

Generative AI – Improving Agile Software Projects Using Contextual Retrieval and LLMs



TECHNISCHE
UNIVERSITÄT
DARMSTADT

November 2023

1 Motivation

With the rise of large-scale agile software projects, the management of user stories has become an intricate task. User stories, serving as the primary unit for conveying requirements in such projects, often proliferate to the point where redundancy and overlap become serious concerns. Recognizing similar user stories can aid in reducing redundancy, optimizing resource allocation, and ensuring cohesive product development. Recent advancements in Language Model Learning (LLM) and the capabilities of vector databases offer a promising avenue for tackling this challenge. Through efficient similarity analysis, we can not only improve the operational efficiency of agile projects but also derive insights about recurrent user needs and preferences, leading to better software design and user satisfaction. This thesis aims to bridge the gap between cutting-edge linguistic model technology and practical challenges in managing agile software projects, leading to actionable insights and more efficient project management.

2 Task Description

The core task of this thesis will revolve around the utilization of LLM methodologies combined with vector databases to find similarities in user story descriptions within large-scale agile software projects.

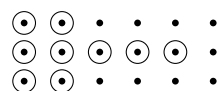
- Dataset Collection & Preprocessing to ensure uniformity, remove noise, and make it suitable for LLM.
- LLM Implementation, focusing on generating embeddings of the user stories which capture the semantic essence of each story.
- Vector Database Integration: Store these embeddings in a vector database, ensuring efficient querying capabilities. This setup will enable the fast retrieval of similar user stories based on their vector representations.
- Similarity Analysis: Design and implement a robust mechanism to query the vector database to identify similar user stories. This step will involve determining a suitable similarity threshold and optimizing for both accuracy and computational efficiency.
- Evaluation: Assess the accuracy, efficiency, and scalability of the implemented system. This will involve creating test sets, defining metrics for evaluation, and comparing results against other standard methodologies if available.
- Insights & Recommendations: Beyond mere similarity detection, the thesis should also offer insights into patterns of redundancy in user stories and make recommendations for optimizing user story creation and management in agile projects.

3 References

- Li Jiwei et al., (2016). Deep reinforcement learning for dialogue generation. In Proc. of EMNLP 2016. pp. 1192-1202.
- Bansal, Trapit, et al. "Learning to Few-Shot Learn Across Diverse Natural Language Classification Tasks." ACL. 2020.

4 Profile of the Thesis

Literature
Programming
Analysis



5 Contact

If you apply for this thesis, we ask you to provide us with a short motivation letter, your CV and your current reference.

Dr. Timo Koppe
timo.koppe@tu-darmstadt.de