Project Title

HAMK Thesis Assistant (Thesizer)

Group Members

Group Member 1 Name: Aleksi Hellgren

Group Member 2 Name: Jaakko Saarikko

Group Member 3 Name: Saku Karttunen

Group Member 4 Name: Samppa Sirnö

Group Member 5 Name: Wais Atifi

Project Description

Provide a concise description of your project objectives, the problem you're addressing, and how AI will be used to solve it.

HAMK has a lot of documentation and guidelines for authoring the theses, but it can be overwhelming and difficult to navigate, since the information is all over the place. For example in different documents and websites, like moodle and hamk.fi. Our idea for the project is to expand a pretrained LLM-model to include information about HAMK's thesis documentation.

The end goal is to create an application that its' users can utilize for easy search of detailed information and the technical details on authoring their thesis. This way of searching for help will (hopefully) be easier and faster so that the user can focus on what is the most important.

The contents of the thesis.

Model Selection

Specify the AI model(s) you will use and justify why you chose this model for your project.

Chosen Model:

Google Gemma 2 Instructions

Justification:

Explain why this model is suitable for your project.

Gemma 2 has felt very high quality for the size of the model that it is, so thanks to that it provides better general knowledge and feels more natural and consistent to talk to. We used the 2B parameter version of the model for the development process and it was already very good quality.

The instructions version was chosen, because it followed the prompts and the RAG context more closely than the non-instruction counterpart.

Fine-Tuning and Adaptation

Fine-Tuning:

Describe your approach to fine-tuning the chosen model. Indicate if you're using an existing dataset or creating your own.

We used the LangChain-framework and the RAG processing technique to feed the model more information dynamically and very easily. The LangChain-framework provided a lot of help especially with the vector store and fetching of the data for the model. It also had good tools for parsing data from different types of files.

Alongside this, we are creating a chat based prompt template, so that the model is able to understand the history of the conversation as well as the roles.

Dataset Creation:

If creating your own dataset, describe its source, format, and preprocessing steps.

We have a dataset that is composed of HAMK thesis guidelines as well as some hamk websites. These documents are written in english and in finnish. So far we have more finnish data than english ones, but we are trying to compile both to the best of our ability and time.

The documents are in html, pdf plaintext and markdown forms. These documents are parsed based on the file extension in the vector_store.py file when creating the vector store database.

Additional Functionality:

Outline any extra functionalities you plan to add (e.g., interactive components, system integration).

The model will be uploaded to Hugging Face -spaces where it can be interacted with using a Gradle based web-UI. This gradle UI is used to chat with the model.

We chose Hugging Face, because it provides easy hosting for an AI project. It is also affordable and provide great availability. This makes it possible to use it for ourselves in the future easily from just a website.

Contributions

Below is a list of possible contributions your project may include. You are encouraged to address multiple points where applicable:

- **New Application Idea:** Innovate by using an existing model in a new context, such as applying it to a domain or language not previously explored.
- **Fine-Tuning with Existing Dataset:** Customize a pre-trained model using an existing dataset to enhance its performance or adapt it to a specific problem.
- **Fine-Tuning with New Dataset:** Create and use a new dataset tailored to your project, and fine-tune the model using this dataset to address unique challenges.
- **Multilingual Support:** Enable support for languages other than English by fine-tuning or extending a model designed for multiple languages.
- **Cross-Modal Functionality:** Extend models to support multiple data types (e.g., text-to-speech, speech-to-text, image generation, or video analysis).
- **Improved Model Efficiency:** Optimize the model's efficiency in terms of speed, memory usage, or inference time without sacrificing performance.
- **Interactive AI Systems:** Integrate your model into an interactive system that can provide real-time feedback or responses, such as a chatbot or interactive assistant.
- **Novel Dataset Creation:** Create a novel dataset from scratch or gather data from an unstructured source (e.g., scraping, collecting, and labeling real-world data).
- **Multi-tasking Abilities:** Adapt a model to handle multiple tasks, such as summarization, translation, and classification, within a single architecture.