HACETTEPE UNIVERSITY

COMPUTER ENGINEERING

BBM497 NATURAL LANGUAGE PROCESSING LAB.

ASSIGNMENT 1

Name: Kürşat Aktaş

Student Number: 21227949

Subject: Language models and smoothing

Content

1 Introduction

First of all I need to say that this is my firsy python program. For this reason my code may looks terrible, sorry for that.

In this paper I tried to explain what I had done and what I had observed with this assignment.

Because program displays all of the tasks results nicely, I thought I did not need to give screenshots.

2 File Hierarchy and Requirements

Before run the given program be sure that file hierarchy look like this;

- `ex.py`
- `data/`
  - `comedies/`
  - `historical/`

After that, program can be run with below command;

- `python3 ex.py`

3 Language Model

When designin my language model I have taken the following decisions;

a) I didn't used sentence and word boundaries. These tokens (!.?...) used for sentence seperations and single space used for word seperation.
b) My language model is case sensetive. So "The" and "the" has different frequencies.
c) Punctuation marks counted as separate words. But these formats not (I'm We're We'll etc.)

4 Tasks

Firstly I trained my program with files which are located in "data/comedies/" path. With the help these files, I create unigrams, bigrams and trigrams. I also created some other things for speed up my program, like unique n-gram counts.

4.1 Task 1

As shown in the code (ex.py) probabilities of given sentences calculated with smoothed bigrams.

The important part in here is smoothing operation, below line says that divide 1 to frequency of first element plus length of the unique token.

```
1 / (unigrams_dict[i] + len(uniq_unigrams)) (line20)
```

4.2 Task 2

In this task program creates 30 sentences (10 sentences per unigram, bigram and trigram) and calculates the probabilities of these.

The important part in here is selecting starting word of a sentence. For this purpose I had created “starting_word” list which holds all word tokens occured after end punctation marks (! . … ?). So bigram and trigrams models firstly selects one word randomly from this list and continues to generating sentences.

4.3 Task 3

Because of the utf problems in given files (in /data/historical path) I can only tested with file named “The Third Part of King Henry VI.txt” and the result didn’t satisfy me.