Sentence Segmentation in Urdu

Muhammad Usman P19-0096

February 7, 2023

1 Introduction

Sentence Segmentation in Urdu through scratch no NLP library is used.

2 Segmentation's

2.1 Word Segmentation

```
def word_segmentation(text):
    words = []
    current_word = ""
    for character in text:
        if character.isspace():
            if current_word:
                words.append(current_word)
                current_word = ""
        elif character in ["", "!", "?"]:
            if current_word:
                words.append(current_word)
            words.append(character)
            current_word = ""
        else:
            current_word += character
    if current_word:
        words.append(current_word)
    return words

print(word_segmentation(text))
```

This part of the code is for performing word segmentation on an Urdu text. In the code, the wordSegmentation function takes a text string as input and splits it into words by iterating over each character in the text. If the character is a whitespace, the current word is added to the list of words, and the current word is reset to an empty string. If the character is one of the punctuation marks (””, ”!”, ”?”), the current word is added to the list of words, followed by the punctuation mark, and the current word is reset to an empty string. Otherwise, the character is added to the current word. After processing all characters, if the current word is not empty, it is added to the list of words. Finally, the list of words is returned.

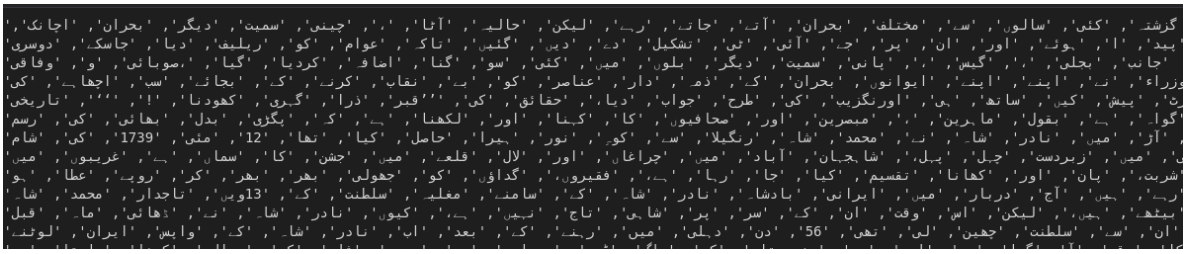

Figure 1: This is Output of the word Segmentation

2.2 Sentence Segmentation

```
def sentence_segmentation(text,stopwords):
    sentences = []
    mystr=""
    current_sentence = ""
    cout=1
    words = word_segmentation(text)
    for character in words:
        cout+=1
        current_sentence += character
        if character in stopwords and cout >5 :
            sentences.append(current_sentence)
            sentences.append(" ")
            current_sentence = ""
            cout=0

        if character not in stopwords:
            sentences.append(current_sentence)
            current_sentence = ""

    mystr=' '.join(map(str,sentences))
    return mystr

print(sentenceSegmentatpartion(text,stopwords))
```

This part of code is trying to segment text into sentences by identifying end words. The end words are passed as an argument stopwords. The words are first segmented using a wordSegmentation function.Then, for each word, the code checks if the word is in the stopwords list. If it is and if cout (a counter) is greater than 5, it appends the current sentence to the sentence list and append " - " in sentence and starts a new sentence. If the word is not in the stopwords list, it still appends the current sentence to the sentences list and starts a new sentence.

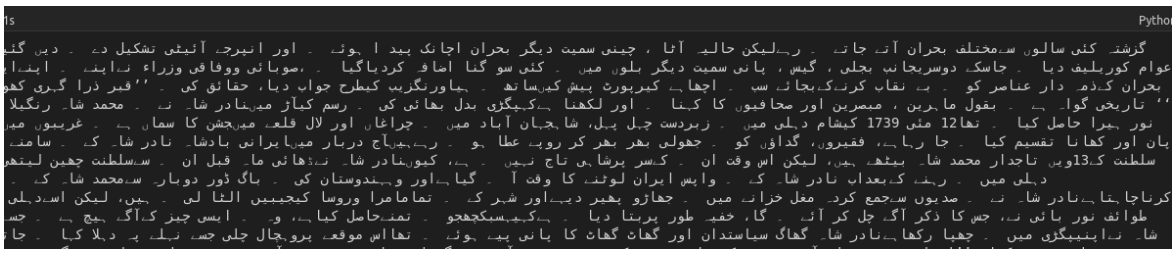

Figure 2: This is Output of the sentences Segmentation.