Natural Language Processing

Assignment 1

Instructions:

- Create a report for the assignment, including a detailed explanation for every question. Describe what you have done, why you used specific methods, and the results you obtained.
- You are not allowed to copy solutions from other students (same/cross section). If any sort of cheating is found, heavy penalties will be given to all students involved.
- Late submission of your solution is not allowed.
- Submit the notebook file (i.e., the .ipynb file).

Regular Expressions and Preprocessing

Install nltk in Python by following steps

1. Go to pip folder in Python installation path (Find python installation path by running "where python" in command prompt (cmd) in windows)

For example: cd C:\Program Files\Python\Python36-32\Scripts

2. Run "pip install –U nltk"
3. Run "nltk download()" (This command will download corpus etc. for nltk)
4. Run "pip install beautifulsoup4" (This software is needed for parsing html files) You can get help on using nltk for this homework from following link

https://www.nltk.org/book/ch03.html

Part 1:

Q1) Describe the class of strings matched by the following regular expressions:

a. [a-z]+
b. [A-Z][a-z]+
c. c[aeiou]{1,2}t

Test your answers using nltk.re_show(). (You will have to import libraries using import nltk, re, pprint.)

Part 2:

Q2) Write a utility function that takes a URL as its argument, and returns the contents of the URL, with all HTML markup removed. Use from urllib import request and then request.urlopen('https://www.dsu.edu.pk/contact-us/').read().decode('utf8') to access the contents of the URL. Use BeautifulSoup(html).get_text() to parse html.

Import the following for this question:

(from urllib import request
from bs4 import BeautifulSoup)

Q3) Tokenize text parsed from the above url using nltk. Find all phone numbers and email addresses from this text using regular expressions. (Do not tokenize text otherwise email

addresses will be incorrectly tokenized)

Q4) Use the Porter Stemmer to normalize some tokenized text, calling the stemmer on each word. Do the same thing with the Lancaster Stemmer and see if you observe any differences.

Use the following libraries for this question:
(import nltk
from nltk.stem import PorterStemmer, LancasterStemmer
nltk.download('punkt'))

Part 3:
Q5) Explore the following advanced NLP techniques and concepts using the provided Urdu dataset (Sentiment Dataset Urdu):

Handling Urdu Text:

- Ensure that the file is read using the correct encoding, typically utf-8, to properly handle Urdu characters.

a. SubwordTokenization
- Apply a subword tokenization technique to the provided Urdu dataset.

b. Byte Pair Encoding (BPE):
- Implement BPE on the Urdu dataset using built-in functions and analyze the resulting subword units.

c. MaxMatch Segmentation:
- Perform MaxMatch segmentation using built-in on a subset of the Urdu dataset.

After applying BPE and MaxMatch, compare their outputs and analyze which tokenizer performs better for Urdu text.

d. Language Modeling:

- Implement and evaluate the performance of each model using perplexity.
    1. Unigram Model
    2. Bigram Model
    3. Trigram Model
- After implementing the models, analyze and compare the outputs of each, and determine which model performs better based on perplexity.

(Use appropriate libraries and tools such as NLTK, Hugging Face, or others for implementation and analysis.)