Multilingual Translation Using LLMs

Introduction/Motivation:

The goal of this project was to create multilingual versions of a company webpage by translating the original English content into several target languages, including the low-resource language Hungarian. A key requirement was to use open-source solutions, ruling out commercial translation services like DeepL and ChatGPT (at the time of the project).

Methodology and Key Challenges:

- **Model Selection:** Choosing an appropriate LLM was challenging due to the inclusion of low-resource languages like Hungarian. The selection process prioritized freely available models capable of handling a diverse set of languages. We evaluated Google T5, Meta's M2M-100, and MarianMT. Using a gold-standard Hungarian translation dataset, we compared their performance using the BLEU score, a common metric for evaluating machine translation quality based on n-gram overlap. MarianMT demonstrated the best performance among the open-source options and was chosen as the baseline LLM.
- **Baseline Comparison:** To assess the effectiveness of the LLM-based approach, we compared MarianMT's performance against Google Translate. MarianMT significantly outperformed Google Translate on the Hungarian dataset, achieving a BLEU score of 18.74 compared to Google Translate's 9.88.
- **Handling Placeholders:** The source text contained special placeholders (e.g., `[fistBrokenName]`) that should *not* be translated. We explored several strategies to address this:
    - **Special Character Manipulation:** Encapsulating placeholders with special characters to prevent the LLM from recognizing them as translatable text.
    - **Prompt Engineering:** Providing explicit instructions to the LLM not to translate these placeholders.
- **Prompt Engineering for Quality Improvement:** To further enhance translation quality, we employed prompt engineering techniques, including:
    - **Audience Definition:** Specifying the target audience for the translations.
    - **Domain Specification:** Defining the domain as "stock exchange" to provide context.
    - **Task Definition:** Clearly stating the task as "translating a company webpage."

Results:

The following BLEU scores were achieved on the Hungarian reference text:

- **LLM with Instructions (Prompt Engineering):** 18.75
- **LLM Baseline (MarianMT without detailed prompts):** 15.02
- **Google Translate Baseline:** 9.88

These results demonstrate that the LLM-based approach using MarianMT significantly outperforms Google Translate. Furthermore, prompt engineering yielded a noticeable improvement in translation quality. However, the best achieved BLEU score is still below 20, indicating room for further improvement.

Discussion:

The results confirm the effectiveness of using LLMs, particularly MarianMT, for multilingual translation, even for low-resource languages. The positive impact of prompt engineering highlights the importance of providing context to the LLM. The relatively low BLEU scores suggest that while the translations are likely understandable, they may lack fluency or naturalness.

Future Work and Potential Improvements:

- **Exploring Larger Models/Commercial Solutions:** Evaluating the performance of larger, more powerful models like those offered by DeepL or current state-of-the-art LLMs (if available under suitable licensing) could further improve translation quality. This was previously not an option due to project constraints.
- **Advanced Prompt Engineering:** Experimenting with more sophisticated prompt engineering techniques, such as few-shot learning (providing examples of correct translations), or more detailed contextual information.
- **Fine-tuning:** Fine-tuning MarianMT on a domain-specific dataset (e.g., financial texts in Hungarian) could significantly improve performance.
- **Evaluation Metrics:** Consider using additional evaluation metrics beyond BLEU, such as METEOR, TER, or human evaluation, to get a more comprehensive assessment of translation quality.
- **Error Analysis:** Conduct a detailed error analysis to understand the types of errors the model is making and tailor improvement strategies accordingly.