# CS 5615: Information Retrieval Assignment 1

## Tokenizing

For tokenizing each data set, Natural Language Toolkit's (NLTK) word_tokenize function is used. When tokenizing the words using this word_tokenize() function following issues were identified.

1. **HTML tags were erroneously tokenized.**
   Single URL was divided into several parts and tokenized into separate tokens.
   Ex -: HTML tag *<br/>* in the text, is tokenized into separate four tokens as *'<', 'br', '/', '>'*. And this invalid tokenization causes difficulties in spell correction.

2. **URLs were erroneously tokenized.**
   Single URL was divided into several parts and tokenized into separate tokens.
   Ex - URL https://t.co/ZOZOSe1CqQ was tokenized into separate tokens as *'https', ':', '//t.co/ZOZOSe1CqQ'*

3. **Hashtags were erroneously tokenized.**
   Hashtags were divided into separate tokens by selecting *'#'* symbol as one token and other words as separate tokens.
   Ex - hashtag '#immigration' was tokenized into separate tokens as *'#', 'immigration'*

4. **Emoji's were erroneously tokenized.**
   Emoji's were divided into separate tokens by each symbol.
   Ex - Smiley emoji :) is divided into two token as *':', ')'*

## Spell Correction

The 'SclstmChecker' which is given by neuspell was used for context sensitive word correction. And the '`edit_distance`' function which is given by the nltk.metrics.distance was used for the isolated word correction.

The 'SclstmChecker' spell checker could be used on three datasets without doing any preprocessing to the contents in each dataset. But 'edit_distance' couldn't be used on the dataset without doing preprocessing. Because it gave an error when a numeric value or symbol is given as input to the edit_distance'() function. So the 'edit_distance' function was used in the Student Course Feedback dataset after removing html tags and punctuations by preprocessing the content.

Results of spell correction

| **Context sensitive word correction** (‘SclstmChecker’) | | **Isolated word correction** (‘`edit_distance`’) | |
|---|---|---|---|
| **Word before correcting spelling** | **Word after correcting spelling** | **Word before correcting spelling** | **Word after correcting spelling** |
| Honestly | Honestly | Honestly | Hester |
| Lectures | Lectures | Lectures | Leonurus |
| `Lecture` | `Lecture` | `Lecture` | `Lactuca` |
| self-study | self-study | self-study | saleslady |
| `Good :)` | `Good :` | `Good :)` | `God` |
| `madame` | `manmade` | `madame` | madame |
| Presentation | Presentation | Presentation | Predentata |
| #cdnpoli | # cdnpoli | #cdnpoli | gave an error |
| multi-task | multi -- task | multi-task | multiflash |
| off-the-shelf | off -- the -- shelf | off-the-shelf | oystershell |
| 16 | 16 | 16 | gave an error |
| ﷺinfantile﷽ | infantile | ﷺinfantile﷽ | gave an error |

When considering the results given by both types of spell correctors on each datasets, Context sensitive word correction outperformed Isolated word correction. Most of the time, Isolated word correction gave an invalid spell correction. It gave a misspelled word even if it works on the correct word.

## Stemming and Lemmatization

For stemming, Natural Language Toolkit’s (NLTK) `PorterStemmer` was used. As the first step. all words were tokenized using NLTK’s word_tokenize function. Then tokenized words were given to the PorterStemmer.stem() function to get stemmed words.

Lemmatization was done using the Natural Language Toolkit’s (NLTK) WordNetLemmatizer. First all words were tokenized using NLTK’s word_tokenize function. Then tokenized words were given to the wordLemmatizer.lemmatize() function to get lemmatized words.

Stemming process removed the capitalization in most of the words which was not seen in the lemmatization process. And also stemming was affected on most of the words but it gave incorrect words as its root words. Lemmatization did not affect more words and it did not give incorrect words as its root words.

## Results of stemming and lemmatization

| **Stemming** | | **Lemmatization** | |
|---|---|---|---|
| **Word before stemming** | **Word after stemming** | **Word before lemmatization** | **Word after lemmatization** |
| Neural | neural | Neural | Neural |
| promising | promis | promising | promising |
| models | model | models | model |
| opportunities | opportun | opportunities | opportunity |
| spaces | space | spaces | space |
| feature | featur | feature | feature |

When comparing the results of stemming and lemmatization, lemmatization works well in retrieving the base forms of words.

Source code Github URL - https://github.com/KaveeshBaddage/DataScienceImpl/blob/main/Information%20Retrieval/Text%20Analysis/Information_Retrieval_%E2%80%93_Assignment_1.ipynb