NLP assignment - Initial Screening Exposys Data Labs

# Question Type Identification

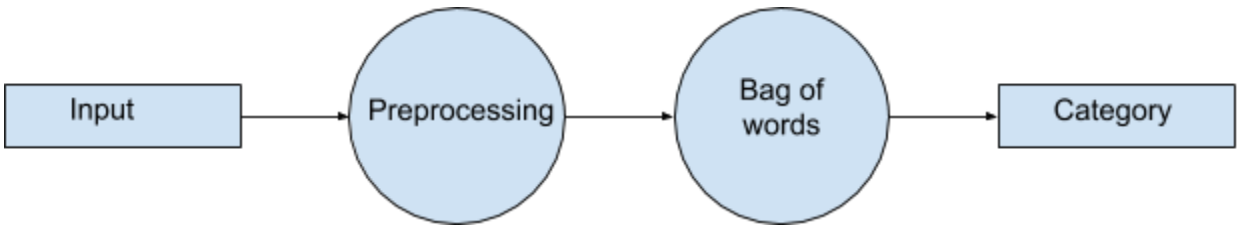
## Abstract

In this assignment, our aim was to identify the question type whether the question is of type and the four categories to handle for this assignment are: Who, What, When, Affirmation(yes/no). If the question does not classify from the above categories then it is labeled as 'unknown'. For this assignment, we have used a simple NLP( Natural Language Processing ) approach to solve this problem.

## Design and architecture

We have a labeled dataset of questions which are labeled as who, when, what, affirmation, and unknown. After looking at the dataset we can see that the text is not clean so the first step is the text cleaning process which is the most important process for text classification in NLP.

```
how did serfdom develop in and then leave russia ? ,,, unknown
what films featured the character popeye doyle ? ,,, what
how can i find a list of celebrities ' real names ? ,,, unknown
what fowl grabs the spotlight after the chinese year of the monkey ? ,,, what
```




### Data Preprocessing

1. **Remove punctuation:**

   One way of doing this is by looping through the Series with list comprehension and keeping everything that is not in string.punctuation, a list of all punctuation we imported at the beginning with import string.

```
12 def removePunctuation(data):
13     for char in data:
14         if char in string.punctuation:
15             data=data.replace(char,' ')
16     return data
```

2. **Remove Apostrophe:**

   In writing, an apostrophe is used to indicate the place of missing letters. So here we have removed this character.

```
18 def removeApostrophe(data):
19     for char in data:
20         if char is "'":
21             data=data.replace(char,'')
22     return data
```

3. **Remove Numbers:**

   Since we are dealing with text, so the number might not add much information to text processing. So, numbers can be removed from the text. We can use regular expressions (*regex*) to get rid of numbers or isdigit() also can be used.

```
49 def removedigits(data):
50     words = nltk.word_tokenize(data)
51     newText=""
52     for word in words:
53         if word.isdigit() == False:
54             newText=newText+" "+word
55     return newText
```

4. **Remove stop words:**

We imported a list of the most frequently used words from the NL Toolkit at the beginning with from nltk.corpus import stopwords. You can run stopwords.word(insert language) to get a full list for every language. There are 179 English words, including 'i', 'me', 'my', 'myself', 'we', 'you', 'he', 'his', for example. We usually want to remove these because they have low predictive power. There are occasions when you may want to keep them though. Such as, if your corpus is very small and removing stop words would decrease the total number of words by a large percent.

```
32 def removeStopwords(data):
33     affir=['am', 'is', 'are', 'was','were','would','can','could','should','d
34     words = nltk.word_tokenize(data)
35     stopwordslist= stopwords.words('english')
36     stopwordslist.remove('what')
37     stopwordslist.remove('when')
38     newText=""
39     for i in range(len(words)):
40         if words[0] in affir and i==0:
41             if words[1] in stopwordslist:
42                 newText+=" "+words[i]+" "+words[1]
43         if words[i] not in stopwordslist:
44             newText=newText+" "+words[i]
45 #     if words[-1] not in stopwordslist:
46 #         newText=newText+" "+words[-1]
47     return newText
```

5. **Stemming:**

Stemming is the process of reducing inflected/derived words to their word stem, base, or root form. The stem need not be identical to the original word. There are many ways to perform stemming such as lookup table, suffix-stripping algorithms, etc. These mainly rely on chopping-off *'s', 'es', 'ed', 'ing', 'ly'* etc from the end of the words, and sometimes the conversion is not desirable. But nonetheless, stemming helps us in standardizing text.

```
57 def stemming(data):
58     stemmer=PorterStemmer()
59     words = nltk.word_tokenize(data)
60     newText=""
61     for word in words:
62         word=stemmer.stem(word)
63         newText=newText+" "+word
64     return newText
```

After completion of data preprocessing we can use the data for further analysis. Now we will create a dictionary of words and their frequency in the questions which will work as our features of the dataset.

## Bag of Words

The **bag-of-words** model is a simplifying representation used in natural language processing and information retrieval (IR). A bag-of-words model, or BoW for short, is a way of extracting features from the text for use in modeling. A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things:

1. A vocabulary of known words.
2. A measure of the presence of known words.

So here we will normalize the frequency of each word by dividing the number of words by the sum of the 20 most common word frequencies. **Let's say if in the bag of words frequency of 'what' is 100 and the sum of the 20 most common word frequencies is 500 then the frequency of the word 'what' is normalized as 100/500 which is equal to 0.20.** We will do the same for all the categories we want to classify.

## Classification

Since we have a bag of words with normalized frequencies, we can classify the questions. For classification, we have to perform data preprocessing on that question which we have done earlier. After that, we will match the respective word from the question and measure the score so that we can get to know that this question falls in which category.

## Conclusion

After performing all the operations we have tested it for the same dataset and we are getting an accuracy of 79.63% which is quite good. We can improve it further by using more methods used for text classification.

## Steps to install and run the code

1. Install pip3 using **sudo apt-get install python3-pip** in the terminal.
2. Install pandas library using **pip install pandas** in the terminal.
3. Install nltk library using **pip install nltk**
4. In the python IDE run,
    a. **import ntlk**
    b. **nltk.download('punkt')**
    c. **nltk.downoad('stopwords')**
5. Unzip the zip file in your system
6. Open the terminal and change to the directory which contains the file 'assignment.py'.
7. Run **'python3 assignment.py'** in the terminal.