Outline of Mandarin-English Code-Switching Final Project

Introduction

- In this project, we will be analyzing two datasets of Mandarin-English code-switching data from the SEAME dataset , one of them are majority-English and the other one are majority-Mandarin. We will be analyzing the syllables of words in the dataset, the part of speeches of the dataset, the phrases and sentence structures that occurs in the dataset. We will also be identifying word boundaries, sentence boundaries, code-switching boundaries in this dataset, as well as calculating the syllable length of each English words in the dataset. These data will be collected in order to answer a list of questions provided below.

Questions to investigate

- Does single-syllable English words or multi-syllables English words occurs more in Mandarin-English code-switching dataset? Is there a difference in the syllable count between the majority-Mandarin dataset, and the majority-English dataset?
    - Hypothesis: The amount of single-syllable words and the amount of multi-syllable words will have a similar amount, and the difference between the two datasets might not be predictable.
- Out of the multisyllabic English words scattered in English speech, what proportion of them conform with the Mandarin syllable template inventory?
  At what part of speeches (a.k.a., lexical category) does Mandarin-English code-switching usually occurs?
    - Hypothesis: Code-switching occurs mostly as nouns, and adjectives, rarely as verbs.
- Are some lexical categories of non-native, switched words more frequently recollected than others in spontaneous fluent mixed-language speech?
  This project will concentrate on analyzing the part of speech distribution of the switched, non-native words.
    - Hypothesis: Considering that retrieval rate was more rapid for non-native verbs and that retrieval accuracy was more rapid for non-native nouns, a higher prevalence of non-native verbs suggests retrieval rate more significantly facilitates use in code-switching.
    - Oppositely, a higher prevalence of non-native nouns hints that retrieval accuracy more significantly facilitates use in code-switching.

Background

- Picture Naming Experiment
  - In the article titled “A Comparison of Noun and Verb Retrieval”, 21 Mandarin-English bilinguals native in only Mandarin and 21 English monolinguals recalled nouns in response to sketches of objects and recalled verbs in response to sketches of actions.
  - Each noun or verb picture naming task was rated 0 for inaccurate and 1 for accurate, plus the retrieval time for each task was also logged.
  - Mandarin-English bilinguals scored higher for verbs in terms of retrieval rate but higher for nouns in terms of accuracy.

Methods

- **Here is the dataset that we will use for the project:**

  - SEAME is a Mandarin-English code-switching corpus replete with 192 hours of interview question-facilitated casual conversations. SEAME-dev-set contains two repackaged SEAME-derived test sets of mixed-language speech, one Mandarin-dominated and one Singaporean English-dominated. Keep in mind that SEAME only reflects only code-switching in Mandarin diasporan communities of Singapore and Malaysia. Any unexpected discrepancies between part of speech distribution in SEAME-dev-set and the noun and word retrieval experiment could result from different populations surveyed or different circumstances. The word retrieval in SEAME transpired in conversation unlike how word retrieval in the noun and word retrieval experiment occurred in response to pictures.

Data Collection:

The data collection and curation process is as follows:

- One Mandarin-dominated test set (dataset1.txt), wherein presumably English is the non-native language, and one English-dominated test set (dataset2.txt), wherein presumably Mandarin is the non-native language, were imported from zengzp0912's repository.
- We will manually convert both dataset1.txt and dataset2.txt to a csv with a NumberTag column and a MandEngSpeech column, so they can be processed by the computer easier.
- We will read each MandEngSpeech column into a list, titling the list mandEngSpeech for dataset1.txt and MandengSpeech for dataset2.txt.

    - We will filter out the English fillers (e.g., "um", "eh"); interjections or abrupt sentence-initial words like "ah," "ha", and "hey"; and proper nouns (e.g., Backstreet Boys) from each string in each list by iterating through lists of fillers, interjections, and proper nouns.
    - Max will use his knowledge of the Chinese language to identify fillers, interjections, and proper nouns in Mandarin, then we will filter out those from the both lists of mixed Mandarin-English sentences.

Algorithms and Pseudocode:

- Here is the algorithm of the project
- This algorithm might be heavily modified later during the project based on needs.
- This project will have 5 difference parts of analysis
    - 0. Read in datasets
    - 1. Dependencies import
    - 2. Sentence tokenize
    - 3. Syllable counts
    - 4. Part of speech tagging
    - 5. Sentence parse tree generation

```
# Input files
Read in the two datasets from file and store them in two lists

# Dependencies import
Import nltk, download popular, and import spacy

# Word tokenize
word tokenize each string, decomposing it into words tokens and punctuation mark tokens.
We will append each tokenized string of mandEngSpeech to mandEngSpeech_tokens
and each tokenized string of MandengSpeech to MandengSpeech_tokens.

# Syllables counting
Count the amount of single-syllables english words,
and the amount of multi-syllables English words in the dataset


# Part of speech tagging
Use spacy to tag all of the words in each sentences of the dataset and generate
a list of lists.

# Count the number of noun and verb tokens in the dataset
initialize noun counter
initialize verb counter
for item in mandEngSpeech_tokens:
        count the number of nouns in item
                add to noun counter
        count the number of verbs in item
                add to verb counter

print(noun counter)
print(verb counter)

# Parse tree generation
Use nltk and the tagged data to generate a parse tree for visual analysis
```