**INFORMATION RETRIEVAL**

**ASSIGNMENT 2**

[CS 5615](https://online.mrt.ac.lk/pluginfile.php/162467/mod_assign/introattachment/0/CS 5615.pdf?forcedownload=1)

Student No : 199378T

student name: R.B.Yeshani

GitHub link : https://github.com/BinuriYeshani/IR\_assignemnt

1. Brief description of the tokenizer, spell correctors, stemmer, and lemmatizer you used.

The given data set assignment\_data.txt is subdivided into 3 txt files as student\_data.txt, tweets\_data.txt, research\_data.txt. Because they are totally different formatted data so per-processing techniques should be varied. It had done by nltk.tokenize package.

**Tokenization**

Accuracy was checked by difference between before and after tokenization.

Defect percentage =

(|beforeTokenizedWordCount – afterTokenizedWordCount|) \* 100%/ beforeTokenizedWordCount

if it is 0 then accuracy is high, when it increases implies that the low accuracy occurred form the used tokenization method

for student\_data.txt

|  |  |
| --- | --- |
| Tokenizer used | word\_tokenize |
| Before tokenize word count is | 401 |
| after tokenize word count is | 434 |
| Defect percentage | =(|434 - 401 |)/401 \*100  =8.23% |

Therefore suitable tokenizer is word\_tokenizer for student data set.

for tweeter\_data.txt

TweetTokenizer is designed specially for tweets tokenization purpose, it is more accurate than word tokenization.

|  |  |
| --- | --- |
| Tokenizer used | TweetTokenizer |
| Before tokenize word count is | 435 |
| after tokenize word count is | 514 |
| Defect percentage | =(|435 - 514 |)/ 435 \*100  =18.16% |

Tweets tokenizer works well for tweets instead of word tokenizer

for research\_data.txt

|  |  |
| --- | --- |
| Tokenizer used | word\_tokenize |
| Before tokenize word count is | 414 |
| after tokenize word count is | 470 |
| Defect percentage | =(|414 - 470|)/ 414 \*100  =13.53% |

2. Carry out isolated word correction and context sensitive word correction on the text. Based on your observations on the processed text, discuss the impact of each type of spell corrections on the three types of text.

**Isolated spell corrector**

from spellchecker import SpellChecker

spell = SpellChecker()

misspelled = spell.unknown(tokenized\_list)

used the above spellchecker to correct misspelled words.

Student data

|  |  |
| --- | --- |
| misspelled words are | {'helpfull', 'lectuers', 'exercisers', "'s", 'undersatand'} |

Total are five words appeared in the given text file

tweet data

|  |  |
| --- | --- |
| misspelled words are | {'lpc', '\x88', '...', 'fasttraffic', 'https://t.co/uibsezoqas', 'sitetraffic', 'onthisday', 'https://t.co/xxdeig7dbu', "d'immigration", 'míème', '@ladymadonna\_\_\_', 'cpcldräó', '@canadidly', 'https://t.co/becgusy2i6', 'https://t.co/0c5obfmxlg', 'https://t.co/5lievho7a4', 'abhinav', 'https://t.co/lsg7c3vle9', 'vaste', 'irrí', 'ays', 'globalist', 'https://t.co/vcmfygadr5', 'äóìsubstantially', 'https://t.co/yxh5w53sro', 'contríîle', 'monsef', 'https://t.co/4k84ee8y63', 'accíâs', 'https://t.co/cneywn40x3', "family's", 'https://t.co/j77devjoix', 'https://t.co/kwz3csvyxm', 'â\_ô', 'æquestions', 'https://t.co/ihpvhw2bag', '\x81', '@theeconomist', 'willl', 'https://t.co/ec3xhoro2s', "canada's", 'äó\_', "it'säó", 'https://t.co/rqrr5nebcg', 'https://t.co/x2ifo0exi2', 'íæ', 'campagne', 'haryanavi', 'jat', '262m', 'ñuvre', 'website', 'https://t.co/yagwmz8ecp', 'https://t.co/zozose1cqq', '@shawhelp', 'greencard', "i've", 'https://t.co/m5ckgyvv8f', '=p', 'cdnpoli', '\x8c', 'https://t.co/cyi867pzrv', 'https://t.co/kxdfmgtzzn', 'https://t.co/uvruw8er1b', '@sweetnessshawnb', 'https://t.co/dkpukywmak', 'fear-mongering', 'besoin', 'https://t.co/99myliuoes', "l'immigration", 'dí', 'https://t.co/wtqk4qdiki', 'https://t.co/s3hu1okkig', 'deyfy', 'guliíâre', 'stäó', '@youtube', 'https://t.co/9i72frhtij', 'https://t.co/zrlj26jnkc', 'läó', 'https://t.co/nefw30mraa', 'æimmigration', '@huffingtonpost', 'rigoureusement', 'https://t.co/f4uto5a7zf', '\_ô', '©', '»', 'cäó', "that's", 'xkofy', 'cortiquí', 'hatecrimeäó'} |

Total 90 words were appeared misspelled in the given text file most of them are url s.

“globalist ,that's , family's , Canada's, fear-mongering”

the above list those words haven’t any spelling mistakes but it takes them as misspelled words

research data

|  |  |
| --- | --- |
| misspelled words are | {'orthogonality', 'shared-private', 'neural-based', '1-', 'luong', 'constraints.specifically', 'task-specific', 'herently', 'task-dependent', 'andweston', 'sharable', '2016c', '\x93infantile\x94', 'task-invariant', 'multi-task', 'collobert'} |

Total are five words appeared in the given text file

“ task-invariant, multi-task, orthogonality, task-specific, task-dependent, neural-based , shared-private “

those above words are actually no any mistakes but it consider as incorrect words

context-sensitive-spelling-correction

3. Stem and lemmatize the text using a suitable stemmer and a lemmatizer. Based on your observations on the processed text, discuss the suitability of stemming and lemmatizing for retrieving base forms of words.

After misspelled word re correction can be apply stemming and lemmatization for three data files

**Stemmer**

from nltk.stem.porter import PorterStemmer

porter\_stemmer = PorterStemmer()

porter\_stemmer.stem(correct\_dataFile)

**Lemmatizer**

from nltk.stem import WordNetLemmatizer

wordnet\_lemmatizer = WordNetLemmatizer()

wordnet\_lemmatizer.lemmatize(correct\_dataFile)

**References**

<https://pythonspot.com/tokenizing-words-and-sentences-with-nltk/>

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

<https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908>

<https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk>

<https://rustyonrampage.github.io/text-mining/2017/11/28/spelling-correction-with-python-and-nltk.html>

https://www.nltk.org/api/nltk.tokenize.html

<https://media.readthedocs.org/pdf/pyspellchecker/latest/pyspellchecker.pdf>

<https://www.tutorialspoint.com/python/python_spelling_check.htm>

<https://www.youtube.com/watch?v=2SjQBRagpkg>

<https://pypi.org/project/pyspellchecker/>

<https://nlp.stanford.edu/IR-book/html/htmledition/context-sensitive-spelling-correction-1.html>

<https://norvig.com/spell-correct.html?fbclid=IwAR2-SyWinhXws01i8FldkXImPw6gBhQGwdVg8IDUbVltMWEAwhUvdJE3K5M>

<https://norvig.com/spell-correct.html>

<https://dzone.com/articles/nlp-tutorial-using-python-nltk-simple-examples>