IncidentNavigator: Intelligent Support Incident Resolution using RAG and LLMs

I Pre-proposal context, positioning, and objective(s)

a Context

In the field of incident management, resolving recurring issues efficiently is a significant challenge. Operators often encounter problems that have already been addressed but struggle to locate the corresponding solutions within historical databases. This inefficiency leads to unnecessary duplicate tickets, increasing both resolution time and workload.

Our project, IncidentNavigator, aims to address this challenge by leveraging a Large Language Model (LLM) integrated with Retrieval-Augmented Generation (RAG) techniques. The objective is to develop a system that retrieves relevant solutions from an existing knowledge base and interacts dynamically with the user to refine queries. This ensures that the most pertinent results are presented, enhancing both accuracy and user experience.

a1 Positioning to State of the Art

Current solutions in incident management leverage Retrieval-Augmented Generation (RAG) to improve information retrieval and resolution accuracy. For instance, systems like WAISE[1] by XWiki use RAG to retrieve information from knowledge bases, aiming to streamline access to relevant solutions. However, these systems typically focus on static retrieval and lack dynamic interaction, which limits their effectiveness when dealing with complex or ambiguous queries.

Incident management systems exploit Retrieval-Augmented Generation (RAG) to improve information retrieval and resolution accuracy. Existing RAG solutions can be categorized into three main approaches:

1. Static Retrieval Systems: Traditional systems like ElasticSearch[2] focus on keyword matching and basic natural language processing. While efficient, they lack contextual understanding, especially for technical incidents involving varied terminologies. Modern approaches like Dense Passage Retrieval (DPR) [5] use deep learning embeddings to improve semantic search but still offer limited interactivity.

2. Advanced RAG Techniques: Recent research integrates LLMs with RAG for enhanced retrieval. For example, systems such as FiD (Fusion-in-Decoder) [2] use multiple retrieved documents to generate comprehensive answers. Techniques like those in the ColBERT model [6] allow efficient retrieval with contextual embeddings, improving query relevance. However, these methods often focus on static retrieval without interactive refinement.

3. Interactive Retrieval and Dynamic Systems: Dynamic retrieval systems represent a more interactive evolution. RAG turn [11] demonstrates how conversational AI can refine queries iteratively, improving accuracy for vague or incomplete inputs. Interactive LLMs can ask clarifying questions, which studies [4, 12] show significantly enhance performance in complex environments like incident management.

4. Transparency and Explainability: A crucial limitation of current LLM-based solutions is the lack of transparency. Systems like LIME [10] and SHAP [8] focus on explainability but are rarely integrated into RAG frameworks. Our approach will prioritize traceable outputs, referencing specific ticket IDs and metadata, reducing hallucinations [17].

[1] https://design.xwiki.org/xwiki/bin/view/Proposal/X-AI/WAISE/
[2] https://www.elastic.co/elasticsearch/

5. Multi-Modal and Cross-Domain Flexibility: Graph-based systems [18] handle structured data well but struggle with unstructured information such as logs or forums. Our system's ability to integrate diverse data sources, similar to approaches by Zhang et al. [19], ensures adaptability across different incident management environments.

In summary, while existing RAG systems show promise, our project will combine dynamic LLM interaction, enhanced transparency, and flexible data handling to offer a comprehensive solution tailored for incident resolution:

1. Dynamic Interaction: Unlike static search engines, our system uses LLMs to guide the user through a step-by-step refinement process. This interactive dialogue helps clarify ambiguities in user queries, leading to more accurate results [4, 12].
2. Enhanced Transparency: While existing systems often generate responses that lack source attribution, IncidentNavigator will prioritize displaying concise, context-aware solutions, directly citing ticket IDs, dates, and other metadata. This ensures users can trace back to verified sources, reducing the risk of "hallucinations" [3, 17].
3. Flexibility Beyond Structured Data: Our system will be able to handle a mix of structured (ticket systems) and unstructured (forums, logs) data, making it more adaptable to various incident management environments compared to graph-based approaches [18], which require complex preprocessing and maintenance. This allow us to keep our databases easily updated [3].

b Project's Objectives and Methodology

b1 Objective 1: Retrieve valid data for training

The data used to fine-tune the chosen LLM should not be specific to one particular domain. To get such data, we can proceed as follow:

- Scrap issue data from open-source IT project, questions / answer from IT forum (stackoverflow[3] for example).
- Rely on existing open database on industrial incident like ARIA's database[4] document detailed reports of industrial incidents across multiple sectors, including oil, nuclear, and chemical industries. These reports include incident descriptions, root cause analyses, and corrective actions, offering valuable context for training the model to handle complex, cross-domain incident data. Other resources, like the U.S. Chemical Safety Board (CSB)[5] repository, also provide comprehensive case studies on industrial safety incidents.

Data should be carefully curated to maintain quality and relevance. This includes filtering out noisy or redundant information, ensuring a balance across various domains, and anonymizing sensitive data to comply with privacy regulations. By combining diverse sources, the fine-tuned LLM will be better equipped to handle a wide range of incident types, providing accurate and context-aware solutions across different sectors.

b2 Objective 2: Develop a reliable and transparent RAG system

The primary goal is to create a RAG-based system that retrieves verified solutions from historical tickets. The LLM will generate responses grounded in fact-based data, including ticket identifiers, tags words, and solution dates. Transparency is key: each response will include source references, fostering trust and facilitating quick decision-making.

To perform semantic similarity searches, we will utilize advanced embedding models. While BERT is commonly used for general natural language tasks [3, 17], it may face limitations with technical tickets containing code or mixed data types. Therefore, we plan to explore more versatile, pre-trained models such as:

[3] https://stackoverflow.com/
[4] https://www.aria.developpement-durable.gouv.fr/?lang=en
[5] https://www.csb.gov/investigations/

- RoBERTa [7]: A robustly optimized BERT variant known for better performance on diverse NLP tasks, including technical text.
- E5 [16]: A recent model optimized for text embedding, known to handle mixed-domain inputs effectively, making it suitable for tickets that combine natural language and code snippets.
- LLaMA [14]: A general-purpose LLM with strong performance across various tasks. Fine-tuning this model for incident management will ensure it remains effective for both standard descriptions and occasional code fragments.

The LLM used afterward to reply and refine the user's request will be a pre-trained LLM (such as LLaMa models) that will be fine-tuned for these specifics tasks. The choice for the precise model will be based on existing benchmark, that allow us to know what model is currently the most solid base for our application. The different benchmark used to do so would be BEIR [13] and SQuAD [9] for retrieval performance, and then MMLU [1] and GLUE [15] for general language understanding.

Finally, we have to conduct some experiment to validate our results. Experimental validation will involve:

- Test Scenarios: Evaluate the system with a dataset of real-world incidents, testing accuracy in retrieving relevant solutions. Compare performance against baseline methods.
- User Testing: Conduct usability studies with incident managers to assess system effectiveness and user satisfaction.
- Metrics: Measure precision, recall, and F1-score for retrieved solutions. Evaluate LLM responses using human-in-the-loop verification.

b3 Objective 3: Implement an interactive user interface for guided search refinement

We aim to design an intuitive, dual-panel interface featuring:

- **Left Panel:** A list of retrieved solutions, each with a brief summary and source link.
- **Right Panel:** An interactive chat interface where the LLM guides users by asking clarifying questions (e.g., system version, error codes).

This dynamic interaction ensures users are directed to the most relevant solutions, particularly for complex or vaguely described issues. Such proactive refinement processes have shown to significantly improve retrieval precision in customer support applications [3, 17]. You can see an example in figure 1, where the refinement process is shown with blue arrows when the LLM refines solutions based on the new informations the user gave, and in red the next question / reply the LLM deduce from this new state.

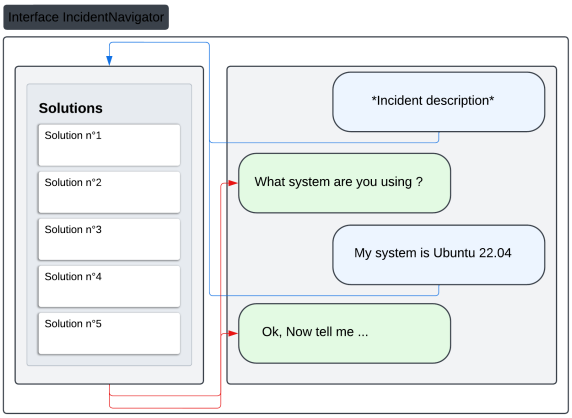


Figure 1. A basic view of the interface of the IncidentNavigator

b4 Final Objective: First Prototype

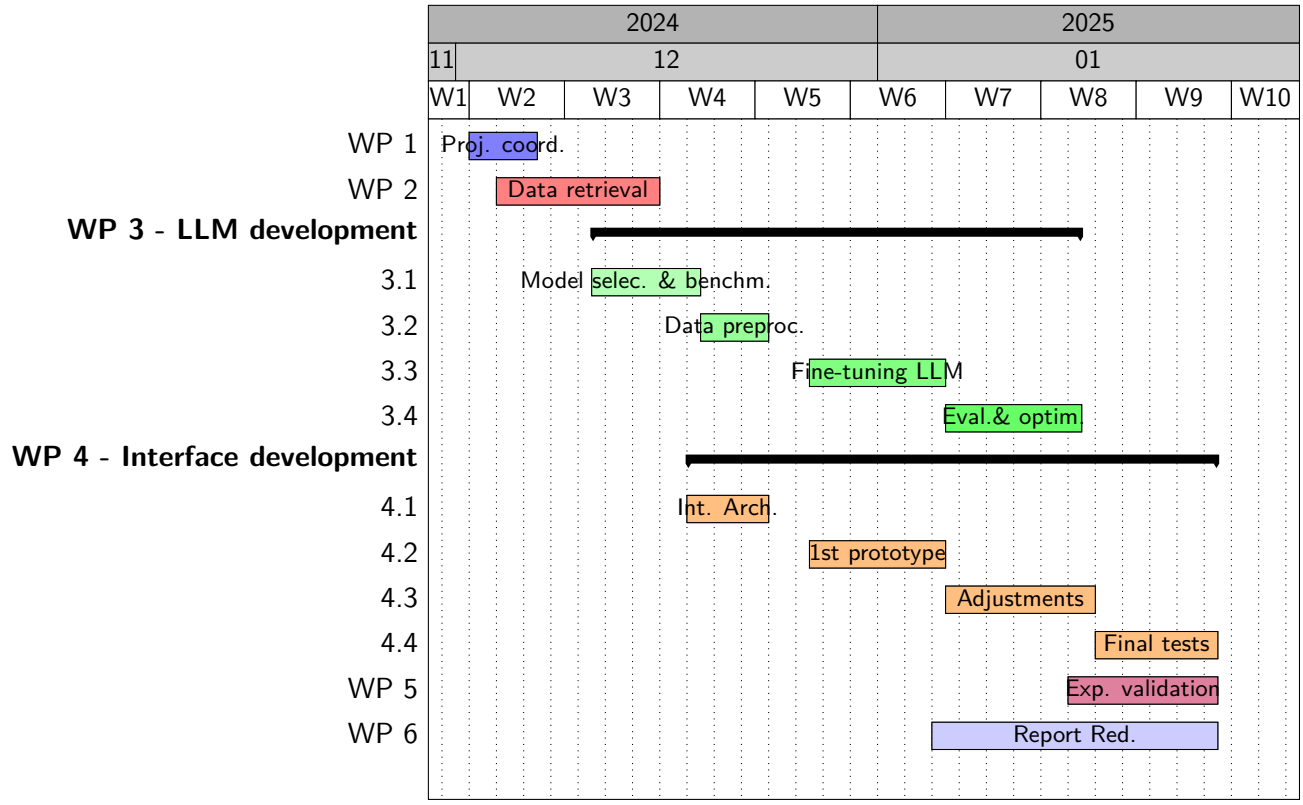
Because this project should be done within 2 months, the ultimate goal is to develop a functional prototype of the IncidentNavigator. This prototype will serve as a proof of concept, demonstrating the system's capability to retrieve relevant solutions from historical tickets and generate fact-based, transparent responses. While the prototype may not include all the features and optimizations outlined in the detailed objectives above, it will focus on the core functionalities needed to validate the system's feasibility and effectiveness.

The key characteristics of the prototype include:

- **Core functionality:** The prototype must effectively perform semantic similarity searches and provide accurate solutions based on real-world incidents.
- **Transparency and reliability:** Each generated response should include clear source references, fostering trust and facilitating quick decision-making.
- **User experience:** The system should feature an intuitive interface that supports guided search refinement, enhancing usability for incident managers.

The prototype will undergo validation to ensure it meets these general objectives through a series of technical tests and user evaluations. This phased approach allows for incremental development and testing, laying the foundation for a more comprehensive system in future iterations.

c Project's schedule



References

[1] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. *Measuring Massive Multitask Language Understanding*. 2021. arXiv: 2009.03300 [cs.CY]. URL: https://arxiv.org/abs/2009.03300.

[2] Gautier Izacard and Edouard Grave. *Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering*. 2021. arXiv: 2007.01282 [cs.CL]. URL: https://arxiv.org/abs/2007.01282.

[3] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. *Atlas: Few-shot Learning with Retrieval Augmented Language Models*. 2022. arXiv: 2208.03299 [cs.CL]. URL: https://arxiv.org/abs/2208.03299.

[4] Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. *Active Retrieval Augmented Generation*. 2023. arXiv: 2305.06983 [cs.CL]. URL: https://arxiv.org/abs/2305.06983.

[5] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. *Dense Passage Retrieval for Open-Domain Question Answering*. 2020. arXiv: 2004.04906 [cs.CL]. URL: https://arxiv.org/abs/2004.04906.

[6] Omar Khattab and Matei Zaharia. *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*. 2020. arXiv: 2004.12832 [cs.IR]. URL: https://arxiv.org/abs/2004.12832.

[7] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL]. URL: https://arxiv.org/abs/1907.11692.

[8] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. arXiv: 1705.07874 [cs.AI]. URL: https://arxiv.org/abs/1705.07874.

[9] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. *SQuAD: 100,000+ Questions for Machine Comprehension of Text*. 2016. arXiv: 1606.05250 [cs.CL]. URL: https://arxiv.org/abs/1606.05250.

[10] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. 2016. arXiv: 1602.04938 [cs.LG]. URL: https://arxiv.org/abs/1602.04938.

[11] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. *Retrieval Augmentation Reduces Hallucination in Conversation*. 2021. arXiv: 2104.07567 [cs.CL]. URL: https://arxiv.org/abs/2104.07567.

[12] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. *Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph*. 2024. arXiv: 2307.07697 [cs.CL]. URL: https://arxiv.org/abs/2307.07697.

[13] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. *BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models*. 2021. arXiv: 2104.08663 [cs.IR]. URL: https://arxiv.org/abs/2104.08663.

[14] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[15] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*. 2019. arXiv: 1804.07461 [cs.CL]. URL: https://arxiv.org/abs/1804.07461.

[16] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. *Improving Text Embeddings with Large Language Models*. 2024. arXiv: 2401.00368 [cs.CL]. URL: https://arxiv.org/abs/2401.00368.

[17] Zhentao Xu, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, and Zheng Li. "Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering". In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR 2024. ACM, July 2024, pp. 2905–2909. DOI: 10.1145/3626772.3661370. URL: http://dx.doi.org/10.1145/3626772.3661370.

[18] Zi Ye, Yogan Jaya Kumar, Goh Ong Sing, Fengyan Song, and Junsong Wang. "A Comprehensive Survey of Graph Neural Networks for Knowledge Graphs". In: *IEEE Access* 10 (2022), pp. 75729–75741. DOI: 10.1109/ACCESS.2022.3191784.

[19] Zihao Zhao, Zhihong Shen, Mingjie Tang, Chuan Hu, Huajin Wang, and Yuanchun Zhou. *PandaDB: Understanding Unstructured Data in Graph Database*. 2022. arXiv: 2107.01963 [cs.DB]. URL: https://arxiv.org/abs/2107.01963.