency, identifying areas for improvement, and ensuring product quality. Our tool can automate the extraction and analysis of log data, creating insightful reports that help manufacturers optimize processes, reduce downtime, and enhance productivity.

7. Conclusion

In conclusion, our comprehensive report generation tool leverages advanced technologies like LLMs, vector databases, and HTML - based WYSIWYG editors to handle diverse and complex data sources efficiently. The setup involves embedding the editor for easy report design, managing template tags, and storing configuration metadata in a relational database. During runtime, an event triggers the report generation, selecting the appropriate template based on the data source. The system retrieves metadata, performs a vector database search, and processes the data using LLMs with precise prompts to extract and populate template tags. This approach ensures the creation of high - quality, data - driven reports, enhancing business decision - making and operational efficiency.

References

[1] S. He, J. Zhu, P. He and M. R. Lyu, "Experience Report: System Log Analysis for Anomaly Detection, " 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), Ottawa, ON, Canada, 2016, pp.207 - 218, doi: 10.1109/ISSRE.2016.21.

[2] Du, Min & Li, Feifei & Zheng, Guineng & Srikumar, Vivek. (2017). DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning.1285 - 1298.10.1145/3133956.3134015.

[3] Revanth Gangi Reddy, Daniel Lee, Yi R. Fung, Khanh Duy Nguyen, Qi Zeng, Manling Li, Ziqi Wang, Clare Voss, Heng Ji. (2023). SmartBook: AI - Assisted Situation Report Generation for Intelligence Analysts. https: //doi. org/10.48550/arXiv.2303.14337

[4] S. Kukreja, T. Kumar, V. Bharate, A. Purohit, A. Dasgupta and D. Guha, "Vector Databases and Vector Embeddings - Review, " 2023 International Workshop on Artificial Intelligence and Image Processing (IWAIIP), Yogyakarta, Indonesia, 2023, pp.231 - 236, doi: 10.1109/IWAIIP58158.2023.10462847.

[5] Patil, N. & Patel, S. & Lawand, S. D. . (2023). Research Paper On Artificial Intelligence And It's Applications. Journal of Advanced Zoology.44.10.53555/jaz.v44iS8.3544.

[6] Oyeniyi, Lawrence & Ugochukwu, Chinonye & Mhlongo, Noluthando. (2024). The influence of AI on financial reporting quality: A critical review and analysis. World Journal of Advanced Research and Reviews.22.679 - 694.10.30574/wjarr.2024.22.1.1157.

[7] Wahle, Jan & Ruas, Terry & Mohammad, Saif & Meuschke, Norman & Gipp, Bela. (2023). AI Usage

Paper ID: SR24707220200 DOI: https://dx.doi.org/10.21275/SR24707220200

Cards: Responsibly Reporting AI - generated Content.10.48550/arXiv.2303.03886.