CS 458/535 - Natural Language Processing

Due Date: Sunday, May 9th by 11:55pm.

Assignments are to be done individually. No late assignments will be accepted.
Submissions that do not comply with the specifications given in this document will not be marked and a zero grade will be assigned.
Write your name and e-mail id in a comment line in top of your notebook in a Text cell. You are required to submit a single ipython notebook on Google Classroom. You should name your notebook as i19-XXXX.ipynb where i19-XXXX represents your student id.

Spell Correction for Roman Urdu

1 Introduction

In this assignment you will be developing spell correction for Roman Urdu. You will be correcting non-word errors using the Noisy Channel model. This spell correction technique has been widely used by word processors and is also being used by Google in its search engine. Whenever you type in a query with a misspelled word such as `corection` Google would instantly return the results for `correction` instead.

2 Background

In this assignment, you will implement a spell correction program for Roman Urdu in Python. Spell correction using the Noisy Channel model is based on Bayes Theorem as shown in Equation 1. Where w is the candidate word and x is the misspelled word.

$$P(w|x) = \frac{P(x|w) * P(w)}{P(x)} \tag{1}$$

In order to get the corrected word $\hat{w}$ we perform an `argmax` over all candidate words to pick the word with the highest probability. The $P(x)$ in the denominator is common for all candidate words and can be ignored. This gives us Equation 2.

$$\hat{w} = \texttt{argmax}_{w \in candidates} \; P(x|w) * P(w) \tag{2}$$

3 Data

This assignment requires two files **data.txt** which will serve as the corpus and **misspellings.txt** for generating error model tables. Make sure that you have access to both files before starting. **data.txt** is a collection of Roman Urdu sentences and **misspellings.txt** contains a list of words and their misspellings.

4 Implementation Details

The spell corrector you will be developing contains four main components. These components correspond to different parts of Equation 2 and are explained below:

1. **Language Model $P(w)$:** The language model gives the probability of each word occurring in the vocabulary V determined using the corpus given in **data.txt**. A simple unigram model should be trained using the provided file **data.txt**. You can assume that all words given in **data.txt** are correcly spelled.

2. **Error Model $P(x|w)$:** This model will be used to estimate the probability of typing x when w was intended. This error is to be modeled using character level *insert, delete, transpose* and *substitute* tables. You will be generating these tables using the provided list of common misspellings in Roman Urdu.

3. **Candidate Words Generation ($w \in candidates$):** The candidate model will give you all the possible words w which could have been mistyped as x. For this assignment, you should only consider words that are one edit away from x and exist in our vacabulary V determined by **data.txt**. Note that there could be many possible words for example for `kitaab` deleting the `b` would give `kitaa` which does not exist in Roman Urdu. Thus, for each error word you will be generating a set of words from V.

4. **Selection Model using argmax:** The final selection using **argmax** will be responsible for choosing the candidate word $\hat{w}$ which has the highest probability specified by $P(x|w) * P(w)$.

4.1 The Need for Generating Error Model Edit Tables

Generating possible combinations for correcting a misspelled word is alone not enough. You also have to know which of those combinations are actual words and how probable is the misspelling given the correct word. Those things will be given by the $P(w)$ Language model and the $P(x|w)$ Error model. Explaining in simple words, take for example a misspelled word: "usary". Generating possible candidates for the correct word you would get "usaye", "usaey", "usay", "usmay" etc. Which all are one edit away from "usary" and all seem correct. The Language model will tell you that words like "usaey" and "usaye" are not very probable and the Error model will tell you it is more likely to mistype "usay" as "usary" as compared to "usmay" so "usay" is the most likely correct word. The tables will be used to calculate $P(x|w)$ which you must have or will learn in class.

4.2 How to Generate Error Model Edit Tables

Originally as you might know these tabels are created using typing data from smartphones and PC's. But we don't have that kind of data available for Roman Urdu so you have to artificially create them using misspellings.txt. This text file contains a list of words, the first word in each line is the correct word and after the comma are all its common misspellings which are one edit away. You have to find what type of edit has been done from insert, delete, substitute and transpose and populate the corresponding table for each misspelt word. Remember Insertion and deletion are conditioned on the previous character. Following are some examples:

1. INSERT - Correct: "usay" Wrong: "usaye" Increment entry for "y" and "e" in insert table

2. DELETE - Correct: "usay" Wrong: "usy" Increment entry for "s" and "a" in delete table

3. SUBSTITUTE - Correct: "usay" Wrong: "usau" Increment entry for "y" and "u" in substitute table

4. TRANSPOSE - Correct: "usay" Wrong: "suay" Increment entry for "u" and "s" in transpose table

In addition to these tables you will also need a character level bigram model for calculating $count(x, y)$ where x and y occur consecutively for calculating $P(x|w)$ for delete and transpose tables.

5 Assignment Tasks

You are expected to complete the following assignment tasks. This list will be used in the marking scheme of this assignment.

1. Train a uni-gram model using the corpus provided in **data.txt**.
2. Create *i*nsert, delete, substitution and *t*ransposition tables for alphabets `a-z` using the provided **misspellings.txt**. The tables can be implemented using Python dictionaries.
3. Create a function that calculates $P(x|w)$ using the Error Model tables.
4. Create a function that returns the set of candidate words which are one insert, delete, substitute or transpose away from a given word x. The set of candidate words must be a subset of the vocabulary V. This means that for each candiate word w, $P(w) > 0$.
5. Create a function that takes a list of candidate words, $P(x|w)$, and $P(w)$ to return the most probable corrected word $\hat{w}$ for the misspelled word x.
6. Your solution must also be able to handle cases when there are no candidate words.

6 Instructions

The components of this assignment can easily be implemented in Python using its native data structures, libraries and functions. So it is recommended that you do **NOT** to use any external libraries for this assignment. Credit will be given for code structure, clean programming logic and proper documentation. Some Python data structures and libraries that you will find useful: Counter data structure and Regex

Honor Policy

This assignment is a learning opportunity that will be evaluated based on your ability to think in a group setting, work through a problem in a logical manner and write a research report on your own. You may however discuss verbally or via email the assignment with your classmates or the course instructor, but you are to write the actual report for this assignment without copying or plagiarizing the work of others. You may use the Internet to do your research, but the written work should be your own. **Plagiarized reports or code will get a zero**. If in doubt, ask the course instructor.