SRT File Translation Using Large Language Models

A Project in COS243 Prompt Engineering and Applications of Generative AI

Dr. Zheng Qu

Table of contents

1 Generative AI Project: SRT File Translation

1.1 Overview

In this project, you will develop a Python application that translates SRT (SubRip Subtitle) files from English to a target language of your choice using Large Language Models (LLMs). This project will help you gain hands-on experience with prompt engineering, API integration, and practical applications of generative AI.

1.2 Objectives

1. Understand and implement prompt engineering techniques
2. Integrate and interact with LLM APIs
3. Process and manipulate structured text data (SRT files)
4. Optimize for efficiency and cost-effectiveness in AI applications
5. Document and present your work using Quarto

1.3 Requirements

1.3.1 1. SRT Processing

- Develop Python code to read and parse SRT files
- Implement a chunking mechanism to handle long SRT files efficiently
- Extract relevant information (timestamp, text) from SRT entries

1.3.2 2. Prompt Engineering

- Design an effective prompt for the LLM to translate subtitle content
- Include instructions for:
  - Refining the English text (grammar, coherence)
  - Translating to the target language (Vietnamese or Chinese)
  - Maintaining the structure and context of chunked subtitles
- Implement a system to handle context between chunks (e.g., using markers like `<chunk-start>` and `<chunk-end>`)

1.3.3 3. LLM Integration

- Choose an LLM provider (e.g., OpenAI, Anthropic, deepinfra, or local models like Ollama)
- Implement API calls to the chosen LLM
- Handle API responses, including error handling and rate limiting

1.3.4 4. Output Processing

- Parse the LLM's response and reconstruct the translated SRT file
- Ensure proper formatting and timing of the translated subtitles

1.3.5 5. Optimization

- Implement strategies to reduce token usage and API calls
- Balance translation quality with cost-effectiveness

1.3.6 6. Documentation and Presentation

- Use Quarto to create a report documenting your project
- Include sections on:
  - Project overview and objectives
  - Methodology (prompt design, chunking strategy, LLM choice)
  - Code explanation (key functions and their purposes)
  - Sample inputs and outputs
  - Challenges faced and solutions implemented
  - Performance analysis (speed, accuracy, cost)
  - Potential improvements and future work

1.4 Deliverables

1. Notebook for the SRT translation application
2. Output translated SRT file for "COS501-01-OOP1-2024-08-27-14-15-28v2.srt" in project folder
3. PDF document with project report

1.5 Evaluation Criteria

- Functionality (30%): Does the application work as intended?
- Prompt Engineering (25%): Effectiveness of the designed prompt
- Code Quality (20%): Readability, organization, and documentation
- Optimization (15%): Efficiency in token usage and API calls
- Documentation (10%): Clarity and comprehensiveness of the report

1.6 Resources

- SRT file handling: `pysrt` library
- LLM providers: OpenAI, Anthropic, Cohere, Ollama (local models)
- Documentation: Quarto (https://quarto.org/)

1.7 Submission Guidelines

- Submit link to your project folder "srt-translate" on google drive inside your individual folder
- Ensure your code is well-commented and follows PEP 8 style guidelines