CS689A COMPUTATIONAL LINGUISTICS FOR INDIAN LANGUAGES - Assignment 01

Student Name : Saqeeb
Roll Number : 22111053

Installation guide

This assignment code requires python library like **string, re, nltk, numpy, collections, pytest, matplotlib, conllu, indic-transliteration**. Install these libraries and run the code. To install indic-transliteration and conllu library see the command below.

```
!pip install indic-transliteration
!pip install conllu
```

Solution 1

- Clean the text file
  - Remove emoticons
  - Remove punctuation marks
  - Remove \n and \t delimeters
  - Remove English alphabets
- Make correction in the unicode
- Convert the text to SLP1

Solution 2

- Create a list of vowels and consonants
- Create a list of risky characters i.e. character with unicode size of 2.
- Count Character, Syllable and tokens
- Count Token Bigram, Character Bigram and Syllable Bigram

Solution 3

Byte Pair Encoding Algorithm

- Create a dictionay of the corpus with the frequency of each token
- Create a vocab with the character of the corpus
- Add token to the vocab by combining the most frequent consecutive token
- Repeat 1K time to generate 1K vocab size
- Repeat 2K time to generate 2K vocab size
- Repeat 5K time to generate 5K vocab size
- Repeat 10K time to generate 10K vocab size

Solution 4

PRECISION = TP/(TP + FP) RECALL = TP / (TP + FN) F-SCORE = 2xPxR/(P + R)

Using NLTK metric python library calculated the precision, recall and f-score.

Solution 5

For extracting a list of lemmas and the corresponding surface forms found from the Universal Dependency tagged files use the python library conllu. Here I have created a dictionary to store the surface form and it's lemma.

Solution 6

Here I have plotted 4 different graphs.

1. frequency of whitespace-separated words vs it's rank.
2. frequency of Syllables vs it's rank.
3. frequency of Characters vs it's rank.
4. frequency of Lemmas vs it's rank.

Ignoring the outliers the text follows the Zipf's law : f ∝ 1/r i.e. f · r = constant
Note : The top two plots exactly follows the Zipf's Laws but the last two are deviated and doesnot exactly follow Zipf's Law.

Solution 7

Given a lemma and the corresponding surface form, derive the suffix. Do an end stripping from the surface form till the lemma or a subset of the lemma is reached. Here, I have created a list of the suffix that are present. Analysing the results recieved from this are convincing and in line with the desired result.

Correct suffix list = {'□□': 108, '□': 65, '□□□□': 31, '□': 22, '□': 6, '□': 5, '□□': 5, '□□': 4, '□□': 4, '□□': 3, '□□': 3, '□': 2, '□□□□': 2, '□□□': 2, '□': 2, '□□': 1}
Incorrect suffix list = {'□□': 5, '□□': 2,'□□□□□': 1,'□□□□□': 1, '□□□□□□□': 1, '□□□': 1,'□□□□': 1, '□□□□': 1, '□□□': 1}

The final result is add to result folder