Tanish Lad (2018114005)

Jashn Arora (2018114006)

Automatic Generation of Code Mixed Sentences plus issues in Automatic Generation

Mentor Name: Pruthwik Mishra

Project Report

Project Description

Code Mixed Sentences have increasingly become more and more a part of our daily life. Almost every person knowingly or unknowingly uses English Words while speaking Hindi and Hindi Words while speaking English. Our task was to generate these code mixed sentences (mixing English and Hindi) using raw English and Hindi sentences data given to us.

How Our System Works

1. **Filtering of Raw Data**

   We were given a dataset of a large number of English Sentences and their corresponding Hindi Translations.

The first step in building the system was filtering the data so as to make a data that satisfies the bare minimum requirements for the next step.

We first removed the sentences where there were no Hindi Translations corresponding to the English Sentences and vice-versa.

The filtering of the remaining data was achieved by extensively using **RegEx** which helped us highly in cases where there were inaccuracies in the punctuation marks of the given raw data.

2. **Chunking the Raw Data**

   The next step was chunking the raw English and Hindi Data. To chunk the English Sentences, we used Stanford Parser (v 3.9.2), and for chunking the Hindi Sentences, we used LTRC Hindi Shallow Parser (v 4.0).

3. **Giza++ (the silent helper)**

   This step was performed simultaneously with step 2. The output (which was of our use) was a file that contained the word alignments and the probabilities. The strategy then suggested by our Mentor was that without thinking of Probability, find pairs that match both when first English was considered as the source language, and then Hindi was considered as the source language and filter them out.

4. **Finding Heads of NP and VP Chunks**

   The Next Step was extraction heads of the chunks of Hindi, and finding heads of chunks of English. The heads of chunks of Hindi were already present in the output that the LTRC Parser gave.

   To find the heads of the chunks of English, we employed a strategy where we found chunks containing tags that are from a given set of tags that could possibly act as a head, and if more than one such word is present which is capable of becoming a head, we chose the word that occurs at the last judging this generality from human observation.

5. **Combining Chunks by replacing Hindi Chunks in English Data and vice-versa**

   For this step, the strategy we employed was:

   > For each chunk of each sentence in Base Language, we find the chunk (if any such exists) of Target Language where the pair of heads of these chunk matches in the Giza++ filtered output.
   >
   > If we find such chunk, we replace the chunk of the base language with the chunk of the target language.
   >
   > In cases where no such replacement is possible, we just use the chunk of the base language.

   We ran this strategy considering firstly English as the Base Language and then considering Hindi as the Base Language.

Results

1. Base Language - **English**
    a. The Success
        i. सरकार is ready to raise हर संभव कदम to reduce कीमतों .
        ii. Potassium plays काफी अहम भूमिका in reducing blood pressure .
        iii. Amongst राज्यों receiving कृषि कर्मण पुरस्कार for the best farming , मध्य प्रदेश got कृषि कर्मण पुरस्कार for बेहतर प्रदर्शन in खाद्दान्न उत्पादन .
    b. The Failures
        i. But various complaints are आ रही हैं up in other territories of देश .
        ii. After this इसके बाद will announce its decision regarding ban on आयात or निर्यात फसले नियंत्रण .
    c. The No-Change Cases
        i. Apart from vitamin C , vitamin B complex , iron , phosphorus Sweet potato is rich in Beta.

2. Base Language - **Hindi**
    a. The Success
        i. People शौक से घरों या दफतरों में Bonsai लगाते हैं |
        ii. the plant जितना पुराना होगा The price उतनी ज्यादा होगी |
    b. The Failures
        i. इस वर्ष लगभग 653 करोड़ रुपये मूल्य के फूलों का exported हुआ था |
        ii. इसके अलावा डेज़ेर्ट रोजेज के पौधे को सर्दियों में यदि water नहीं भी give तो वह सूखता नहीं है |
    c. The No-Change Cases
        i. अब जरा ज़मीनी हकीकत पर नजर डाली जाए |

The No Change cases are the cases where no chunk is found whose head matches in the Giza++ output

Future Work

We plan on running a language model to find out the naturalness of the code-mixed sentences we get and extract the ones which are the most natural.

Github Repository Link

https://github.com/destinyson7/CL-1-Project

This repository contains all the work done by us.

References

[1] Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, Kalika Bali. ***Language Modeling for Code-Mixing: The Role of Linguistic Theory based Synthetic Data (2018)***

[2] Rafiya Begum, Kalika Bali, Monojit Choudhury, Koustav Rudra, Niloy Ganguly. ***Functions of Code-Switching in Tweets: An Annotation Scheme and Some Initial Experiments (2016)***

[3] Kalika Bali, Jatin Sharma, Monojit Choudhury, Yogarshi Vyas. ***"I am borrowing ya mixing ?" An Analysis of English-Hindi Code Mixing in Facebook (2014)***