## **Idea #1**

## **Project Proposal: Extending ROMANSETU to Diverse Languages**

**1. Introduction**

Large Language Models (LLMs) have revolutionized Natural Language Processing (NLP) tasks, demonstrating remarkable proficiency in various English applications. However, extending their capabilities to non-English languages, particularly those with non-Latin scripts, remains a challenge. This project investigates the applicability and effectiveness of "ROMANSETU," a novel approach utilizing romanization for bridging the language gap and enhancing LLM performance in diverse languages.

**2. Research Objectives**

The primary objectives of this project are:

* **Evaluate the effectiveness of ROMANSETU:** To assess the performance improvement in LLM tasks achieved by romanized text compared to the native script across diverse languages.
* **Analyze influencing factors:** To investigate the impact of script complexity and the presence of established romanization systems on ROMANSETU's efficacy.
* **Explore language-specific adjustments:** To identify potential modifications to the ROMANSETU approach necessary for optimal performance in specific languages.
* **Explore Many-to-Many NMT: Zero-shot Transfer:** To identify the performance of LLM translation between two or more non-Latin languages

**3. Methodology**

**3.1 Language Selection:**

We will evaluate the effectiveness of the ROMANSETU paper by selecting Hindi, and performing the tasks proposed in the paper.

Additionally, we will select from a set of four languages representing diverse script types and levels of established romanization systems:

* **Arabic:** Complex script with a well-defined romanization system (ISO 2332)
* **Cyrillic:** Relatively complex script with various romanization schemes (e.g., GOST, Library of Congress)
* **Japanese:** Logographic script with existing romanization systems like Hepburn and Kunrei-shiki
* **Swahili:** Latin script language with limited formal romanization standards

**3.2 Dataset Preparation:**

For each selected language, we will:

* Gather a balanced dataset of text and labels for the chosen LLM task (e.g., machine translation, sentiment analysis).
* Prepare two versions of the dataset: one in the native script and another in its corresponding romanized form using established standards or a consistent transliteration scheme.

**3.3 LLM Training and Evaluation:**

* We will train an LLM model on each language dataset (native and romanized) using a pre-trained LLM like LLaMA2 7B, BART or T5.
* The trained models will be evaluated on the held-out test set for the chosen LLM task based on established metrics (e.g., BLEU score for translation, F1 score for sentiment analysis).
* Performance comparisons will be made between the models trained on native and romanized data for each language.

**3.4 Data Analysis and Adjustment Exploration:**

* We will statistically analyze the performance differences between models to assess the effectiveness of romanization across languages.
* Script complexity and the presence of established romanization systems will be analyzed as potential influencing factors on the observed results.
* We will explore potential language-specific adjustments to the ROMANSETU approach, such as:
  + Customized romanization schemes for languages with diverse script types
  + Addressing homophones and ambiguities arising from romanization
  + Investigating the use of language identification techniques to selectively apply romanization.

**4. Expected Outcomes**

This project is expected to contribute to the following:

* **Enhanced understanding:** Improved knowledge of how romanization impacts LLM performance across diverse languages.
* **Identification of influencing factors:** Insights into the impact of script complexity and existing romanization systems on the effectiveness of the approach.
* **Language-specific adaptations:** Recommendations for potential modifications to ROMANSETU for optimal performance in different language contexts.

**5. Tentative Timeline**

1. Language Selection (March 1 - March 5)
   1. Selection of another non-Latin language to run experiments on.
   2. Selection and familiarization with the new language’s corresponding romanization tool.
2. Replication of ROMANSETU Results for Hindi (March 6 - March 15)
3. Dataset Preparation for the new language (March 18 - March 22)
4. LLM Training and evaluation for the new language (March 25 - April 2)
5. Analysis and preparation for presentation (April 3 - April 7)
6. Report Preparation (April 8 - April 20)

**5. Conclusion**

By extending ROMANSETU to diverse languages and analyzing its efficacy, this project aims to contribute to bridging the language gap in LLM applications. The findings could pave the way for more inclusive and effective LLMs that empower communication and information access across different languages and cultures.

## **Idea #2**

## **Project Proposal: Comparative Analysis of ROMANSETU with Alternative Methods for Adapting LLMs to Non-English Languages**

**1. Introduction**

Large Language Models (LLMs) have demonstrated remarkable capabilities in various English language processing tasks. However, adapting them to non-English languages, particularly those with complex scripts or limited resources, remains a challenge. This project aims to conduct a comparative analysis of "ROMANSETU," a novel method utilizing romanization for bridging the language gap, with existing techniques for adapting LLMs to non-English languages.

**2. Research Objectives**

The primary objectives of this project are:

* **Compare the performance of ROMANSETU:** To assess the performance of ROMANSETU against existing techniques (transfer learning and code-switching) for LLM adaptation on different tasks and languages.
* **Analyze strengths and weaknesses:** To identify the specific strengths and weaknesses of each approach under various conditions, including resource availability and language complexity.
* **Identify optimal scenarios for ROMANSETU:** To determine the scenarios where ROMANSETU offers the most effective and beneficial solution for LLM adaptation.

**3. Methodology**

**3.1 Approach Selection:**

We will select three adaptation approaches for comparison:

* **ROMANSETU:** Training LLMs on romanized text of the target language.
* **Transfer Learning:** Fine-tuning pre-trained LLMs on a target language dataset.
* **Code-switching:** Training LLMs to handle code-mixed text containing both English and the target language.

**3.2 Experiment Design:**

* **Languages:** We will select two languages with varying script complexity and resource availability:
  + **High Complexity, Limited Resources:** Arabic (complex script, limited in-domain data)
  + **Moderate Complexity, Moderate Resources:** Swahili (Latin script, growing amount of data)
* **Tasks:** We will choose two LLM tasks for evaluation:
  + Machine Translation (English to target language and vice versa)
  + Sentiment Analysis (identifying sentiment in target language text)
* **Datasets:** Balanced datasets for each language and task will be prepared in native script, romanized form, and code-mixed form (English and target language).
* **Models:** A pre-trained LLM (e.g., BART or T5) will be used as the base model for all approaches.

**3.3 Training and Evaluation:**

* Each adaptation approach (ROMANSETU, Transfer Learning, Code-switching) will be implemented on the chosen languages, tasks, and datasets.
* Trained models will be evaluated on held-out test sets using relevant metrics (e.g., BLEU score for translation, F1 score for sentiment analysis).
* Statistical analysis will be conducted to compare the performance of each approach across tasks, languages, and conditions.

**3.4 Strengths and weaknesses analysis:**

* Based on the performance and evaluation results, the strengths and weaknesses of each approach will be analyzed under different conditions:
  + **Resource availability:** How does the amount of available training data impact each approach's performance?
  + **Language complexity:** How does the complexity of the script and language structure affect each approach's effectiveness?
  + **Task type:** Are there specific tasks where one approach consistently outperforms others?
* Additionally, practical considerations like ease of implementation and computational cost will be factored in.

**3.5 Identifying optimal scenarios for ROMANSETU:**

Based on the comprehensive analysis, we will identify the scenarios where ROMANSETU offers the most effective and beneficial solution compared to the other approaches. This may include:

* Situations with limited target language data
* Languages with complex scripts and limited established romanization systems
* Specific LLM tasks where romanization benefits understanding

**4. Expected Outcomes**

This project is expected to contribute to a deeper understanding of:

* The relative effectiveness of different LLM adaptation methods for non-English languages.
* The strengths and weaknesses of each approach under various conditions.
* The scenarios where ROMANSETU exhibits the most potential for effective and beneficial LLM adaptation.

**Idea - Extending ROMANSETU's idea to fine-tuning for specific tasks directly using Romanized low-resourced languages (say Japanese)**

Can experiment with fine-tuning pre-trained language models (like BERT) on our custom romanized Japanese text for specific NLP tasks and languages - can choose a niche domain here as well. We can then compare this with the results of a

* Japanese to English translated model
* Japanese-only model or
* pre-trained on romanized Japanese model

The metric will depend on the specific task chosen. This will help in determining how crucial the continual pre-training step actually is, and whether decent enough results can be obtained without it (would make sense if they’re not as good). If this is the case, then can this computationally expensive step potentially be removed from the pipeline?

**Idea - Extending ROMANSETU's work to compare the results of romanized/transliterated text to translated text**

By performing our own custom experiments, we can determine whether romanization outperforms translation. In the latter, a non-English language would first be translated to English, and then fed into models which have already been pre-trained on large English corpora. Since translation is a relatively main-stream problem, it is possible that translation schemes are more efficient than the romanization ones.

## **Idea #3**

## **Project Proposal: Resume Builder Automator**

**1. Introduction**

In the competitive job market, crafting a resume that stands out to potential employers is crucial. The Resume Builder Automator (ReBA) aims to streamline the process of creating customized resumes by automatically generating them based on job descriptions. This tool leverages Natural Language Processing (NLP) to identify key skills and requirements from job postings and aligns the user's experience and skills accordingly.

**2. Project Objectives**

**Automated Customization:** Develop an NLP model that can parse job descriptions and identify essential qualifications, skills, and attributes.

**User Profile Matching:** Create a system that matches the user's profile, including work experience, education, and skills, with the job description's identified requirements.

**Dynamic Resume Generation:** Automate the generation of tailored resumes that highlight the user's qualifications most relevant to the job description.

**User Interface: (Maybe just chat based)** Design a user-friendly interface that allows users to input their professional profiles and select job descriptions for which they wish to apply.

**3. Methodology**

**3.1 Data Collection and Preprocessing**

* Job Descriptions: Aggregate a diverse dataset of job descriptions from various fields and levels of seniority.
* User Profiles: Collect sample user profiles to train and test the system, ensuring data privacy and ethical use.

**3.2 NLP Model Development**

Skill and Requirement Extraction: Use NLP techniques such as Named Entity Recognition (NER) and keyword extraction to identify key skills and requirements from job descriptions.

Profile Matching Algorithm: Develop an algorithm that maps the user's skills and experiences to the job requirements, prioritizing matches and identifying gaps.

**3.3 Resume Generation Engine**

Template Design: Create multiple resume templates catering to different job sectors and preferences.

Dynamic Content Placement: Implement logic to dynamically populate templates with user information tailored to the job description's requirements.

**3.4 User Interface Development**

Interactive Dashboard: Develop an intuitive dashboard where users can upload their profiles, paste job descriptions, and receive tailored resumes.

Feedback Mechanism: Incorporate a feature for users to provide feedback on the generated resumes for continuous improvement.

**4. Expected Outcomes**

* Efficiency: Significantly reduce the time and effort required to customize resumes for different job applications.
* Increased Match Potential: Enhance users' chances of being shortlisted for interviews by aligning their resumes more closely with job descriptions.
* Scalability: Create a scalable solution that can adapt to various industries and job levels.

**5. Conclusion**

The Resume Builder Automator project represents a significant step towards leveraging AI in the job application process, making it more efficient and tailored to individual needs and opportunities. By automating the customization of resumes based on specific job descriptions. It aims to empower job seekers and enhance their prospects in a competitive market.

## **Idea #4 (Large Action Models)**

## **Project Proposal: LLM-Based Call Screening Automation for Smartphones (SmartScreen)**

Reference Project:<https://autodroid-sys.github.io/>

**1. Introduction**

In the era of ever-increasing communication, managing incoming calls efficiently while maintaining privacy and minimizing interruptions has become a significant challenge for smartphone users. The SmartScreen project proposes the use of advanced Large Language Models (LLMs) to automate call screening, enabling smartphones to intelligently handle calls based on context, caller identity, and user-defined rules.

**2. Project Objectives**

Automated Call Screening: To develop an LLM-based system capable of understanding the context and intent of incoming calls to screen them automatically.

Customizable User Preferences: To allow users to set preferences and rules for handling different types of calls (e.g., work, personal, spam).

Privacy Protection: To ensure the system respects user privacy and data security, using minimal and encrypted data processing.

Integration and Usability: To create a seamless integration with existing smartphone platforms and ensure ease of use for a broad user base.

**3. Methodology**

**3.1 System Design and Development**

LLM Selection and Customization: Choose an appropriate LLM for understanding and generating natural language, customizing it for telephony contexts.

User Interface (UI) Development: Design an intuitive UI for users to set up their call screening preferences and review call logs.

Privacy and Security Framework: Implement robust data handling and processing protocols to protect user privacy and comply with regulations.

**3.2 Integration with Telephony Systems**

API Development: Develop APIs for integrating the LLM system with smartphone telephony functions (e.g., call receiving, SMS sending).

Call Handling Logic: Create logic for the system to decide how to handle each call based on the LLM's analysis and user preferences.

**3.3 Testing and Iteration**

Pilot Testing: Conduct pilot testing with a closed group of users to collect feedback on system effectiveness and user experience.

Iterative Improvement: Refine the system based on feedback and testing results, focusing on improving accuracy, privacy protection, and user interface.

**4. Expected Outcomes**

Efficient Call Management: A sophisticated call screening system that automates handling of incoming calls, reducing interruptions and prioritizing important calls.

Enhanced User Control: Users gain greater control over their communication, with the ability to customize how different types of calls are managed.

Privacy Assurance: The system is designed with a strong emphasis on user privacy, ensuring sensitive data is handled securely.

**5. Conclusion**

By leveraging the capabilities of LLMs for understanding and responding to natural language, the SmartScreen project aims to redefine how smartphone users manage their incoming calls. This project not only promises to enhance productivity and privacy for individuals but also sets a precedent for the application of advanced AI technologies in everyday communication tools.

## **Idea #5 (Large Action Models)**

## **Project Proposal: Evaluating LLMs for Tool Selection Guidance**

Reference on how to benchmark:<https://github.com/HowieHwong/MetaTool>

Current LLM leaderboard: <https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard>

#### 1. Introduction

The rapid advancement in Large Language Models (LLMs) has opened new avenues for automating decision-making processes, including the selection of appropriate tools for specific tasks. This project aims to analyze the capabilities of state-of-the-art LLMs in recommending the correct tooling for given tasks, comparing their performance across various domains and tasks with the latest open-source LLMs available. The goal is to understand how these models can facilitate decision-making in software development, data science, and beyond, providing insights into their applicability, accuracy, and efficiency.

#### 2. Project Objectives

* Capability Analysis: To assess how well LLMs can identify and recommend the most suitable tools for a range of tasks based on given specifications.
* Comparative Evaluation: To compare the performance of various LLMs, including the latest open-source models, in selecting appropriate tooling.
* Usability and Integration: To evaluate the potential for integrating LLM recommendations into development workflows, enhancing productivity and decision-making.
* Best Practices and Guidelines: To develop best practices and guidelines for leveraging LLMs in tool selection processes.

#### 3. Methodology

##### 3.1 Selection of LLMs and Tasks

* LLM Selection: Choose a diverse set of LLMs, including leading commercial models and the latest open-source alternatives, for evaluation.
* Task Definition: Define a broad range of tasks across different domains (e.g., software development, data analysis, content creation) for which tool selection is critical.

##### 3.2 Development of Evaluation Framework

* Criteria Definition: Establish evaluation criteria, including accuracy of tool recommendation, reasoning capability, adaptability to task specifics, and user satisfaction.
* Benchmark Creation: Develop or identify benchmark datasets for each task, including specifications and correct tooling options, to test LLM recommendations.

##### 3.3 Comparative Analysis

* Performance Testing: Conduct systematic testing of each LLM's recommendations against the benchmark datasets, recording accuracy and any insights provided.
* User Experience Study: Incorporate feedback from potential end-users to assess the practicality and integration of LLM recommendations into real-world workflows.

##### 3.4 Best Practices and Integration Strategies

* Guidelines Development: Based on the analysis, develop guidelines for effectively using LLMs in tool selection.
* Integration Recommendations: Propose strategies for integrating LLM recommendations into existing tool selection processes and workflows.

#### 4. Expected Outcomes

* Performance Insights: Detailed insights into the capabilities and limitations of various LLMs in providing accurate tool recommendations.
* Comparative Analysis: A comprehensive comparison of the performance of leading and open-source LLMs in the context of tool selection.
* Practical Guidelines: A set of best practices and guidelines for leveraging LLMs to enhance decision-making in tool selection.
* Integration Strategies: Recommendations for integrating LLM-based tool selection into development and operational workflows.

#### 5. Conclusion

This project seeks to explore the frontier of AI-assisted decision-making, specifically in the context of selecting the correct tools for various tasks. By conducting a comparative analysis of the latest LLMs, this research will offer valuable insights into their practical application in improving efficiency and decision-making in software development and other domains. The outcomes will not only contribute to the academic and practical understanding of LLMs' capabilities but also provide actionable guidance for integrating these technologies into everyday workflows, potentially transforming how decisions are made in technical environments.