**Homework 1: Regex and Normalization**

**Aim:**

Create a dictionary to be used in a simple spell checker program using python.

**Observation Criteria and Analysis:**

Regex:

Dataset used: “War of Worlds.”

1. Packages used: re (regular expression)
2. Replace British English Spellings for words that have an extra ‘u’ with American English spellings:
   1. RE function used: ‘re.sub()’
   2. Approach 1: Using simple regex function where in the words to be replaced were hard coded, but that approach isn’t feasible.
   3. Approach 2: Using a dictionary of words with British spellings as key and American spellings to be the value and manually replacing them using the sub function. This approach wasn’t feasible too as there might be many such key-value pairs and it’s hard to hardcode all of them.
   4. Approach 3: ‘[a-zA-Z]\*?(our)’ This expression captured British words with an extra ‘U’ but also captured words such as your, sour, tour, etc from the dataset.
   5. Approach 4: ‘[^(y|p|s|t|h|f|\s)](our)\b’ This expression captured words with British words with an extra ‘U’ but replaced the last 4-characters of the spelling, i.e.: consider ‘colour’ gets replaced as ‘coor.’ Also, this expression didn’t capture words like your, tour, sour like the previous approach.
   6. Approach 5: ‘([a-zA-Z]+)[^(?:y|p|s|t|h|f|\s)](our)’ This expression captured British words such as ‘colour’, etc but still failed to capture words like ‘coloured’.
   7. Approach 6: ‘\b([a-z]+)our(ed|ing|s|hood)?\b: This approach covered words suchas ‘your’.
   8. Approach 7:

‘\b([a-zA-Z]+)[^(?:y|p|s|t|h|f|\s)](our)(ed|ing|s|hood)\*\b’: This expression address the drawbacks of the previous approach.

1. Replace titles with appropriate expansions of words: This was done using a simple dictionary and ‘re.sub()’ function.

Normalization:

Dataset used: Processed text from previous regex steps.

1. Package used: Regular Expression
2. Convert texts with uppercase to lower case, for standardization using ‘re.findall()’ and ‘replace()’ functions.
3. Removing special characters, numbers and punctions to clean the dataset such that the focus remains on the words using ‘re.sub()’ function.
4. Removing most used and recurring words such as ‘the’, ‘and’, ‘is’.
5. Tokenization (breaking of file into words) using the ‘re.split()’ function.

**Summary:**

1. Regex was successfully used to replace titles with appropriate expansion and covert British spelling to American spellings. How so ever, some British spellings were not captured and converted as desired.
2. Text Normalization for cleaning and filtering the dataset was successful. By observation, all the words were converted into standard format and characters, punctions and numbers were discarded, additionally recurring words were discarded too to improve the quality of the dictionary created.
3. Finally, the text was tokenized into individual words to create a dictionary.