Proposition de sujet de stage recherche, niveau M2
Année universitaire 2024-2025

Établissement : UCA, Clermont-ferrand
Laboratoire: LIMOS

Encadrement :
Sébastien Salva, sébastien.salva@uca.fr
Redha Taguelmimt

Profil recherché :
Master 2 Informatique ou équivalent (école d'ingénieur)

Financement: Chaire industrielle

Période de stage : Mars à Juillet 2025

Compétences recherchées :
Programmation, python pour IA, Tests, anglais

Mots-clés : LLM, RAG, Tests, Sécurité

Titre du stage: GUI browsing and testing with LLM with RAG

Many applications, particularly web applications, are composed of graphical user interfaces (GUIs) that allow users to navigate through various functionalities. The content of these GUIs can evolve over time or change dynamically during execution. This dynamic behavior makes automatic browsing and functional testing of these applications challenging. Test scripts must be continuously modified to adapt to the evolving structure of the GUIs, both in terms of content and hierarchical organization.

The objective of this internship is to propose a method and tool **to automate browser-based workflows and testing** (conformance from user stories and security) **by leveraging large language models (LLMs).** The LLM will be employed to interpret user requirements and **dynamically generate scripts to navigate through GUIs** and perform verifications. A secondary objective is to use small language models (SLMs), complemented with retrieval-augmented generation (RAG) techniques [2], deployed locally. These SLMs should help reduce power consumption while preserving the confidentiality of the data used by the RAG techniques. But RAG techniques require Data. Hence, to reach these objectives, the following points will be considered :

Data generation: This involves generating information from the current project as well as other projects, including GUI details such as content, names, links, and structures, using a tool like FireCrawl. Additionally, a knowledge graph representing the application structure is generated, e.g., through the use of a web crawler;

Expert Knowledge Data Extraction: A knowledge graph, expressing user knowledge about browsing and testing, is designed to represent the reasoning aspect of the method. The graph will express steps to follow if the GUI exploration is blocked or actions to perform for enhancing GUI coverage, and for improving testing.

Use of RAG techniques to enhance LLM (or SLM) responses: This section focuses on building vector storage and implementing two RAG techniques: one using Word2Vec and the other using Graph2Vec. Additionally, two key RAG concepts will be explored: the Retriever, which rephrases user instructions to improve browsing and testing, and Memory, which enhances prompts by recalling previous actions. These techniques aim to make the LLM (or SLM) more adaptive and context-aware, allowing it to generate more precise and relevant outputs. The incorporation of vector storage further enhances the system's ability to understand and link patterns, improving both accuracy and response speed over time;

Evaluation: This section will use various metrics to assess the effectiveness of the LLMs or SLMs. Metrics considered will include Entropy, self-reflective measures, and test passing correctness. Additionally, tools based on user requests and responses (e.g., RAGAS [1]) could be employed to further evaluate the LLMs with RAG. This approach will not only help in determining the effectiveness of every LLM or SLM with RAG combinations but also provide a basis for comparing two LLMs with RAG having different parameters. By incorporating these diverse metrics, we aim to offer a comprehensive evaluation of the models' ability to handle dynamic workflows and user interactions.

Should time allow, the candidate for this internship may also explore the use of networks of small language models (SLMs) dedicated to specific tasks, which can interact with each other to fulfill a user request. The Langroid framework [4] can be used to facilitate interaction and coordination between these different models, providing a modular and flexible approach for handling complex requests.

Publication associées au sujet :
[1] RAGAS: Automated Evaluation of Retrieval Augmented Generation, https://arxiv.org/abs/2309.15217
[2] Retrieval-Augmented Generation for Large Language Models: A Survey, https://export.arxiv.org/abs/2312.10997v2
[3] A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions." (Huang, Lei, et al., 2023) https://arxiv.org/pdf/2311.05232
[4] Langroid Multi Agent Framework of LLMs applications https://github.com/langroid/langroid/tree/main

Some Libs and documentation:
https://www.langchain.com/ LLM tool chain, etc.
https://www.llamaindex.ai/ LLM lib. Specialised for RAG
https://www.trychroma.com/ chromadb, embeddings storage
https://github.com/mendableai/firecrawl, API service that takes a URL, crawls it, and converts it into clean markdown or structured data.

Durée et début : 5 à 6 mois, début mars/avril suivant possibilités. Proposition de thèse par la suite.

Encadrant : Sébastien Salva
Courriel / Téléphone :sebastien.salva@uca.fr, https://perso.limos.fr/~sesalva/
Unité de rattachement : LIMOS
Equipe : LIMOS / DSI
Etablissement de rattachement : UCA

Co-encadrant : Redha Taguelmimt
Unité de rattachement : LIMOS

Etablissement de rattachement : UCA

Si vous souhaitez candidater, merci de déposer votre candidature avant fin décembre par mail à sebastien.salva@uca.fr en joignant un CV complet, relevés de notes et lettres de soutien éventuelles.