Journal for Educators, Teachers and Trainers
The PROFESIOLab electronic, peer-reviewed, open-access Magazine

ISSN 1989-9572

DOI:10.47750/jett.2024.15.05.29

AI-Powered Server Log Management for Automated Error Resolution

Ayub Baig1, M. Sowmya2, G. Amulya2, K. Jayasaumya2

Journal for Educators, Teachers and Trainers, Vol.15(5)

https://jett.labosfor.com/

Date of Reception: 24 Oct 2024

Date of Revision: 20 Nov 2024

Date of Publication : 31 Dec 2024

Ayub Baig1, M. Sowmya2, G. Amulya2, K. Jayasaumya2 (2024). AI-Powered Server Log Management for Automated Error Resolution,Vol.15(5).296-308

Journal for Educators, Teachers and Trainers
The LabOSfor electronic, peer-reviewed, open-access Magazine

Journal for Educators, Teachers and Trainers, Vol. 15(5)

ISSN1989 –9572

https://jett.labosfor.com/

AI-Powered Server Log Management for Automated Error Resolution

Ayub Baig[1], M. Sowmya[2], G. Amulya[2], K. Jayasaumya[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Information Technology

[1,2]Malla Reddy Engineering College for Women (UGC-Autonomous), Maisammaguda, Hyderabad, 500100, Telangana.

ABSTRACT

Modern technology environments generate vast amounts of server logs, each potentially containing critical information about system errors. Traditional methods of resolving these errors typically involve time-consuming manual searches across multiple platforms—ranging from search engines like Google and Bing to various online forums—in hopes of finding the correct solution. This process often proves inefficient, as users must sift through extensive search results and compare inconsistent or irrelevant information, risking further errors and delays. In response, this research aims to develop an AI-powered server log management software that delivers accurate, automated solutions to errors by analyzing historical log data and corresponding resolutions. By consolidating server logs and training a predictive AI model, the proposed platform offers a one-stop solution capable of reducing the time, effort, and complexity currently associated with error resolution. Users simply input an error, and the system provides an intelligently derived, context-aware solution—eliminating the need for manual searches. In doing so, the platform streamlines workflows, reduces user frustration, and improves overall efficiency in managing complex technical issues in real-world environments.

Keywords: Server Log, Streamlining, Error Resolution, Burdensome, AI

1. INTRODUCTION

The growth of digital infrastructure in India has led to an exponential increase in server usage across industries. According to a report by NASSCOM, India's IT sector is a significant contributor to the global economy, with a revenue of over $194 billion in 2020. However, with this growth comes the challenge of managing complex server environments and resolving errors quickly. Traditional error resolution methods often require manual searches, leading to inefficiencies and frustration. As server infrastructures become more sophisticated, the need for automated solutions to handle error resolution has become increasingly critical. The AI-Driven Error Resolution Platform is an innovative solution designed to streamline error resolution by automating the process of identifying and predicting solutions to server errors. This platform leverages machine learning algorithms to analyze server logs and predict the most relevant solutions, reducing the time and effort required by users to resolve issues. Before the advent of machine learning, error resolution was predominantly manual. Users had to search through multiple online platforms, sifting through vast amounts of data to find possible solutions. This process was time-consuming, prone to errors, and often led to frustration due to the overwhelming volume of information and the lack of a guarantee that the correct solution would be found. The motivation behind this research stems from the growing complexity of server environments and the increasing frequency of errors that need resolution. As technology continues to evolve, traditional methods of error resolution are becoming less effective, necessitating the development of an intelligent system that can quickly and accurately predict solutions. This research aims to reduce user frustration, enhance productivity, and improve the efficiency of error resolution processes.

2. LITERATURE SURVEY

Agt et al. [1] presented an automated method for constructing a large semantic network of related terms, significantly improving domain-specific modelling by enhancing the organization and retrieval of relevant terms. Agt-Rickauer et al. [2] developed a system that automatically recommends related model elements during domain modelling, enhancing model creation efficiency and accuracy, and later introduced DoMoRe [3], a recommender system designed to streamline the domain modelling process.

Alspaugh et al. [4] explored the practices of professional data analysts, emphasizing the need for user-centric tools to support exploratory data analysis. Ángel et al. [5] focused on integrating heterogeneous information sources to improve the accuracy and relevance of domain models. Annett [6] examined digital inking technologies, offering insights into their potential to support more intuitive and efficient modelling. Arora et al. [7] proposed a method for extracting domain models from natural-language requirements to improve modelling consistency, and later introduced an active learning approach [8] to enhance the accuracy of automated domain model extraction.

Aßmann et al. [9] discussed the relationship between ontologies, meta-models, and the model-driven paradigm, providing foundational insights for complex software system development. Barat et al. [10] developed an actor-based simulation for supply chain management using reinforcement learning, extending their work with a digital twin [11] for analyzing complex business systems, and applying reinforcement learning [12] within a multi-agent simulation to develop adaptable supply chain control policies. They also used an agent-based digital twin [13] to explore interventions for controlling the COVID-19 pandemic.

Barriga et al. [14] proposed a system for automatic model repair using reinforcement learning, enhancing system reliability. Bikakis et al. [15] introduced the DR-Prolog tool suite for defeasible reasoning in the semantic web, supporting intelligent systems in handling uncertainty. Bill et al. [16] provided an overview of MoMoT, a tool for exploring transformation spaces in model-driven engineering. Black et al. [17] proposed voice-driven modelling using automated speech recognition to make software modelling more accessible. Bordea et al. [18] developed a methodology for extracting taxonomies from the Wikipedia knowledge graph, supporting context-aware taxonomy development. Brambilla et al. [19] explored crowdsourcing for shaping domain-specific languages, and also authored a comprehensive guide [20] on model-driven software engineering.

3. PROPOSED SYSTEM

The system is built using Django and incorporates various functionalities for both admin and user access. Admins can train the AI model using a dataset of errors and corresponding solutions. The AI model uses a TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer to process and transform error descriptions into numerical representations, which are then used to compare with known errors and predict the most likely solution. The solution prediction is based on the cosine similarity between the TF-IDF vectors of the input error and the errors in the database.

Error Search and Solution Prediction:

Users can search for solutions by entering an error description. The system cleans and processes the input text, compares it with pre-existing error data, and predicts the best matching solution. If the AI model identifies a match, it provides the solution along with a relevant Google search link to explore further. If the model is unable to predict a solution, users are encouraged to refine their query and try again.

User and Admin Interfaces:

The research includes various interface screens for users and admins, such as login, signup, and solution search pages. Admins have access to a dashboard where they can view registered users, train the AI model with new data, and monitor the overall system performance. Users can sign up, log in, and search for solutions, with personalized responses based on the AI model's predictions.

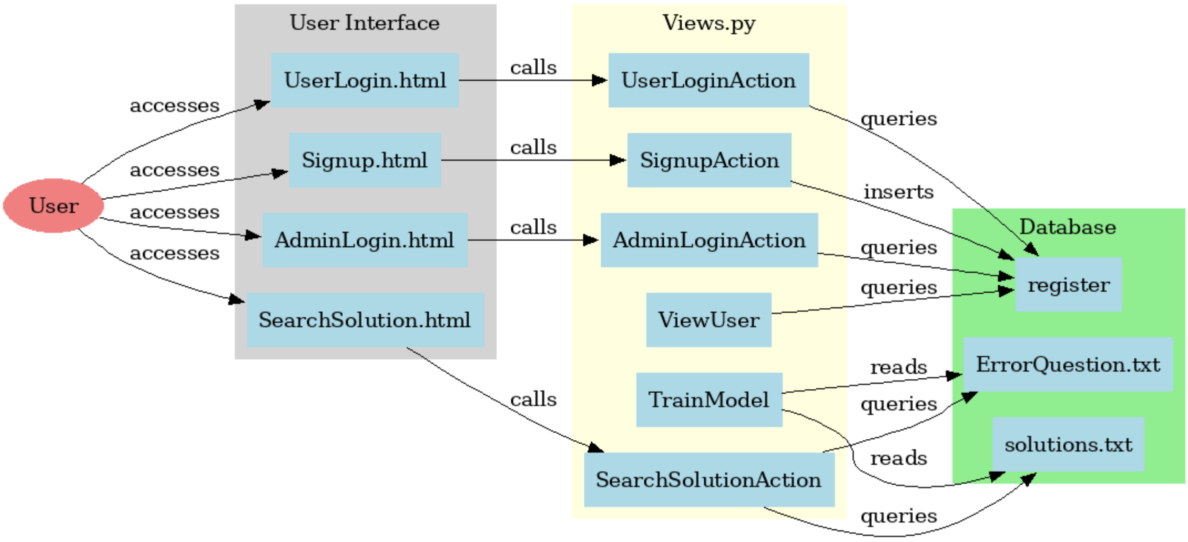

Figure 3.1 Architecture Diagram

Training the AI Model:

The AI model training process involves reading error and solution data from CSV files, processing and cleaning the text data, and then fitting the data to the TF-IDF vectorizer. The model is then capable of transforming new error descriptions into vectors and predicting the closest matching solution based on similarity scores.

Database Interaction:

The research uses a MySQL database to manage user data, including storing registration details and handling user authentication. This ensures secure and organized storage of user information, allowing for efficient user management and seamless interaction between the web application and the backend database.

3.1 Proposed Algorithm

NLTK

NLTK, or the Natural Language Toolkit, is a comprehensive suite of libraries and programs designed for natural language processing (NLP) in Python. It provides tools for text processing, such as tokenization, stemming, lemmatization, tagging, parsing, and more, making it a valuable resource for researchers, developers, and data scientists working with language data.

How It Works:

NLTK works by breaking down text into its fundamental components, like words and sentences, and then applying various NLP techniques to process and analyze the text. For example, tokenization splits text into individual words or sentences, while stemming reduces words to their root forms. These processes allow for more efficient text analysis, especially in large datasets.

The typical workflow in NLTK involves loading text data, preprocessing it using NLTK's tools, and then applying algorithms or models to extract meaningful insights. The toolkit includes resources like corpora (collections of text) and lexical resources (e.g., WordNet) that aid in understanding and analyzing the text.

Architecture:

NLTK's architecture is modular, consisting of several core components:

1. **Tokenization:** Splits text into words or sentences.
2. **Stemming:** Reduces words to their root form (e.g., "running" becomes "run").
3. **Lemmatization:** Converts words to their base or dictionary form.
4. **Tagging:** Assigns part-of-speech tags to words (e.g., noun, verb).
5. **Parsing:** Analyzes the grammatical structure of sentences.
6. **Chunking:** Identifies and classifies phrases within sentences.
7. **Corpora:** Provides access to various text corpora and lexical resources.

Each module can be used independently or combined to create complex NLP workflows.

Advantages:

1. **Comprehensive Toolkit:** NLTK offers a wide range of NLP tools and resources, making it a one-stop solution for text processing.
2. **Ease of Use:** It has a user-friendly interface with extensive documentation, making it accessible for beginners.
3. **Flexibility:** NLTK's modular design allows users to customize their NLP workflows.
4. **Educational Resources:** NLTK includes tutorials and example datasets, making it a great learning tool for NLP.

Explanation of the Predict Function:

In the research, the TrainModel function is designed to train a machine learning model for error resolution based on a dataset of errors and their corresponding solutions. Here's a brief explanation of the process:

1. **Data Preprocessing:** The function first reads error and solution data from text files. The error data is then cleaned using the cleanText function, which involves removing punctuation, stop words, and applying stemming and lemmatization (using NLTK tools). This preprocessing step ensures that the text is standardized and ready for vectorization.

2. **Vectorization:** The cleaned text data is transformed into numerical vectors using the TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer. TF-IDF measures the importance of a word in a document relative to a corpus, helping the model to focus on meaningful terms.

3. **Model Training:** The function then uses these TF-IDF vectors to train the model. It compares the vector of a test query (in this case, a sample error description) with the vectors of the errors in the dataset. The comparison is done using cosine similarity, which measures the angle between two vectors. The model predicts the solution corresponding to the error vector that has the highest similarity (accuracy) with the test query vector.

4. **Prediction:** The cosine similarity is calculated as the dot product of the two vectors divided by the product of their norms. This score indicates how similar the test query is to each error in the dataset. The error with the highest score is considered the closest match, and its corresponding solution is predicted by the model.

4. RESULTS AND DISCUSSION

The homepage of an AI-Driven Error Resolution Platform for Streamlined Solutions. The page features a red banner at the top, a navigation bar with links to "Home", "Admin Login Here", "User Login Here", and "New User Signup Here", and a prominent banner showcasing an illustration of a person standing in front of a large "AI" chip surrounded by clouds and data cubes. The homepage conveys the platform's focus on using AI to efficiently resolve errors and streamline solutions.

Fig 1: Home Page

Fig 2: Admin Login Page

The admin login screen of an AI-Driven Error Resolution Platform for Streamlined Solutions. The page features a red banner at the top, a navigation bar with links to "Home", "Admin Login Here", "User Login Here", and "New User Signup Here", and a prominent banner showcasing an illustration of a person standing in front of a large "AI" chip surrounded by clouds and data cubes. Below the navigation bar is the admin login form, which requires admins to enter their username and password. Once the correct credentials are entered, admins can log in to the system to access and manage various administrative functions related to the platform's operations.

Fig 3: Admin Home Page After Login

The admin homepage of an AI-Driven Error Resolution Platform for Streamlined Solutions. The page features a red banner at the top, a navigation bar with links to "Train AI Server Log Model", "View Users List", and "Logout", and a prominent banner showcasing an illustration of a person standing in front of a large "AI" chip surrounded by clouds and data cubes. Below the navigation bar is a welcome message that greets the admin, "Welcome admin". This homepage provides admins with access to key functionalities like training AI models using server logs and viewing a list of registered users.

Fig 4: Train AI Server Log Model

An AI-Driven Error Resolution Platform for Streamlined Solutions after successfully training an AI server log model. The page features a red banner at the top, a navigation bar with links to "Train AI Server Log Model", "View Users List", and "Logout", and a prominent banner showcasing an illustration of a person standing in front of a large "AI" chip surrounded by clouds and data cubes. Below the navigation bar is a message indicating that the AI training has been completed with an accuracy of 0.9999999999999998, suggesting a highly accurate model for error resolution. This successful training enables the platform to effectively analyze and resolve errors, providing streamlined solutions to users.

Fig 5: View Users List

The View Users screen of an AI-Driven Error Resolution Platform for Streamlined Solutions. The page features a red banner at the top, a navigation bar with links to "Train AI Server Log Model", "View Users List", and "Logout", and a prominent banner showcasing an illustration of a person standing in front of a large "AI" chip surrounded by clouds and data cubes. Below the navigation bar is the admin login form, which requires admins to enter their username and password. Once the correct credentials are entered, admins can log in to the system to access and manage various administrative functions related to the platform's operations.

Fig 6: New User Signup

The new user signup screen of an AI-Driven Error Resolution Platform for Streamlined Solutions. The page features a red banner at the top, a navigation bar with links to "Home", "Admin Login Here", "User Login Here", and "New User Signup Here", and a prominent banner showcasing an illustration of a person standing in front of a large "AI" chip surrounded by clouds and data cubes. Below the navigation bar is the new user signup form, which requires users to enter their username, password, contact number, email ID, and address. Once the required information is entered, users can click the "Submit" button to create a new account and gain access to the platform's error resolution services.

Fig 7: User Login

The user login screen of an AI-Driven Error Resolution Platform for Streamlined Solutions. The page features a red banner at the top, a navigation bar with links to "Home", "Admin Login Here", "User Login Here", and "New User Signup Here", and a prominent banner showcasing an illustration of a person standing in front of a large "AI" chip surrounded by clouds and data cubes. Below the navigation bar is the user login form, which requires users to enter their username and password. Once the correct credentials are entered, users can log in to the system to access the platform's AI-powered error resolution services and streamline their problem-solving processes.

Fig 8: User Home Screen

The user homepage of an AI-Driven Error Resolution Platform for Streamlined Solutions. The page features a red banner at the top, a navigation bar with links to "Search Error Solution" and "Logout", and a prominent banner showcasing an illustration of a person standing in front of a large "AI" chip surrounded by clouds and data cubes. Below the navigation bar is a welcome message that greets the user, "Welcome kumar". This homepage provides users with the ability to search for error solutions using the platform's AI-powered capabilities, along with the option to log out of their account.

Fig 9: Search Error Solution

The "Search Error Solution" screen of an AI-Driven Error Resolution Platform for Streamlined Solutions. The page features a red banner at the top, a navigation bar with links to "Search Error Solution" and "Logout", and a prominent banner showcasing an illustration of a person standing in front of a large "AI" chip surrounded by clouds and data cubes. Below the navigation bar is the "Error Solution Screen" section, which provides a text box labeled "Your Error". Users can enter their error description into this box and click the "Submit" button to initiate the AI-powered error resolution process. The platform will then analyze the error and provide a suitable solution, leveraging its advanced machine learning capabilities to deliver efficient and effective problem-solving assistance.

Fig 10: Result Search Error Solution

The "Search Error Solution" screen of an AI-Driven Error Resolution Platform for Streamlined Solutions. The page features a red banner at the top, a navigation bar with links to "Search Error Solution" and "Logout", and a prominent banner showcasing an illustration of a person standing in front of a large "AI" chip surrounded by clouds and data cubes. Below the navigation bar is the "Error Solution Screen" section, which displays the error message "Solution: Unable to predict solution. Please try other query". This indicates that the AI model encountered difficulties in providing a solution for the specific error entered by the user. The user is prompted to try entering a different query or rephrasing their error description to obtain more accurate results from the platform's AI-powered problem-solving capabilities.

5. CONCLUSION

The AI-Driven Error Resolution Platform represents a significant advancement in the field of server management, offering a streamlined and efficient solution to the challenges posed by traditional error resolution methods. By leveraging machine learning algorithms, this platform can predict accurate solutions to server errors based on historical data, significantly reducing the time and effort required by users to resolve issues. The platform not only enhances productivity but also improves system reliability by minimizing downtime and ensuring that errors are resolved quickly and accurately. In conclusion, the traditional approach to error resolution is no longer sufficient in today's complex server environments. The manual process is time-consuming, prone to errors, and lacks the efficiency needed to keep up with the growing volume of server errors. The AI-Driven Error Resolution Platform addresses these challenges by providing an intelligent, automated solution that streamlines the error resolution process. By aggregating server logs and training an AI model to predict solutions, the platform offers users a one-stop solution for resolving errors quickly and accurately. This represents a significant improvement over traditional methods, reducing the cognitive load on users and ensuring that errors are resolved in a timely manner.

REFERENCES

[1] Agt, H., Kutsche, RD.: Automated construction of a large semantic network of related terms for domain-specific modeling. In: International Conference on Advanced Information Systems Engineering, Springer, pp 610–625 (2013)

[2] Agt-Rickauer, H., Kutsche, RD., Sack, H.: Automated recommendation of related model elements for domain models. In: International Conference on Model-Driven Engineering and Software Development, Springer, pp 134–158 (2018a)

[3] Agt-Rickauer, H., Kutsche, RD., Sack, H.: DoMoRe? a recommender system for domain modeling. In: Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development, pp 71–82 (2018b)

[4] Alspaugh, S., Zokaei, N., Liu, A., Jin, C., Hearst, M.A.: Futzing and moseying: interviews with professional data analysts on exploration practices. IEEE Transact. Vis. Comput. Gr. **25**(1), 22–31 (2019). https://doi.org/10.1109/TVCG.2018.2865040

[5] Ángel, M.S., de Lara, J., Neubauer, P., Wimmer, M.: Automated modelling assistance by integrating heterogeneous information sources. Comput. Lang. Syst. Struct. **53**, 90–120 (2018)

[6] Annett, M.: (digitally) inking in the 21st century. IEEE Comput. Gr. Appl. **37**(1), 92–99 (2017). https://doi.org/10.1109/MCG.2017.1

[7] Arora, C., Sabetzadeh, M., Briand, L., Zimmer, F.: Extracting domain models from natural-language requirements: approach and industrial evaluation. In: Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, pp 250–260 (2016)

[8] Arora, C., Sabetzadeh, M., Nejati, S., Briand, L.: An active learning approach for improving the accuracy of automated domain model extraction. ACM Transact. Softw. Eng. Methodol. (TOSEM) **28**(1), 1–34 (2019)

[9] Aßmann, U., Zschaler, S., Wagner, G.: Ontologies, meta-models, and the model-driven paradigm. In: Ontologies for software engineering and software technology, Springer, pp 249–273 (2006)

[10] Barat, S., Khadilkar, H., Meisheri, H., Kulkarni, V., Baniwal, V., Kumar, P., Gajrani, M.: Actor based simulation for closed loop control of supply chain using reinforcement learning. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, pp 1802–1804 (2019a)

[11] Barat, S., Kulkarni, V., Clark, T., Barn, B.: An actor based simulation driven digital twin for analyzing complex business systems. In: 2019 Winter Simulation Conference (WSC), IEEE, pp 157–168 (2019b)

[12] Barat, S., Kumar, P., Gajrani, M., Khadilkar, H., Meisheri, H., Baniwal, V., Kulkarni, V.: Reinforcement learning of supply chain control policy using closed loop multi-agent simulation. In: Paolucci, M., Sichman, J.S., Verhagen, H. (eds.) Multi-Agent-Based Simulation XX, pp. 26–38. Springer International Publishing, Cham (2020)

[13] Barat, S., Parchure, R., Darak, S., Kulkarni, V., Paranjape, A., Gajrani, M., Yadav, A.: An agent-based digital twin for exploring localized non-pharmaceutical interventions to control

covid-19 pandemic. Transact. Indian Natl. Acad. Eng. (2021). https://doi.org/10.1007/s41403-020-00197-5

[14] Barriga, A., Rutle, A., Heldal, R.: Personalized and automatic model repairing using reinforcement learning. In: 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), IEEE, pp 175–181 (2019)

[15] Bikakis, A., Papatheodorou, C., Antoniou, G.: The DR-Prolog tool suite for defeasible reasoning and proof explanation in the semantic web. In: Darzentas J, Vouros GA, Vosinakis S, Arnellos A (eds) Artificial Intelligence: Theories, Models and Applications, 5th Hellenic Conference on AI, SETN 2008, Syros, Greece, October 2-4, 2008. Proceedings, Springer, Lecture Notes in Computer Science, vol 5138, p 345–351, https://doi.org/10.1007/978-3-540-87881-0_31, (2008)

[16] Bill, R., Fleck, M., Troya, J., Mayerhofer, T., Wimmer, M.: A local and global tour on momot. Softw. Syst. Model. **18**(2), 1017–1046 (2019). https://doi.org/10.1007/s10270-017-0644-3

[17] Black, D., Rapos, EJ., Stephan, M.: Voice-driven modeling: Software modeling using automated speech recognition. In: 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), IEEE, pp 252–258 (2019)

[18] Bordea, G., Faralli, S., Mougin, F., Buitelaar, P., Diallo, G.: Evaluation dataset and methodology for extracting application-specific taxonomies from the Wikipedia knowledge graph. In: Proceedings of the 12th Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, pp 2341–2347, https://www.aclweb.org/anthology/2020.lrec-1.285 (2020)

[19] Brambilla, M., Cabot, J., Cánovas Izquierdo, JL., Mauri, A.: Better call the crowd: using crowdsourcing to shape the notation of domain-specific languages. In: Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering, pp 129–138 (2017a)

[20] Brambilla, M., Cabot, J., Wimmer, M.: Model-Driven Software Engineering in Practice: Second Edition, 2nd edn. Morgan&; Claypool Publishers (2017b)