Kshitija Shelar **Text Analytics COMP4760** Date: 10/10/20

20200155 Practical 2

1) Try to get used to using then Nltk package. Create a text file with,50 words in it (written as a collection of sentences) and make sure it has items that challenge the tokeniser (e.g., I.B.M. and other weird things). But you need to make these up yourself, so you think about what it is doing.

a. Load the file in and use nltk.word_tokenizer() on it. Report the list of tokens that are produced from it and note any oddities that arise.

Ans:

Original text file data	Tokenized data
this is a test file for tokenizer assignment from T.A (Text analytics) module. i want to perform many actions, including, but not limited to the following: removing stop words, stemming ,pos tagging etc. I'd like to use complex words like I.B.M and U.S.A just to confuse the library functions. after performing these actions, I'll also analyze the results. I could've said Goodbye! So I'm saying it now	['this', 'is', 'a', 'test', 'file', 'for', 'tokenizer', 'assignment', 'from', 'T.A', '(', 'Text', 'analytics', ')', 'module', '.', 'i', 'want', 'to', 'perform', 'many', 'actions', ',', 'including', ',', 'but', 'not', 'limited', 'to', 'the', 'following', ':', 'removing', 'stop', 'words', ',', 'stemming', ',', 'pos', 'tagging', 'etc', '.', 'I', "'d", 'like', 'to', 'use', 'complex', 'words', 'like', 'I.B.M', 'and', 'U.S.A', 'just', 'to', 'confuse', 'the', 'library', 'functions', '.', 'after', 'performing', 'these', 'actions', ',', 'I', "'ll", 'also', 'analyze', 'the', 'results', '.', 'I', 'could', "'ve", 'said', 'Goodbye', '!', 'So', 'I', "'m", 'saying', 'it', 'now']

- The tokenized text is the output to the method data_token=nltk.word_tokenize(data) where data is the content from text file.
- Oddities observed:
  - The words I'd, I'll, could've and I'm got converted to 2 words i.e I and 'd and to could and 've
- Other thing observed, too many punctuations is actually confusing and it's also taking up space and frequency as these occur more frequently (specially .). It would help if punctuations are removed specially when word count is of utmost importance.
- To fix the oddities observed, maybe we can write a method which removes the punctuations as well as write a separate function to fix apostrophe words such that it either splits into two whole words or skip one of it as and how it makes sense. i.e I've should become I have or I. Since most of such words are stop words, we can remove them later using stop words removal functionality.

b. Now, take this output from the tokenizer and do the normalization step

Original text file data	Normalized
words like I.B.M and U.S.A just	'words', 'like', 'i.b.m', 'and', 'u.s.a', 'just'

- Used .lower to convert it into lowercase. Other type of normalization consists of stop words removal, punctuation removal etc. This normalization is really important to filter out unnecessary data so that processing of these words is avoided.

c. Now, take output from normalized step and run it through pos-tagger. Report this output as your answer and highlight any inaccuracies that occur at this stage.

input	POS-tagged output
‘i.b.m’, 'and', 'u.s.a', 'just', ‘the’, ‘library’	('i.b.m', 'NN'), ('and', 'CC'), ('u.s.a', 'JJ'), ('the', 'DT'), ('library', 'JJ'),

- POS is part of speech tagging. Every tokenized word is assigned a tag depending on its usage example IBM here is a noun. This is done by stem_data.append(porter_stem.stem(word))
- Oddities observed here are U.S.A is tagged as adjective and Library is also tagged as adjective. However, both these words are actually noun.

---

2) Now, take a second, new-text-file do the following:

a. Tokenize the new-text-file (50 words) and the stem it using Porter Stemming. Report your Outputs and some of weird things that Porter Stemming does.

Ans:

Tokenized words	Porter Stemmed words
‘the’, ‘intention’, ‘of’, ‘this’, ‘file’, ‘is’, ‘that’, ‘we’, ‘will’, ‘perform’, ‘the’, ‘same’, ‘actions’, ‘,’, ‘but’, ‘more’, ‘specifically’, ‘emphasise’, ‘on’, ‘Lemmatization’, ‘and’, ‘Porter’, ‘Stemming’, ‘.’, ‘intervention’, ‘of’,	[‘the’, ‘intent’, ‘of’, ‘thi’, ‘file’, ‘is’, ‘that’, ‘we’, ‘will’, ‘perform’, ‘the’, ‘same’, ‘action’, ‘,’, ‘but’, ‘more’, ‘specif’, ‘emphasis’, ‘on’, ‘lemmat’, ‘and’, ‘porter’, ‘stem’, ‘.’, ‘intervent’, ‘of’,

- Stemming is used to convert words into their root form. This is important because when the frequency of words is considered, words which mean the same can be grouped together under one frequency bracket.
- Used porter stemmer to stem words in the file using stem_data.append(porter_stem.stem(word)).
- Observed that it does normalization by itself. Porter Stemmer got converted to porter stem.
- Oddities observed specifically gets converted to specif whereas the root word for specifically is specific. Intervention is also converted to intervent when the root word for it is intervene. Porter stemmer is just removing suffix/prefix sometimes and not converting into root words.

b. Tokenize the new text-file and then lemmatize it using WordNet Lemmatizer; note you may have to pos-tag the sentences first and then convert the tags to make this work. Report the result of these steps and point out some of the things that look wrong.

POS Tagged word	WordNet Lemmatized word
('hoping', 'VBG'), ('recommended', 'VBN'), ('specifically', 'RB'), ('papers', 'NNS')	“hop”,” recommend”,” specifically”,” paper”

- WordNet lemmatizer is used to convert words to root form too but it takes the context of the word in consideration. Example papers is converted into paper, its root form.
- For example, specifically remains specifically after using word net lemmatization. This should get converted into specific. Hoping is also converted into hop which is the wrong root word for hoping. It should get converted to hope.

c. Compare the outputs from Porter Stemming and the Lemmatisation of the same file. Which do you think is the best to use and why?

Porter Stemmed words	WordNet Lemmatized word
'specif', 'emphasis', 'on', 'lemmat', 'porter', 'stem', 'hope', 'recommend', 'paper', 'and', 'scissor', 'includ'	“specifically”, “emphasise”, “Lemmatization”, “Porter”, “Stemming”. “hop”,” recommend”,” paper”,“ scissors”, “include”

- We can observe lot of differences between Porter stemmer and wordnet lemmatizer. Like specifically become specif after stemming and remains specifically after lemmatization. Hoping becomes hope after stemming and hop after lemmatization. Scissors becomes scissor after stemming and remains scissors after lemmatization.
- I think, Lemmatization is better than stemming as it considers the context of word and the words post lemmatization makes sense. Some of the basic words become very confusing after performing stemming on it words like this becomes thi and including becomes includ after stemming and include after lemmatization. So when the text has to be pragmatically considered for analysis, lemmatization should be considered.