The Translatinator

A Natural Language Learning Translation System

Michael Olacsi
Boise State University
Boise, ID, USA
michaelolacsi@u.boisestate.edu

Bryce Kratzer
Boise State University
Boise, ID, USA
brycekratzer@u.boisestate.edu

Brian Wu
Boise State University
Boise, ID, USA
brianwu@u.boisestate.edu

Project Goal

This project aims to develop a machine translation system that seamlessly converts spoken English into Spanish and vice versa. It transcribes spoken input to text, translates that text into the target language, and can optionally provide an audio pronunciation of the translation. This interactive tool is designed to bridge communication gaps between speakers of different languages through robust back-end processing and a responsive web-based interface.

I. Introduction

Imagine a world where language barriers dissolve through the power of advanced translation systems. In that realm, state-of-the-art neural architectures decode and reconstruct linguistic nuances, delivering translations with natural flow and remarkable precision.

Our project doesn't quite achieve that, but it's still a powerful and useful tool. It combines powerful open-source models with modern web technologies and presents translations via an interactive, user-friendly web application. The front end is built with HTML, CSS, and JavaScript, while the back end is powered by Flask to ensure seamless communication between the user interface and our machine learning models.

II. Model Architecture

1) **Speech Recognition**
   - **Objective:** Capture and transcribe spoken input.
   - **Tool:** Whisper AI – an open-source model for converting speech to text.
2) **Text Translation**
   - **Objective:** Translate transcribed text into the target language.
   - **Tool:** MarianMT – a translation model for multiple language pairs.
3) **Text-to-Speech**
   - **Objective:** Convert translated text back into spoken language.
   - **Tool:** Coqui TTS – integrated optionally for audio output.

III. Technical Implementation

Back-End Processing

- **Data and Resources**
  - **Languages:** English and Spanish.
  - **Data Sources:** Utilize existing datasets for speech recognition, translation, and TTS. Specific datasets will be chosen during preliminary research.
- **Machine Learning Techniques & Libraries**
  - **Fine Tuning:** TensorFlow, PyTorch, and Keras.
  - **Data Handling:** Pandas, NumPy.
  - **NLP and Tokenization:** Hugging Face Transformers and associated tokenizer libraries.
  - **Evaluation:** SacreBLEU for translation quality, Matplotlib for visualization, plus human evaluation.

Web/Mobile Application Development

- **Back-End Framework: Flask**
  - Manages routing, server-side logic, and API endpoints.
- **Front-End Technologies: HTML, CSS, and JavaScript**
  - **HTML:** Structures the web pages with forms for audio input and areas for translation output.
  - **CSS:** Crafts a responsive and visually appealing interface.
  - **JavaScript:** Handles dynamic content, asynchronous API calls, and audio capture integration.
- **Technical Skills Needed**
  - **Programming and Scripting:** Proficiency in Python, including experience with TensorFlow, PyTorch, and Hugging Face Transformers.
  - **NLP and Machine Learning:** Understanding of neural network architectures, sequence-to-sequence models, fine-tuning methods, and evaluation metrics.
  - **Speech and Audio Processing:** Familiarity with speech recognition systems, audio data handling, and model fine-tuning.

  - **Web Development:** Expertise in HTML, CSS, JavaScript, and Flask to build interactive, full-stack web applications.
  - **Data Management:** Skills in data cleaning, preprocessing, and augmentation using tools like Pandas and NumPy.
- **Compute Resources:** Secured access to GPU resources (Borah and other GPUs) for efficient model training and fine-tuning.

IV. Project Timeline

- **Week 1 (Mar 10–12): Preliminary Research & Proposal Finalization**
  - Finalize the proposal and conduct research on Whisper AI, MarianMT, and speech-to-text methodologies.
- **Week 2 (Mar 13–19): Speech Recognition Development**
  - Collect and label data for fine-tuning Whisper AI.
  - Design and implement the audio capture and processing pipeline.
  - Initiate data gathering for translation fine-tuning.
- **Weeks 4–6 (Mar 27–Apr 16): Integration & Web/Mobile Development**
  - Complete the text output pipeline.
  - Develop the web/mobile application with Flask and integrate front-end components.
  - Implement features such as live audio capture, real-time translation display, and asynchronous communication.
  - *Optional:* Begin fine-tuning Coqui TTS for audio output.
- **Week 7 (Apr 17–23): Testing and Refinement**
  - Debug and optimize both the ML pipelines and the web interface.
  - Finalize documentation and prepare user guides.

V. Prototype Design

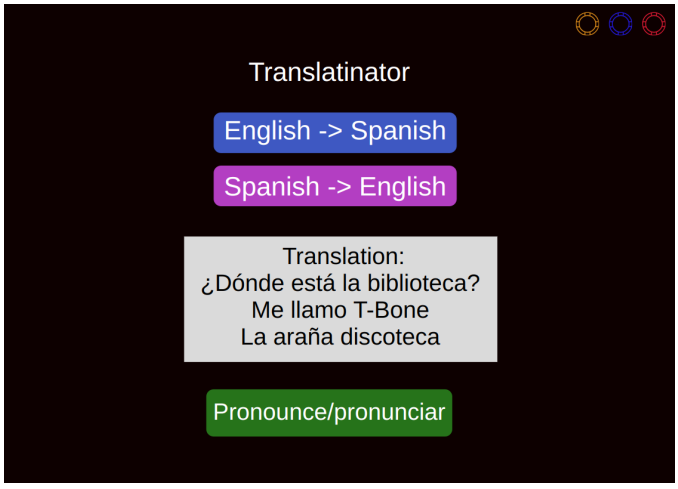

Fig. 1. Prototype of the Translatinator website.