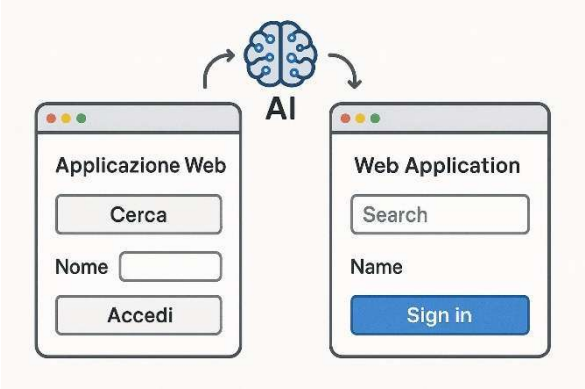
Project Q2

Generative AI & Automatic GUI Translation

Project's summary

The primary goal of this project is to develop a module that integrates with web-based applications (e.g., built with React/Angular) to provide real-time, automatic translation of the Graphical User Interface (GUI). Utilizing a locally installed generative AI model (such as LLAMA-2 or LLAMA-3), the module will take a target language selection from the user and dynamically translate all visible text elements within the application's interface into that target language, irrespective of the GUI's original source language. This aims to enhance accessibility and usability of web applications for a global audience without requiring developers to manually create translations for every supported language.



Problem Definition

Web applications often need to support multiple languages to reach a wider user base. Traditional localization methods involve manually translating UI strings and managing language files, which can be time-consuming, costly, and difficult to maintain, especially for applications with frequently changing content.
This project explores using large language models (LLMs) for on-the-fly GUI translation. Key challenges include:

- **Integration:** Seamlessly linking the translation module with existing web frameworks (React/Angular) to intercept and translate text elements without disrupting application functionality.
- **Context Preservation:** Ensuring translations maintain the correct meaning within the context of the UI element (e.g., button labels vs. descriptive text).
- **Performance:** Achieving real-time translation speed using a locally hosted LLM without significantly impacting the user experience.

- **Dynamic Content:** Handling text that is dynamically generated or updated within the application.
- **Accuracy:** Evaluating the quality and accuracy of translations generated by the selected LLM for various languages and UI contexts.
- **Local Deployment:** Managing the setup and resource requirements of running a generative AI model locally.

Required Background

- Strong programming skills (language depends on implementation, likely JavaScript/TypeScript for web integration, Python for AI interaction).
- Experience with web development frameworks.
- Understanding fundamental AI/ML concepts and experience using LLM APIs or libraries.
- Familiarity with basic software engineering principles (modularity, testing).
- Courses covering web development, AI/Machine Learning, and potentially system programming would be beneficial.

Working Environment

- Students can work on laptops or desktops (Unix-like or Windows).
- **Core Implementation:** Web framework (React/Angular), integration module (e.g., JavaScript/TypeScript).
- **AI Backend:** Local installation of a generative AI package (e.g., LLAMA-2, LLAMA-3), potentially accessed via Python or dedicated APIs.
- Access to reasonably powerful hardware might be needed to host the local LLM effectively.

Deliverables

All projects must be delivered, including the following material:

- **Source Code:** Well-commented code for the translation module, any backend interaction scripts, and potentially a sample web application for demonstration.
- **README File:** Instructions on how to set up the environment (including the local LLM), install dependencies, configure the module, and run the demonstration application.
- **Documentation File:** A report (Word, LaTeX, Markdown, etc.) detailing:
  - System Architecture: How the module integrates with the web app and interacts with the AI model.
  - Design Choices: Rationale for framework selection, AI model choice, translation strategy (e.g., how text is identified and replaced), dynamic content handling.

    - Experimental Evaluation: Assessment of translation quality (e.g., using sample UIs and target languages, potentially BLEU scores if reference translations are available), performance benchmarks (translation latency), discussion of limitations.
- **Presentation:** Slides (PowerPoint or similar) for a final project presentation (approx. 15 minutes) covering the project goals, design, results, and conclusions.

Contact

Prof. Stefano Quer (stefano.quer@polito.it).