PyLinguist: A Python code to Hindi code translator

Authors: *Ankit Kumar, Antara Tewary, Homa Haghighi*

Introduction

- **Problem:** Translation of Python code to Hindi is challenging due to the lack of a comprehensive translation system.
- **Objective:** Develop a translation system that accurately converts English Python code to Hindi.
- **Approach:** A hybrid translation system combining keyword preservation and GPT-4o-mini.

Motivation

- **Accessibility:** Facilitate learning and understanding of Python programming for Hindi speakers.
- **Inclusivity:** Enable Hindi speakers to access Python resources and contribute to the programming community.
- **Efficiency:** Enhance productivity by providing a comprehensive translation system for Python code.

System Architecture

- **Frontend:** Streamlit-based web interface with dual code editors.
- **Translation Engine:** Hybrid system combining keyword preservation and GPT-4o-mini.
  - Dictionary based keyword translation
  - Deep translation using google translate to translate non-keyword elements
  - Syntax preservation mechanism
- **Data Management:** checkpointing and logging for translation data.

Technical Implementation

- **Stage 1 - Base Translation:** Implementation of a hybrid translation system combining keyword preservation and Google Translate:
  - Keyword dictionary maintains Python-specific term translations
  - Parser identifies code components (variables, comments, strings)
  - Google Translate handles non-keyword elements
- **Stage 2 - GPT Enhancement:** Example-based translation using GPT-4o-mini:
  - Utilizes successful translations from Stage 1 as examples
  - Performs partial translation using keyword dictionary
  - GPT model completes translation based on examples
- **Stage 3 - Quality Assessment:** Comprehensive evaluation framework:
  - Back-translation verification
  - Syntax validation
  - BLEU score evaluation
  - Semantic preservation testing

Challenges

- **Syntax Preservation:** Maintaining code structure and functionality during translation.
- **Keyword Translation:** Accurate translation of Python-specific terms to Hindi.
- **Evaluation:** Developing a robust framework for translation quality assessment.

Evaluation Framework

- **Back-translation Verification:** Translation of Hindi code back to English for comparison.
- **Syntax Validation:** Verification of code structure and functionality.
- **BLEU Score Evaluation:** Comparison of translated code with reference translations.
- **Semantic Preservation Testing:** Evaluation of code functionality and logic.

Results

- Successful translation of Python code for Hindi languages.
- Preservation of code functionality and structure.
- High accuracy in translation of Python keywords.
- Robust evaluation framework for translation quality assessment.
- potential for further improvement and expansion.

Conclusion

- **PyLinguist:** A comprehensive translation system for English Python code to Hindi.
- **Future Work:**
  - Expand translation system to other languages
  - Improve translation accuracy
  - Develop a user-friendly interface

Stage-1: Initial Translation
500k python code
Keyword Dictionary
Google Translate
Merge Translations
Hindi Code v1
Stage-2: GPT Enhancement
Example Pairs
GPT Model
Enhanced Hindi Code
English Python Code
Keyword Dictionary
Stage-3: Quality Evaluation
Back Translation
Compare with Original
Quality Metrics
Semantic
BLEU
Syntax

Email → {akumar37, atewary, hhaghigh} @gmu.edu