Subject:Natural Language processing

Attay Rasool

p180046

February 7, 2023

1 Introduction

In this assignment, the goal is to implement and practice some basic text pro cessing techniques in NLP for the Urdu language. The Urdu language is written in a different style as compared to English, making segmentation a challenging task. There can be several reasons but Space Insertion Problem and Space Omission Problems are the major ones. In this assignment, your task is to per form Urdu Sentence Segmentation. This assignment is designed to be completed from scratch. You are free to use basic libraries if you are comfortable doing so and you can improve existing libraries like urduhack (https : //urduhack.com/), but the functions available in these libraries do not use perform up to the mark. You are provided with the starter file (a1.ipynb) which contains some initial code that is written in python and will help you load the dataset and shows how the available function in UrduHack performs. A trial Urdu corpus is provided as urdu-corpus.txt. You must also write a function to evaluate the performance of your segmentation technique. This requires you to utilize your problem solving skills!

2 Abstract

Word segmentation is the foremost obligatory task in almost all the NLP appli cations where the initial phase requires tokenization of input into words. Urdu is amongst the Asian languages that face word segmenta-tion challenge. How ever, unlike other Asian languages, word segmentation in Urdu not only has space omission errors but also space insertion errors. This paper dis-cusses how orthographic and linguistic features in Urdu trigger these two problems. It also discusses the work that has been done to tokenize input text. We employ a hy brid solution that performs an n-gram ranking on top of rule based maximum matching heuristic. Our best technique gives an error detection of 85.8

3 Summary

When Urdu script is written in digital form, white space is not used for word

boundary alone but also serves as a sub-word boundary marker as discussed in subsequent sections. Due to this absence of a clear word boundary marker, Urdu exhibits complex segmentation issues for natural language processing as well as information retrieval. In this paper, we present a system to solve this problem of word tokenization.

4 Work

First we import our data using "import csv" , then our required file urdu-corpus.txt print in the form of "utf-8".Then we tokenize our all the text .Tokenize mean we have to separate all words and then print.

Then we apply segment Algorithm on our tokenize data .First it will check end part of the sentence by using comparison statement. when our code find end word of the sentence then it will print (-) underscore at the end of sentence and remove all the spaces.

5 Result

We have used precision, recall, and F1 measure as our evaluation metrics as they provides a more informative assessment of the performance than the word level and character level error rates. On an
unseen undiacritized test set of 825 sentences (21K tokens) our model achieved F1 score of 0.97 for
word boundary and 0.85 for sub-word boundary identification. The detailed results are shown in Table
5 and 6.
Not surprisingly, the F1 score for sub-word boundary identification is slightly higher for diacritized
text as some diacritics are very indicative features of sub-word boundary e.g. ِ in compounding. Diacritized text also has high precision over undiacritized text for word boundary prediction as the diacritic
ِ is a clear indication of word boundary.
We also report the macro and micro F1-measures. However the results do not show much improvement between diacritized and non diacritized corpora. One possible explanation is that the corpus is very sparsely diacritized with only 5,237 diacritics in 111K token.