



Arabic Sentiment Analysis

Submit by :

Atheer Alharbi (2191003390)

Amjad Almutairi (2171003385)

Abrar Alrashidi (2191003368)

Hanan Aljuhani (2191003341)

Date : 22/5/2022

Supervisor : Dr. Aminat Ajibola

Bag-of-Words Representation of Arabic Tweets Dataset

1. Arabic Sentiment Analysis Dataset (ASAD)

This section describes the dataset that we use in the project.

1. When was it developed? By whom?

Arabic Sentiment Analysis Dataset (ASAD) is a twitter-based benchmark dataset for Arabic Sentiment Analysis. It was launched in competition under the sponsorship of KAUST. The developing team has many researchers involved in the making of this dataset. The names are Basma Alharbi, Hind Alamro, Manal Alshehri, Zuhair Khayyat, Manal Kalkatawi, Inji Ibrahim Jaber, and Xiangliang Zhang. Approximately 100K tweets were gathered in the dataset between 2012 to 2020. [1] [2]

2. How many classes in the dataset? What are they?

As per the given stats, the dataset has been divided into three main classes which are Positive, Negative and Neutral. [1]

3. Into how many sets was the dataset divided?

The Dataset is principally divided into three sets. One is training set containing 80% of the tweets and other two are test sets I and II. [1]

4. Useful statistics (samples in each class? Samples in each set)

ASAD has a total of 15,282 positive tweets, 15,349 negative tweets, and 69,369 neutral tweets. The tweets were collected in the period between May 2012 and April 2020. They are written in different Arabic dialects including Khaleeji, Hijazi, Egyptian, and Modern Standard Arabic. [1] [2]

	TRAINING		TEST 1		TEST 2	
	No. Tweets	(%)	No. Tweets	(%)	No. Tweets	(%)
POSITIVE	8821	16.0	3150	16.0	3244	16.0
NEGATIVE	8820	16.0	3252	16.0	3195	16.0
NEUTRAL	37359	68.0	13598	68.0	13561	68.0
TOTAL	55000	100.0	20000	100.0	20000	100.0

The Dataset used in the Project

In this project, 3000 tweets were selected randomly from the training data of ASAD, 1000 samples from each class. The selected tweets will be divided evenly into 80% for training and 20% for testing.

Show samples tweets for each class

Positive: إن كان لك نصيب في شيء ، سيقلب الله كل الموازين لكي تحصل عليه

Negative: " ما يراد و يخطط لم يحدث حتى في فترات الاحتلال لمصر .نحن العن علي أنفسنا من الاحتلال

Neutral: "لا يوجد رقم شكوى فقط تم ارسال هذا الايميل @EtihadHelp"

2. Bag-of-Words Representation

This Section provides brief introduction to BoW Representation:

1. describe the main steps in BoW representation

Main steps involve in BoW representation are:

- Data Collection.
 - Dictionary or Vocabulary Creation
 - Document Vectorization. [3]
2. study the code in the function `clean_tweet()` in the file “2_Build_Dictionary.py” and describe the main preprocessing steps that we implemented.

When we read and analyze the function `clean_tweet()` and look at the program flow, we can easily understand that following preprocessing steps are being implemented by this functions, i-e,

- Remove Non-Arabic Words.
- Removing White Spaces and Punctuations.
- Removal of Stop Words from the text.
- After that, remaining words are converted into tokens. [3]

All of this is shown in the figure below, i-e, [3]

```
# turn a tweet into clean tokens
def clean_tweet(tw):
    #remove non-Arabic text
    tw = removeNonArabicLetters(tw)
    # split into tokens by white space
    tokens = tw.split()
    # remove punctuation from each token
    table = str.maketrans('', '', punctuation)
    tokens = [w.translate(table) for w in tokens]
    # filter out stop words
    stop_words = set(stopwords.words('arabic'))
    tokens = [w for w in tokens if not w in stop_words]
    # filter out short tokens
    tokens = [word for word in tokens if len(word) > 2]
    return tokens
```

3. Producing Bag-of-words Representation:

In this project, we utilize the Tokenize class of Keras API to automatically produce the BoW of our dataset based on the vocabularies (dictionary) that we learned it from the dataset.

The Tokenizer is created, fit on the dataset, and use to produce BoW to each tweet is our data set:

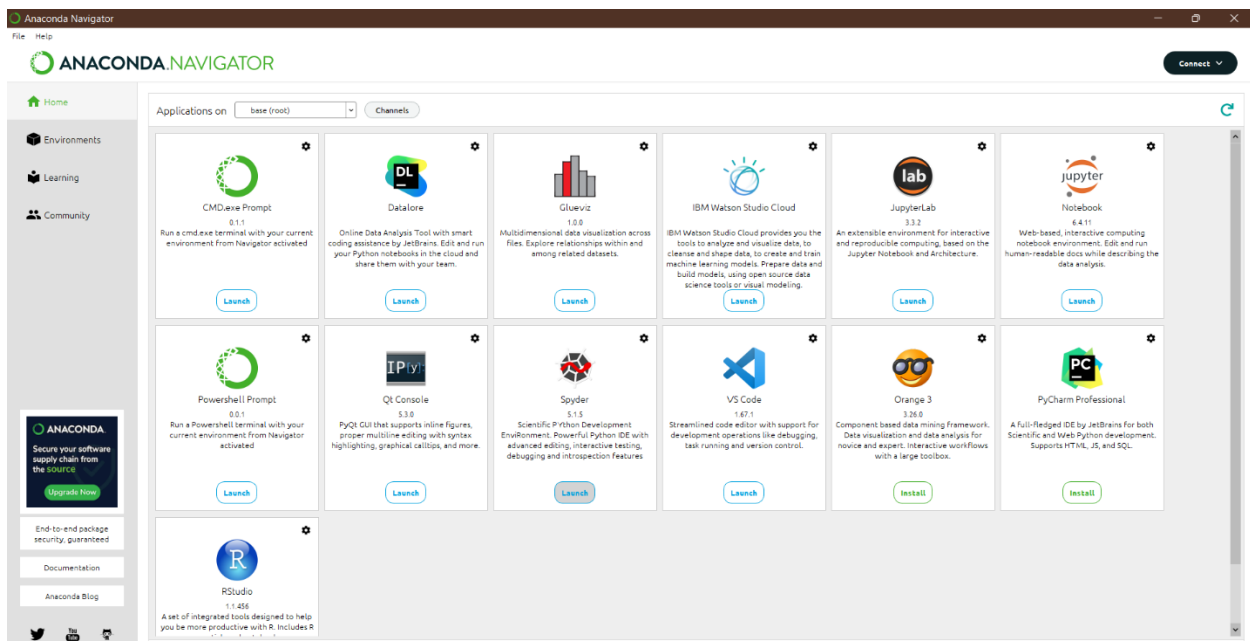
```
# create the tokenizer
tokenizer = Tokenizer()

# fit the tokenizer on all tweets
all_tweets = pos_tweets_lines + neg_tweets_lines + neu_tweets_lines
tokenizer.fit_on_texts(all_tweets)

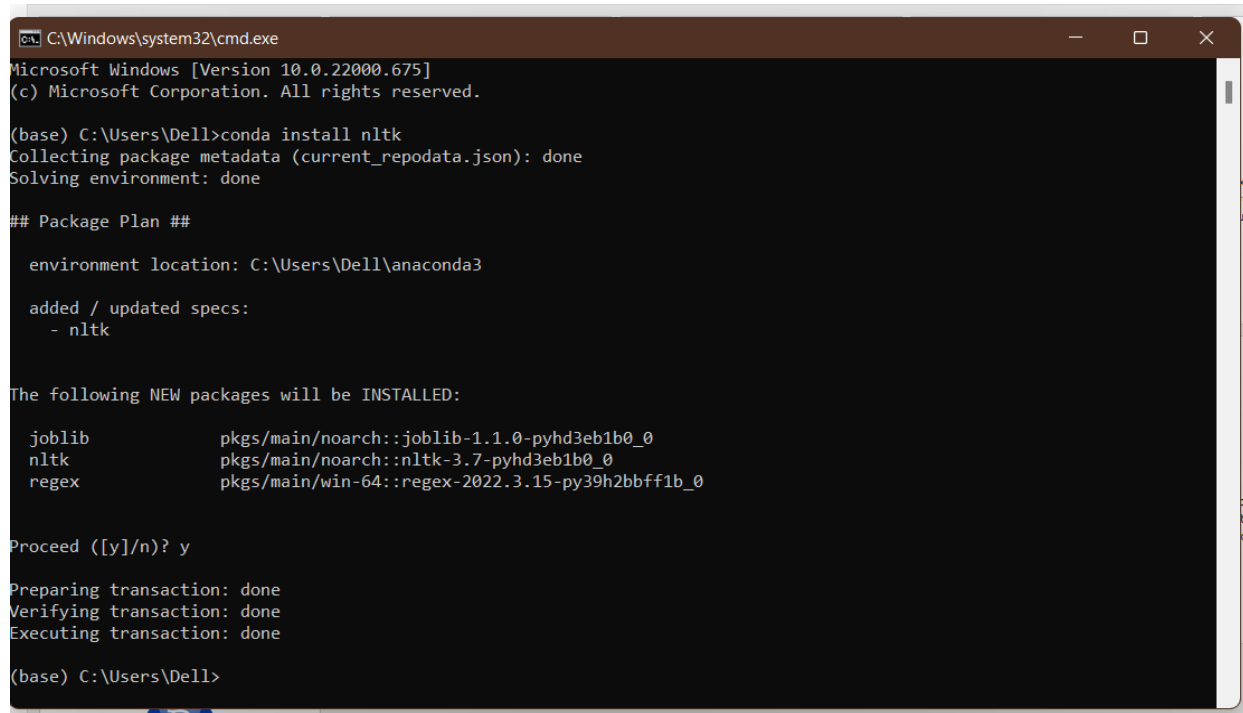
# encode tweets
bow = tokenizer.texts_to_matrix(all_tweets, mode='freq')
print(bow.shape)
```

Now, we will try to showcase the working of the python codes for extracting the useful information for Bag of Words representation.

Firstly, we shall start the anaconda program by opening anaconda navigator. This consists of different programs which can be used as per the requirements of the user. We need Spyder for our operations. The GUI of anaconda navigator and icon of Spyder IDE is shown below, i-e,



After launching Spyder, we open “1_prepare_environment.py”, in that file we see that we have to install nltk and keras packages in order for our code to run properly. We open the Anaconda command prompt and install the packages by using command “conda install packages name”. This operation is shown in the figures below, i-e,



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

(base) C:\Users\Dell>conda install nltk
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\Dell\anaconda3

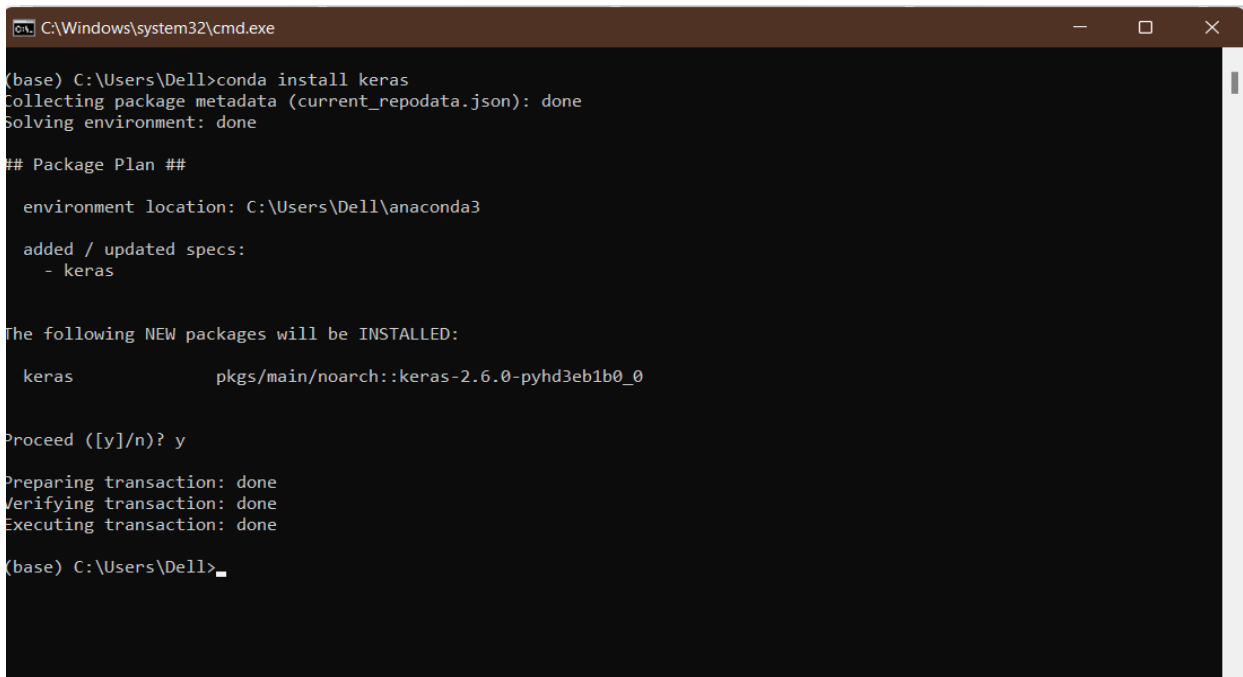
  added / updated specs:
    - nltk

The following NEW packages will be INSTALLED:

  joblib           pkgs/main/noarch::joblib-1.1.0-pyhd3eb1b0_0
  nltk             pkgs/main/noarch::nltk-3.7-pyhd3eb1b0_0
  regex           pkgs/main/win-64::regex-2022.3.15-py39h2bfff1b_0

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) C:\Users\Dell>
```



```
C:\Windows\system32\cmd.exe

(base) C:\Users\Dell>conda install keras
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\Dell\anaconda3

  added / updated specs:
    - keras

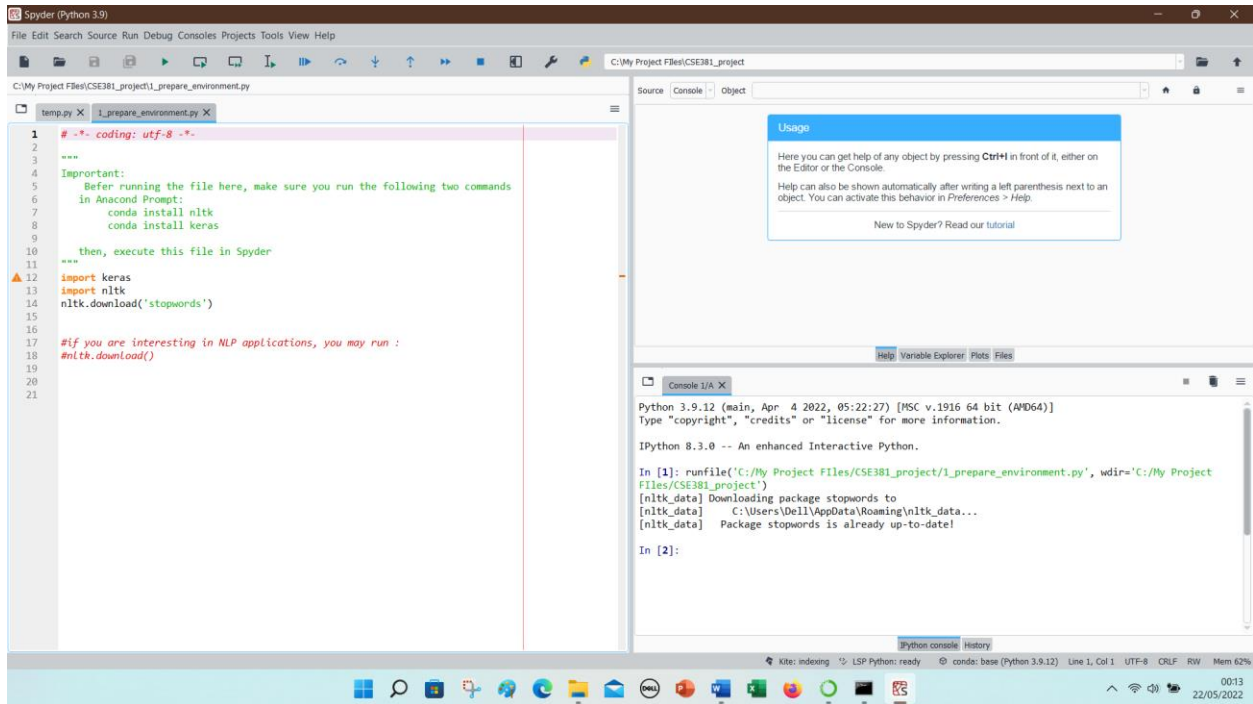
The following NEW packages will be INSTALLED:

  keras           pkgs/main/noarch::keras-2.6.0-pyhd3eb1b0_0

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) C:\Users\Dell>
```

Then, we run the first python program to get nltk stopwords for our dataset.



The screenshot shows the Spyder Python IDE interface. The editor window displays a script named `1_prepare_environment.py` with the following content:

```
1 # -*- coding: utf-8 -*-
2
3
4 """
5 Important:
6 Refer running the file here, make sure you run the following two commands
7 in Anacond Prompt:
8 conda install nltk
9 conda install keras
10
11 then, execute this file in Spyder
12 """
13 import keras
14 import nltk
15 nltk.download('stopwords')
16
17 #if you are interesting in NLP applications, you may run :
18 #nltk.download()
19
20
21
```

The console window shows the output of the script:

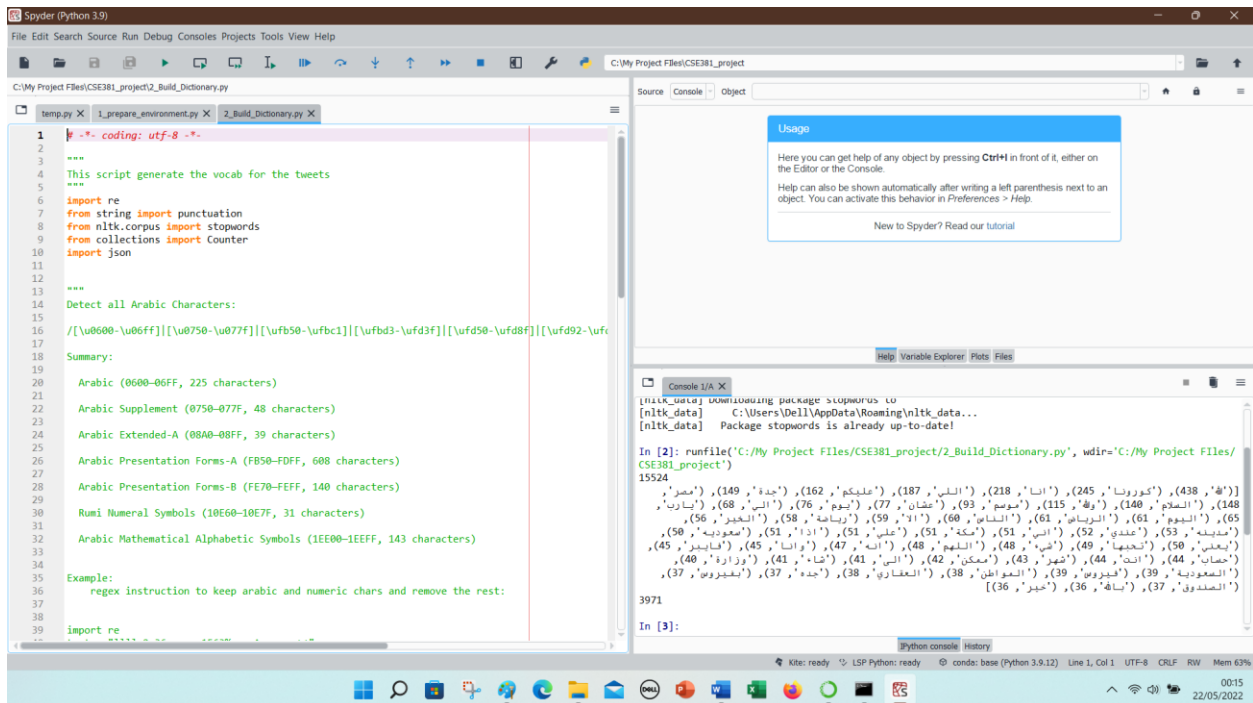
```
Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 8.3.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/My Project Files/CSE381_project/1_prepare_environment.py', wdir='C:/My Project
Files/CSE381_project')
[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\Dell\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!

In [2]:
```

After that we run “2_Build_Dictionary.py”. After successful execution of this file, we get a text file in our project folder with the name Vocab.txt










The screenshot shows the Spyder Python IDE interface. The editor window displays a script named `2_Build_Dictionary.py` with the following content:

```
1 # -*- coding: utf-8 -*-
2
3 """
4 This script generate the vocab for the tweets
5 """
6 import re
7 from string import punctuation
8 from nltk.corpus import stopwords
9 from collections import Counter
10 import json
11
12
13
14
15
16 Detect all Arabic Characters:
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

The console window shows the output of the script:

```
In [2]: runfile('C:/My Project Files/CSE381_project/2_Build_Dictionary.py', wdir='C:/My Project Files/
CSE381_project')
15524
[('أ', 438), ('كورونا', 245), ('إس', 218), ('التي', 187), ('عليكم', 162), ('جدة', 149), ('مصر', 148), ('المدى', 140), ('و', 115), ('موسم', 93), ('عشان', 77), ('يوم', 76), ('التي', 68), ('بارب', 65), ('اليوم', 61), ('الرياض', 61), ('النايف', 60), ('أ', 59), ('رياسة', 58), ('الجور', 56), ('مدينة', 53), ('عندى', 52), ('السي', 51), ('مكة', 51), ('علي', 51), ('أ', 51), ('معدية', 50), ('يغنى', 50), ('تجديا', 49), ('في', 48), ('الليهم', 48), ('أ', 47), ('والا', 45), ('فانيفر', 45), ('جمان', 44), ('أنا', 44), ('فقر', 43), ('ممكن', 42), ('التي', 41), ('أنا', 41), ('وزارة', 40), ('المدى', 39), ('فيديو', 39), ('المواطن', 38), ('العقاري', 38), ('جدة', 37), ('ليفرزون', 37), ('المدى', 37), ('أ', 36), ('خير', 36), ('أ', 36)]
3971

In [3]:
```

Name	Date modified	Type	Size
 1_prepare_environment.py	17/05/2022 19:48	Python Source File	1 KB
 2_Build_Dictionary.py	11/04/2021 12:37	Python Source File	4 KB
 3_Build_BoW_representation.py	11/04/2021 13:35	Python Source File	3 KB
 neg_tweets.json	11/04/2021 09:13	JSON Source File	175 KB
 neu_tweets.json	11/04/2021 09:13	JSON Source File	173 KB
 pos_tweets.json	11/04/2021 09:13	JSON Source File	170 KB
 vocab.txt	21/05/2022 22:41	Text Document	46 KB

After that, we run “3_Build_BoW_representation.py” to check and print the size of Bag of Words Representation printed by the tokenizer.

The screenshot shows the Spyder Python IDE interface. The main editor window displays a Python script named `3_Build_BoW_representation.py`. The script performs the following steps:

- Imports `re`, `json`, `string`, `punctuation`, `nlTK.corpus`, `stopwords`, and `keras.preprocessing.text`.
- Defines a `load_doc` function to load a document from a file.
- Defines a `removeNonArabicLetters` function to remove non-Arabic characters from a string.
- Defines a `clean_tweet` function to clean a tweet by removing non-Arabic text, splitting into tokens, removing punctuation, and filtering out stopwords and short tokens.
- Executes the `clean_tweet` function on a sample tweet.
- Builds a BoW matrix using `CountVec` from `keras.preprocessing.text`.
- Prints the size of the BoW representation.

The console output shows the execution of the script, including the size of the BoW representation: `The size of BoW representation = (3000, 3972)`. A 'Usage' dialog box is open, providing information about the `Ctrl+H` shortcut for finding symbols in the code.

```
In [3]: runfile('C:/My Project Files/CSE381_project/3_Build_Bow_representation.py', wdir='C:/My
Project Files/CSE381_project')
The size of Bow representation = (3000, 3972)
```

References

- [1] KAUST, "<https://wti.kaust.edu.sa>," King Abdullah University of Science and Technology (KAUST), [Online]. Available: <https://wti.kaust.edu.sa/SoD/Arabic-Sentiment-Analysis-Challenge>. [Accessed 07 May 2022].
- [2] H. A. M. A. Z. K. M. K. I. I. J. X. Z. Basma Alharbi, "ASAD: A Twitter-based Benchmark Arabic Sentiment Analysis Dataset," arxiv.org, 2020.
- [3] J. Brownlee, "<https://machinelearningmastery.com>," 03 September 2020. [Online]. Available: <https://machinelearningmastery.com/deep-learning-bag-of-words-model-sentiment-analysis/>. [Accessed 08 May 2022].