Language Typology and Universals Project

A computational approach to understanding how conditional clauses work in Telugu and English. Created for the course project of the Spring '21 Course - Language Typology and Universals.

Introduction

Problem Statement - Identify and search for patterns in conditional constructions in Hindi and Telugu sentences and compare them with English.

Current Knowledge

Conditional sentences are sentences that express one thing contingent on something else, e.g. "If it rains, the picnic will be cancelled". They are called because the impact of the main clause of the sentence is conditional on the dependent clause. A full conditional thus contains two clauses: the dependent clause expressing the condition, called the antecedent (or protasis); and the main clause expressing the consequence, called the consequent (or apodosis).

Types Of Conditionals

Zero Conditional

> **Form** - *(If + Present Simple, ... Present Simple)*

> **Example** - If you boil water to hundred degrees, then it boils.

First Conditional

> **Form** - *(If + Present Simple, will + Infinitive)*

> **Example** - If it rains tomorrow, then we will go to the cinema.

Second Conditional

> **Form** - *(If + Past Simple, ... would + Infinitve)*

> **Example** - If I had a lot of money, I would travel around the world.

Third Conditional

> **Form** - *(If + Past Perfect, ... would + have + Past Participle)*

> **Example** - If I had gone to bed early, I would have caught the train.

Literature Review

- Structure of Hindi Conditional Clauses
- Paper on Hindi Conditional Clauses
- Paper on English Conditional Clause Forms

Tasks

- [x] Find Dataset of 500 English sentences with conditional clauses.
- [x] Translate the sentences into Hindi and Telugu.
- [x] POS Tagging for Hindi, English, and Telugu sentences.
- [x] Chunk the Telugu sentences.
- [ ] Identity the type of conditional clause according to the rules as seen in **Types of Conditionals**. **TODO**
- [ ] Find patterns and draw graphs to visualise the observations. **TODO**
- [ ] Write the final report. **TODO**

Directory Structure

The directory structure for the repository is as follows:

```
.
├── annotation
│   ├── data
│   │   ├── eng_pos_tags.txt
│   │   └── hin_pos_tags.txt
│   ├── eng.ipynb
│   ├── eng.py
│   ├── headers.py
│   ├── helper_hindi.py
│   ├── hindi.ipynb
│   └── hindi.py
├── data
│   ├── austen-emma.txt
│   ├── austen-persuasion.txt
│   ├── austen-sense.txt
│   ├── bible-kjv.txt
│   ├── blake-poems.txt
│   ├── bryant-stories.txt
│   ├── burgess-busterbrown.txt
│   ├── carroll-alice.txt
│   ├── chesterton-ball.txt
│   ├── chesterton-brown.txt
│   ├── chesterton-thursday.txt
│   ├── edgeworth-parents.txt
│   ├── melville-moby_dick.txt
│   ├── milton-paradise.txt
│   ├── shakespeare-caesar.txt
│   ├── shakespeare-hamlet.txt
│   ├── shakespeare-macbeth.txt
│   └── whitman-leaves.txt
```

```
├── dataset_create.py
├── eng_conditional_sentences.txt
├── final_hindi_sens.txt
├── hin_conditional_sentences.txt
├── LICENSE
├── README.md
└── translate_sentences.py
```

Creating The Dataset

The corpus was created by taking sentences from the NLTK corpus of the *Gutenberg Library*. The sentences were then filtered out so taht only sentences that were of the conditional clause format were added. The code for this can be found in `dataset_create.py`. The list of books from which the data was scraped can be found in the directory `data`.

Code Snippet for converting the data:

```
for text in texts:
    f = open('data/' + text, 'r')
    sentences = f.readlines()
    sentences = [sentence.rstrip() for sentence in sentences]
    for tsentence in sentences:
        sentence = list(tsentence.split(" "))
        if sentence[0] == "If":
            final_data.append(tsentence)

f = open('conditional_sentences.txt', 'w')
for sentence in final_data:
    f.write(sentence + '\n')
```

Translation Of The Sentences Into Hindi and Telugu

The sentences were translated using the `google_trans_new` library. This library connects to the Google Translate API and provides a translation into the destination language. The accuracy of the translator was not very good and the sentences that had inaccurate translations were then corrected. The code for the translations can be found in `translate_sentences.py`.

Code Snippet for translating the sentences into Hindi:

```
f = open("eng_conditional_sentences.txt", 'r')
sentences = f.readlines()
sentences = [sentence.rstrip() for sentence in sentences]

g = open("final_hindi_sens.txt", 'w')

for sentence in sentences:
```

```
    g.write(translator.translate(sentence, lang_src='en', lang_tgt='hi') +
'\n')
```

POS Tagging

The POS tagging was done using Stanza, a python NLP package, which comes with an in-built POS-Tagger. Each word in every sentence of the dataset was tagged and stored in the format as seen in engtags in annotation/eng.py. A similar approach was followed for Hindi. The tags along with the tokens and sentences were then stored in annotation/data/eng_pos_tags.txt. The output was created by a helped function in annotation/headers.py. If the output was inaccurate, it was manually corrected.

Code Snippet for POS tagging sentences in English:

```
for sentence in engdata:
    engdoc = nlp(sentence)
    bigtemp = {}
    tokens = []
    for token in engdoc:
        temp = {}
        temp["word"] = token
        temp["POS_TAG"] = token.pos_
        tokens.append(temp)
    bigtemp["word_tags"] = tokens
    engtags.append(bigtemp)
```

Snippet of example output in annotation/data/eng_pos_tags.txt:

```
4. If he had said it of my wife , you English would yourselves have
pardoned me for beating him like a dog in the market place .

TAGS:

    1. If - SCONJ
    2. he - PRON
    3. had -    AUX
    4. said -   VERB
    5. it - PRON
    6. of - ADP
    7. my - DET
    8. wife -   NOUN
    9. , -  PUNCT
    10. you -   PRON
    11. English -   PROPN
    12. would - VERB
    13. yourselves -    NOUN
    14. have -  AUX
    15. pardoned -  VERB
    16. me -    PRON
```

```
17. for -   ADP
18. beating -   VERB
19. him -   PRON
20. like -  SCONJ
21. a - DET
22. dog -   NOUN
23. in -    ADP
24. the -   DET
25. market -    NOUN
26. place - NOUN
27. . - PUNCT
```

Contributors

> Tejasvi Chebrolu

> Padakanti Srijith