DTI5125: Data Science Applications

Text Summarization

Term Project

Group Members:

Basma Reda Shaban Abd-Elsalam Abd-Elwahab

Amir Safwat Halim Youssef

Nada Ashraf Ismail AboElfetoh

Yasmine Ahmed Elsayed Mohamed

1- Problem formulation:

Reading long articles is tedious especially if only a single bit of information is needed from an article that's why automatic text summarization is needed to save time.
Automatic text summarization is a Natural Language Processing (NLP) problem that aims at producing a summary of a long article that contains the most important information in this article.
Extractive text summarization does not use words aside from the ones already in the text, and selects some combination of the existing words most relevant to the meaning of the source. we chose various articles with different categories after that we made preprocessing, we applied Summarization Models like Bert and LSA .

1- Implementation steps:

1- Upload the dataset and explore the data:

```
[3] #read the dataset
    df = pd.read_csv("/content/dataset.csv")
```

```
#print head of data
df.head(5)
```

	label	text	Text_summary
0	business	The Federal Reserve approved Ally Financial In...	The Federal Reserve approved Ally Financial In...
1	business	— Major shareholders of Duke Energy Corp. have...	— Major shareholders of Duke Energy Corp. have...
2	business	Photos taken earlier this month show that Nort...	Photos taken earlier this month show that Nort...
3	business	Thanks to dogged reporting by the Associated P...	Thanks to dogged reporting by the Associated P...
4	business	The energy giant says it is committed to clean...	The energy giant says it is committed to clean...

So, lets describe the data and get some information about the data:

	label	text	Text_summary
count	36891	36891	36891
unique	4	33288	33169
top	health	Why did this happen?\n\nPlease make sure your ...	Why did this happen?\nPlease make sure your br...
freq	10505	350	350

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36891 entries, 0 to 36890
Data columns (total 3 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   label         36891 non-null  object
 1   text          36891 non-null  object
 2   Text_summary  36891 non-null  object
dtypes: object(3)
memory usage: 864.8+ KB
```

2- Data preparation and data pre-processing:

Make a unique ID for each label:

```
df['labelID'] = df['label'].factorize()[0]
df
```

	label	text	Text_summary	labelID
0	business	The Federal Reserve approved Ally Financial In...	The Federal Reserve approved Ally Financial In...	0
1	business	— Major shareholders of Duke Energy Corp. have...	— Major shareholders of Duke Energy Corp. have...	0
2	business	Photos taken earlier this month show that Nort...	Photos taken earlier this month show that Nort...	0
3	business	Thanks to dogged reporting by the Associated P...	Thanks to dogged reporting by the Associated P...	0
4	business	The energy giant says it is committed to clean...	The energy giant says it is committed to clean...	0
...	...	...	...	...
36886	technology	Why did this happen?\n\nPlease make sure your ...	Why did this happen?\nPlease make sure your br...	3
36887	technology	Google Inc (NASDAQ:GOOGL) GOOGL +1.12% (NASDAQ...	Google Inc (NASDAQ:GOOGL) GOOGL +1.12% (NASDAQ...	3
36888	technology	Google has purchased New Mexico-based unmanned...	Google has purchased New Mexico-based unmanned...	3
36889	technology	hidden\n\nLooks like Facebook's plans to get I...	Google has beaten the world's largest social n...	3
36890	technology	Google Has Plans For Titan Drones\n\nTitan Aer...	Google Has Plans For Titan DronesTitan Aerospa...	3

36891 rows × 4 columns

Mapping treebank to wordnet lemmatized:

```
part = { # mapping treebank to wordnet lemmatizer
    'N' : 'n',
    'V' : 'v',
    'J' : 'a',
    'S' : 's',
    'R' : 'r'
}
```

```
def get_tag(tag): #  used for lemmatizing
  if tag[0] in part.keys():
    return part[tag[0]]
  else:
    return 'n'
```

Let's start to clean our data:

1- Remove stop words.
2- Perform stemming.
3- Perform lemmatization.
4- Perform tokenization.

Data after cleaning:

	label	text	Text_summary	labelID	cleaned_text
0	business	The Federal Reserve approved Ally Financial In...	The Federal Reserve approved Ally Financial In...	0	federal reserve approve ally financial inc.'s ...
1	business	— Major shareholders of Duke Energy Corp. have...	— Major shareholders of Duke Energy Corp. have...	0	major shareholder duke energy corp. call compa...
2	business	Photos taken earlier this month show that Nort...	Photos taken earlier this month show that Nort...	0	photo take earlier month show north carolina r...
3	business	Thanks to dogged reporting by the Associated P...	Thanks to dogged reporting by the Associated P...	0	thanks dog report associate press, know active...
4	business	The energy giant says it is committed to clean...	The energy giant says it is committed to clean...	0	energy giant say committed clean dan river spi...

tokenized	summary_len	text_len	cleaned_Text_summary
[federal, reserve, approve, ally, financial, i...	382	383	federal reserve approve ally financial inc.'s ...
[major, shareholder, duke, energy, corp., call...	1037	2796	major shareholder duke energy corp. call compa...
[photo, take, earlier, month, show, north, car...	799	3563	photo take earlier month show north carolina r...
[thanks, dog, report, associate, press, ,, kno...	681	3269	thanks dog report associate press, know active...
[energy, giant, say, committed, clean, dan, ri...	613	1392	energy giant say committed clean dan river spi...

Now, we can remove rows where text is shorter than summary:

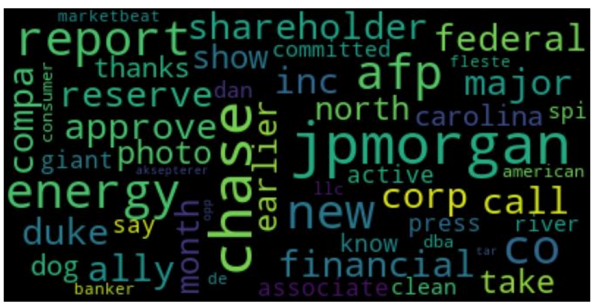
	label	text	Text_summary	labelID	cleaned_text	tokenized
0	business	The Federal Reserve approved Ally Financial In...	The Federal Reserve approved Ally Financial In...	0	federal reserve approve ally financial inc.'s ...	[federal, reserve, approve, ally, financial, i...
1	business	— Major shareholders of Duke Energy Corp. have...	— Major shareholders of Duke Energy Corp. have...	0	major shareholder duke energy corp. call compa...	[major, shareholder, duke, energy, corp., call...
2	business	Photos taken earlier this month show that Nort...	Photos taken earlier this month show that Nort...	0	photo take earlier month show north carolina r...	[photo, take, earlier, month, show, north, car...
3	business	Thanks to dogged reporting by the Associated P...	Thanks to dogged reporting by the Associated P...	0	thanks dog report associate press, know active...	[thanks, dog, report, associate, press, ,, kno...
4	business	The energy giant says it is committed to clean...	The energy giant says it is committed to clean...	0	energy giant say committed clean dan river spi...	[energy, giant, say, committed, clean, dan, ri...

summary_len	text_len	cleaned_Text_summary	sentences
382	383	federal reserve approve ally financial inc.'s ...	[federal reserve approve ally financial inc.'s...
1037	2796	major shareholder duke energy corp. call compa...	[major shareholder duke energy corp. call comp...
799	3563	photo take earlier month show north carolina r...	[photo take earlier month show north carolina ...
681	3269	thanks dog report associate press, know active...	[thanks dog report associate press, know acti...
613	1392	energy giant say committed clean dan river spi...	[energy giant say committed clean dan river sp...

Show the wordCloud of the data:

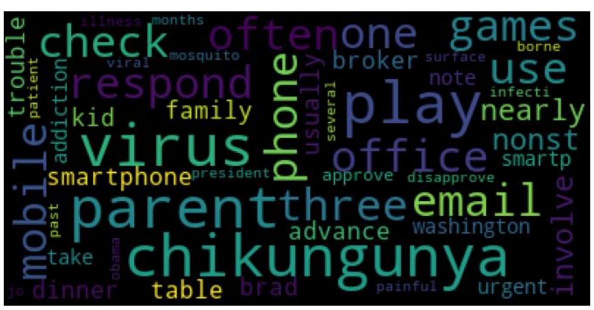
The WordCloud is to show the most frequent words in each label.

Label 1: business

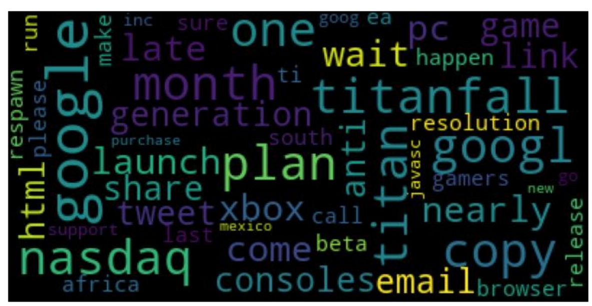


Label 2: entertainment

Label 3: health



Label 4: technology



Split the dataset into train and test split:

```
[26] #split dataset into subsets that minimize the potential for bias in your evaluation and validation process.
     from sklearn.model_selection import train_test_split
     x= np.array(cleaned_df['cleaned_text'])
     y=np.array(cleaned_df['labelID'])
     x_train, x_test, y_train, y_test = train_test_split(x, y, test_size= 0.3, random_state= 0)
```

3- Perform Text Feature engineering:

1- Bag Of Words (BOW):

A bag of words is a representation of text that describes the occurrence of words within a document.

```
[28] vectorizer = CountVectorizer(min_df=2)
     BOWtraining= vectorizer.fit_transform(x_train)
     BOWtesting = vectorizer.transform(x_test)
```

2- TF-IDF:

Term frequency (TF) vectors show how important words are to documents.

```
from sklearn.feature_extraction.text import TfidfVectorizer
TF_IDF = TfidfVectorizer(min_df=6,norm='l2',smooth_idf=True,use_idf=True)
TFIDF_Train = TF_IDF.fit_transform(x_train)
TFIDF_Test = TF_IDF.transform(x_test)
```

4- Apply 3 Classification models:

1- SVM Classification Model.
2- Decision Tree Classification Model.
3- KNN Classification Model.

SVM with BOW Training classification report

	precision	recall	f1-score	support
0	0.96	0.98	0.97	5002
1	0.99	0.98	0.99	5771
2	0.98	0.98	0.98	6939
3	0.98	0.97	0.98	6914
accuracy			0.98	24626
macro avg	0.98	0.98	0.98	24626
weighted avg	0.98	0.98	0.98	24626

SVM with TFIDF Training classification report

	precision	recall	f1-score	support
0	0.95	0.97	0.96	5002
1	0.98	0.98	0.98	5771
2	0.97	0.97	0.97	6939
3	0.98	0.96	0.97	6914
accuracy			0.97	24626
macro avg	0.97	0.97	0.97	24626
weighted avg	0.97	0.97	0.97	24626

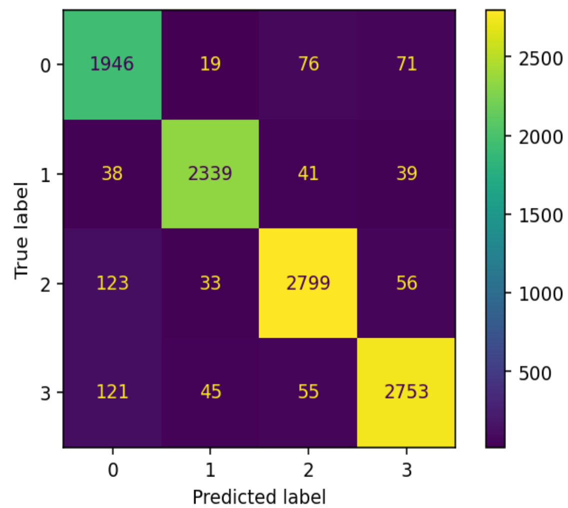
SVM with BOW Testing classification report

	precision	recall	f1-score	support
0	0.87	0.92	0.90	2112
1	0.96	0.95	0.96	2457
2	0.94	0.93	0.94	3011
3	0.94	0.93	0.93	2974
accuracy			0.93	10554
macro avg	0.93	0.93	0.93	10554
weighted avg	0.93	0.93	0.93	10554

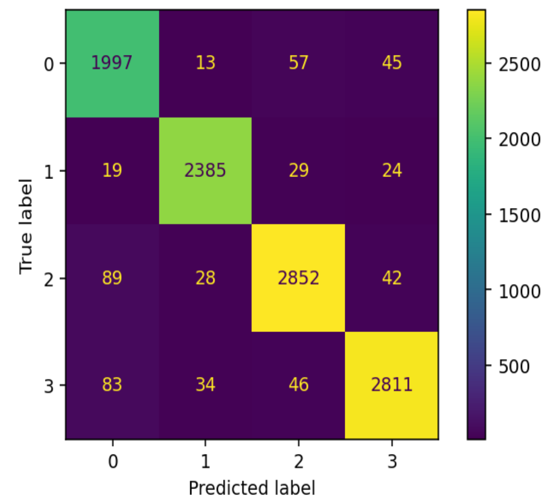
SVM with TFIDF Testing classification report

	precision	recall	f1-score	support
0	0.91	0.95	0.93	2112
1	0.97	0.97	0.97	2457
2	0.96	0.95	0.95	3011
3	0.96	0.95	0.95	2974
accuracy			0.95	10554
macro avg	0.95	0.95	0.95	10554
weighted avg	0.95	0.95	0.95	10554

SVM with BOW confusion matrix



SVM with TFIDF confusion matrix



In training phase, the accuracy of SVM model with BOW is 98%, and accuracy of SVM with TFIDF is 97%
In testing phase, the accuracy of SVM model with BOW is 93%, and accuracy of SVM with TFIDF is 95%

DT with BOW Training classification report

	precision	recall	f1-score	support
0	0.96	0.98	0.97	5002
1	0.98	0.99	0.98	5771
2	0.98	0.98	0.98	6939
3	0.99	0.97	0.98	6914
accuracy			0.98	24626
macro avg	0.98	0.98	0.98	24626
weighted avg	0.98	0.98	0.98	24626

DT with TFIDF Training classification report

	precision	recall	f1-score	support
0	0.96	0.98	0.97	5002
1	0.98	0.99	0.98	5771
2	0.98	0.98	0.98	6939
3	0.99	0.97	0.98	6914
accuracy			0.98	24626
macro avg	0.98	0.98	0.98	24626
weighted avg	0.98	0.98	0.98	24626

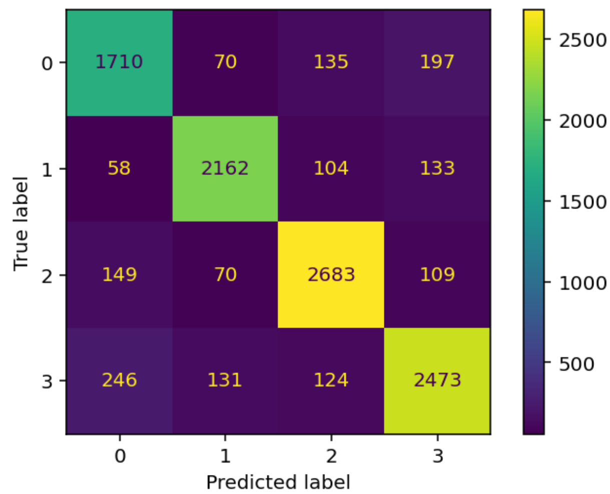
DT with BOW Testing classification report

	precision	recall	f1-score	support
0	0.79	0.81	0.80	2112
1	0.89	0.88	0.88	2457
2	0.88	0.89	0.89	3011
3	0.85	0.83	0.84	2974
accuracy			0.86	10554
macro avg	0.85	0.85	0.85	10554
weighted avg	0.86	0.86	0.86	10554

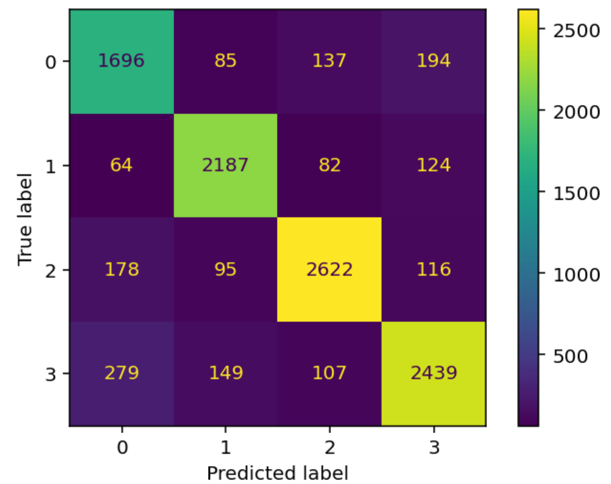
DT with TFIDF Testing classification report

	precision	recall	f1-score	support
0	0.76	0.80	0.78	2112
1	0.87	0.89	0.88	2457
2	0.89	0.87	0.88	3011
3	0.85	0.82	0.83	2974
accuracy			0.85	10554
macro avg	0.84	0.85	0.84	10554
weighted avg	0.85	0.85	0.85	10554

DT with BOW confusion matrix



DT with TFIDF confusion matrix



In training phase, the accuracy of DT model with BOW is 98%, and accuracy of DT with TFIDF is 98%
In testing phase, the accuracy of DT model with BOW is 86%, and accuracy of DT with TFIDF is 85%

KNN with BOW Training classification report

	precision	recall	f1-score	support
0	0.66	0.94	0.77	5002
1	0.91	0.79	0.84	5771
2	0.92	0.84	0.88	6939
3	0.90	0.81	0.85	6914
accuracy			0.84	24626
macro avg	0.85	0.84	0.84	24626
weighted avg	0.86	0.84	0.84	24626

KNN with TFIDF Training classification report

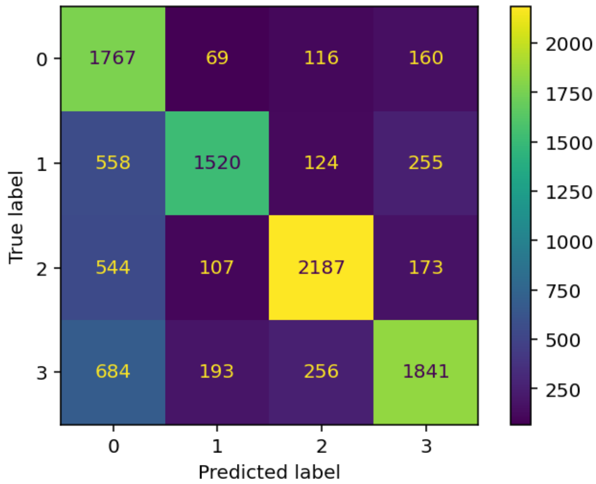
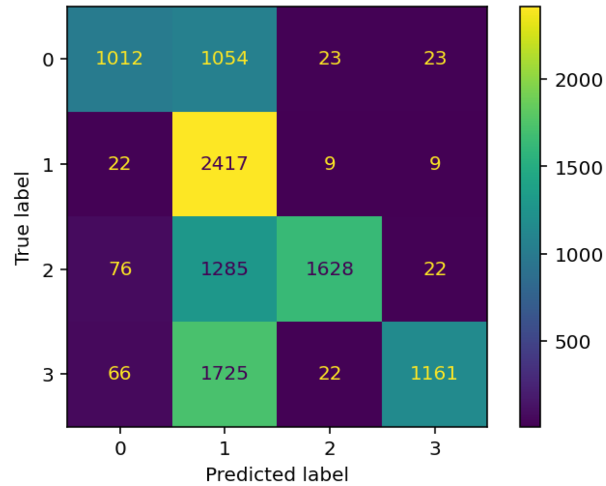
	precision	recall	f1-score	support
0	0.91	0.64	0.75	5002
1	0.45	0.99	0.62	5771
2	0.98	0.67	0.79	6939
3	0.97	0.54	0.70	6914
accuracy			0.70	24626
macro avg	0.83	0.71	0.71	24626
weighted avg	0.84	0.70	0.72	24626

KNN with BOW Testing classification report

	precision	recall	f1-score	support
0	0.50	0.84	0.62	2112
1	0.80	0.62	0.70	2457
2	0.82	0.73	0.77	3011
3	0.76	0.62	0.68	2974
accuracy			0.69	10554
macro avg	0.72	0.70	0.69	10554
weighted avg	0.73	0.69	0.70	10554

KNN with TFIDF Testing classification report

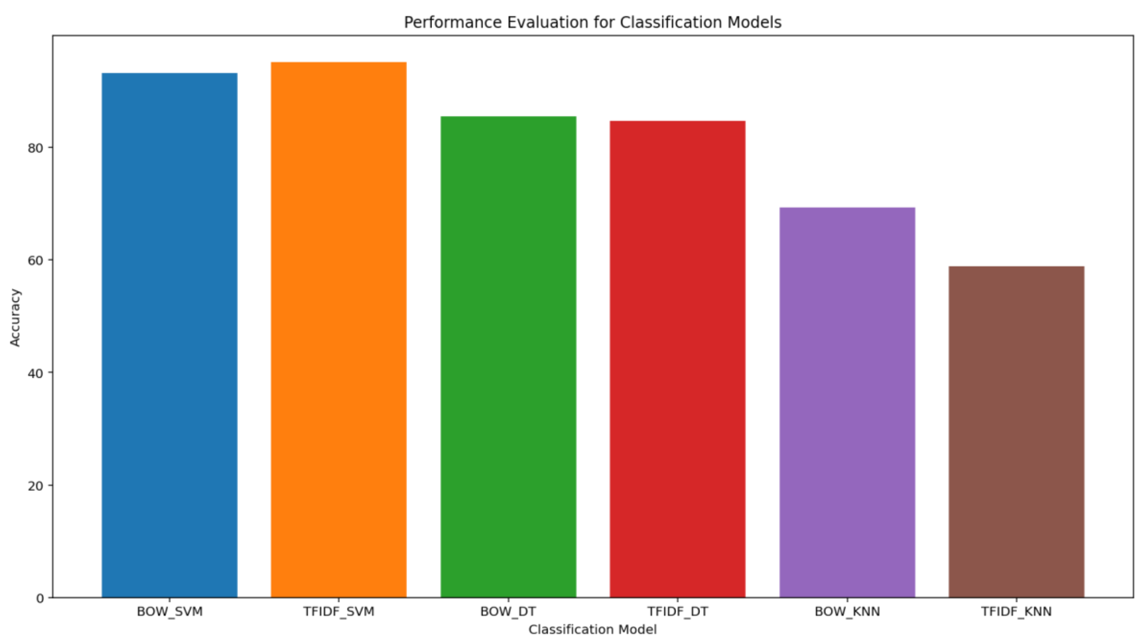
	precision	recall	f1-score	support
0	0.86	0.48	0.62	2112
1	0.37	0.98	0.54	2457
2	0.97	0.54	0.69	3011
3	0.96	0.39	0.55	2974
accuracy			0.59	10554
macro avg	0.79	0.60	0.60	10554
weighted avg	0.80	0.59	0.60	10554

KNN with BOW confusion matrix	KNN with TFIDF confusion matrix
	

In training phase, the accuracy of KNN model with BOW is 84%, and accuracy of KNN with TFIDF is 70%
In testing phase, the accuracy of KNN model with BOW is 69%, and accuracy of KNN with TFIDF is 59%

Compare between the accuracies of the 3 models:

	0	1	2	3	4	5
0	BOW_SVM	TFIDF_SVM	BOW_DT	TFIDF_DT	BOW_KNN	TFIDF_KNN
1	0.932064	0.951772	0.85541	0.847451	0.693102	0.589161



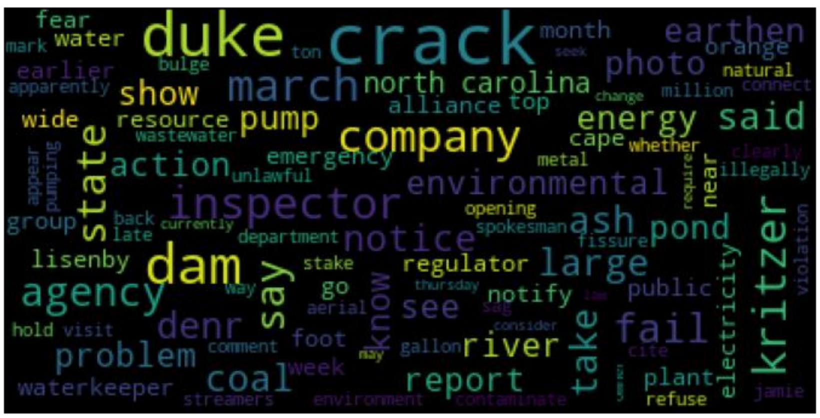
The highest accuracy model is the SVM model with TFIDF transformation.

5- Error analysis for the 3 models:

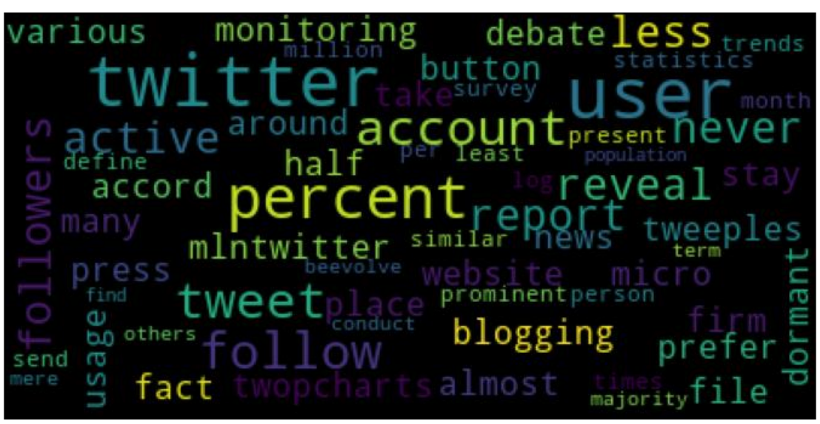
The most locations that the models misclassified are: 1, 2 and 35178, they are labels 0 and 3.

WordCloud for location 1

WordCloud for location 2



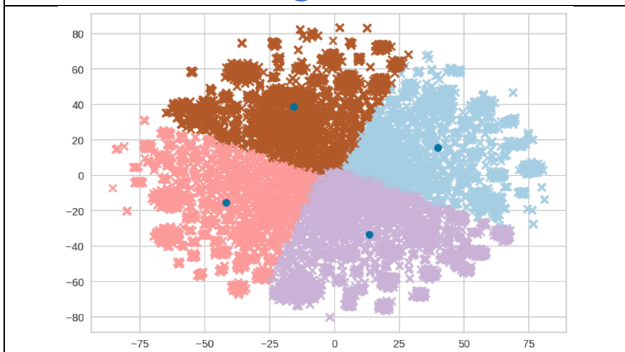
WordCloud for location 35178



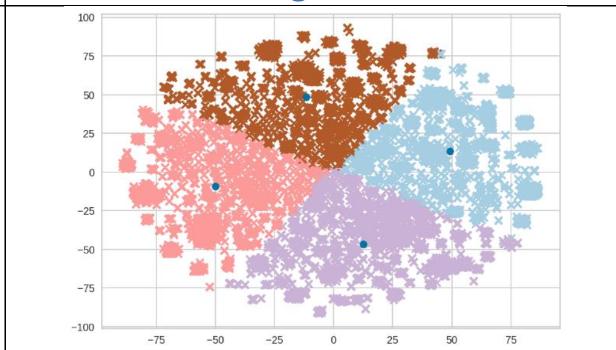
6- Perform KMeans clustering:

```
from yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans
def BuildingKMeansModel(clusters, X_data):
  kMeansModel= KMeans(n_clusters= clusters, init='k-means++', random_state=0)
  Y_Prediction = kMeansModel.fit_predict(X_data)
  return kMeansModel, Y_Prediction
```

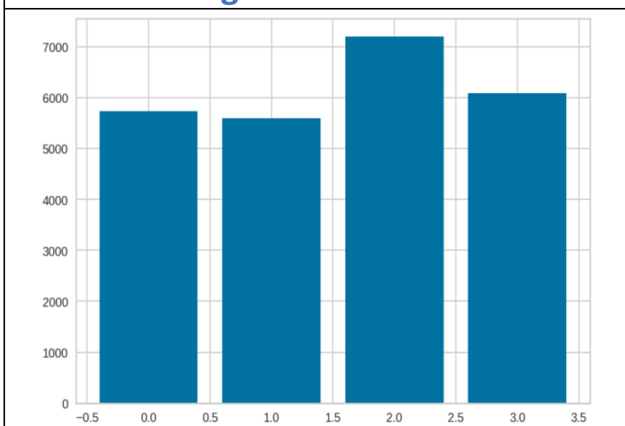
Clustering with BOW



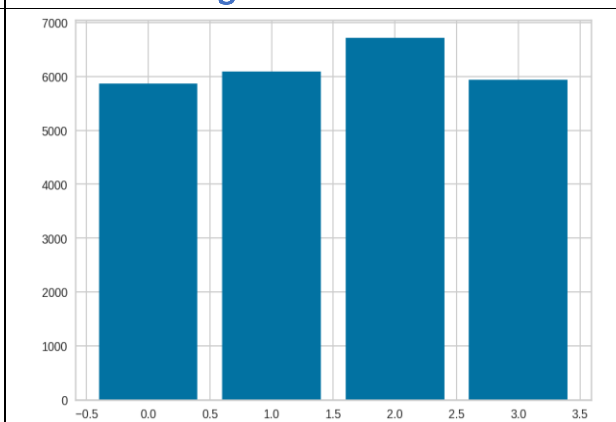
Clustering with TFIDF



Counting clusters with BOW

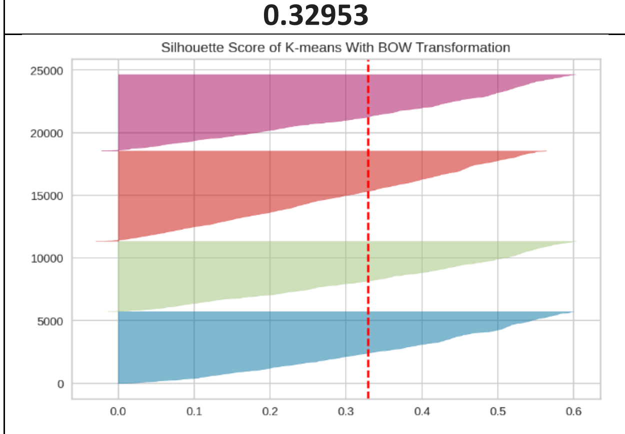


Counting clusters with TFIDF



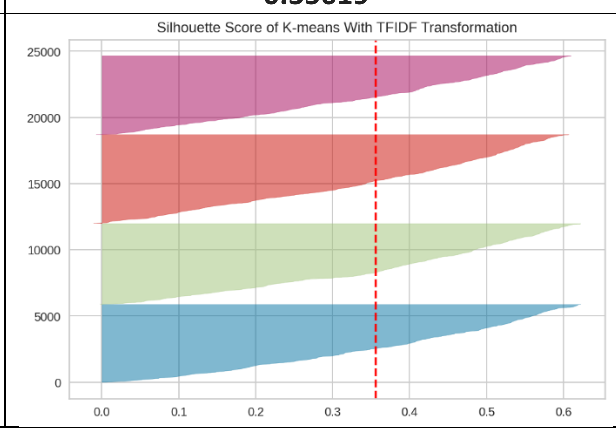
Silhoutte score with BOW

0.32953



Silhoutte score with TFIDF

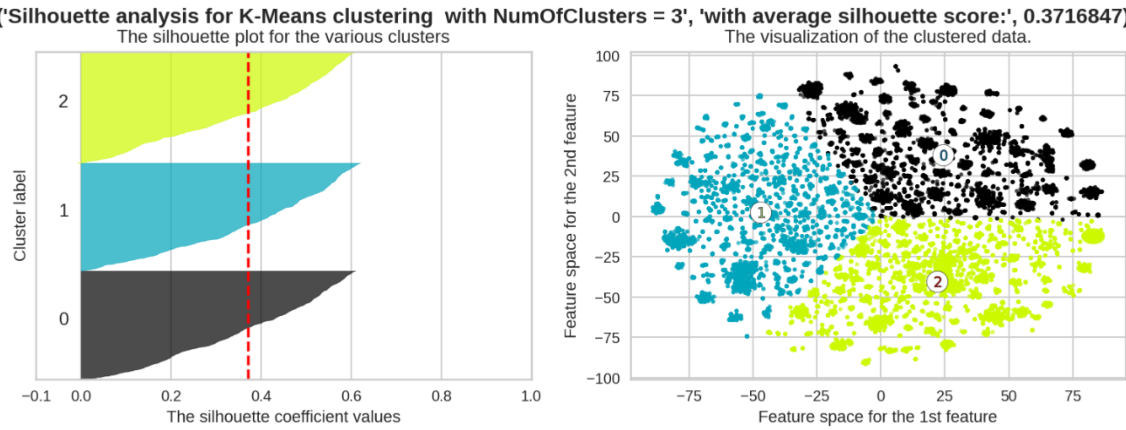
0.35619



7- Perform Error Analysis and choose the best number of k:

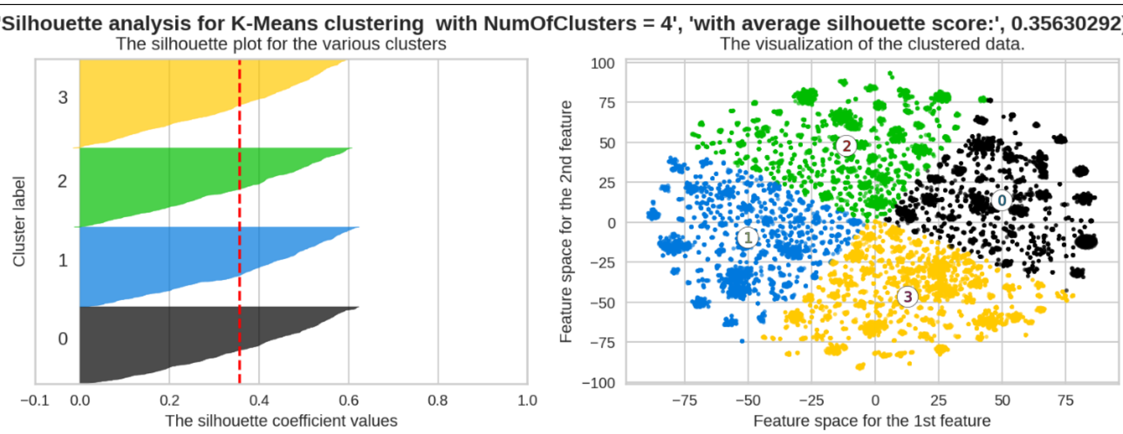
Silhouette score when K = 3

0.3716847



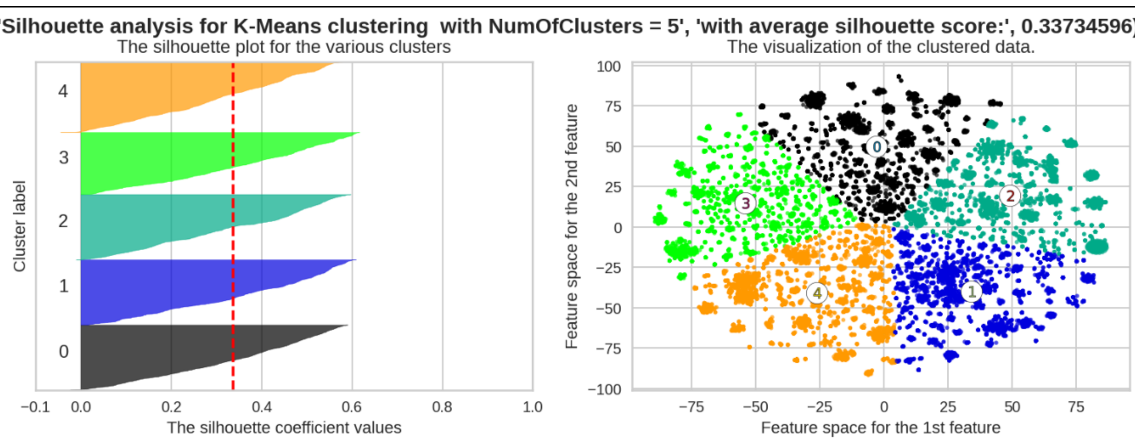
Silhouette score when K = 4

0.35630292



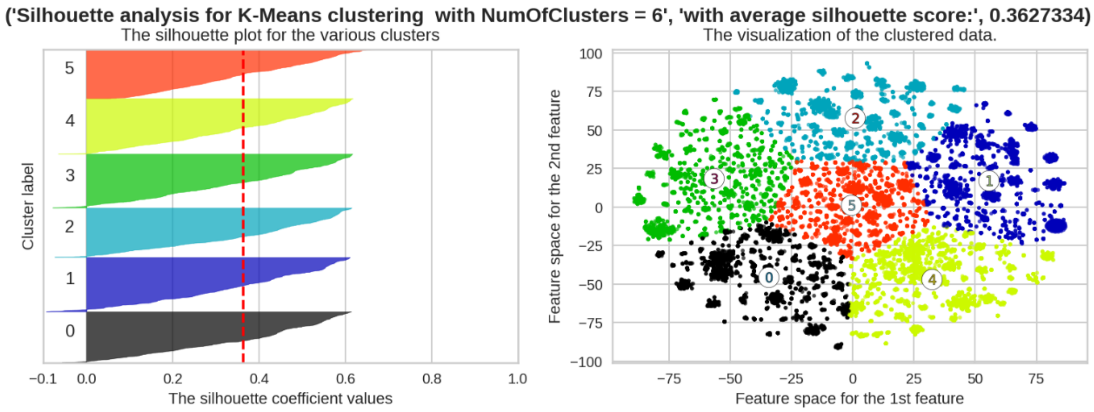
Silhouette score when K = 5

0.33734596



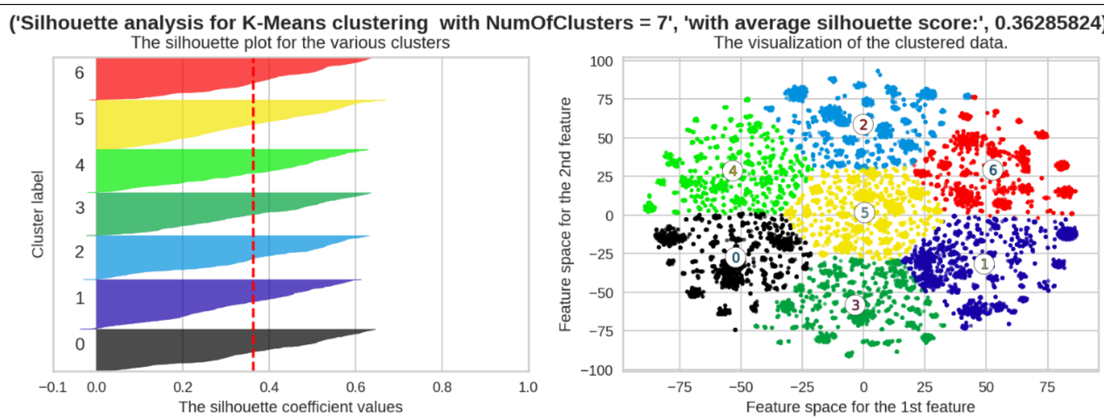
Silhouette score when K = 6

0.3627334



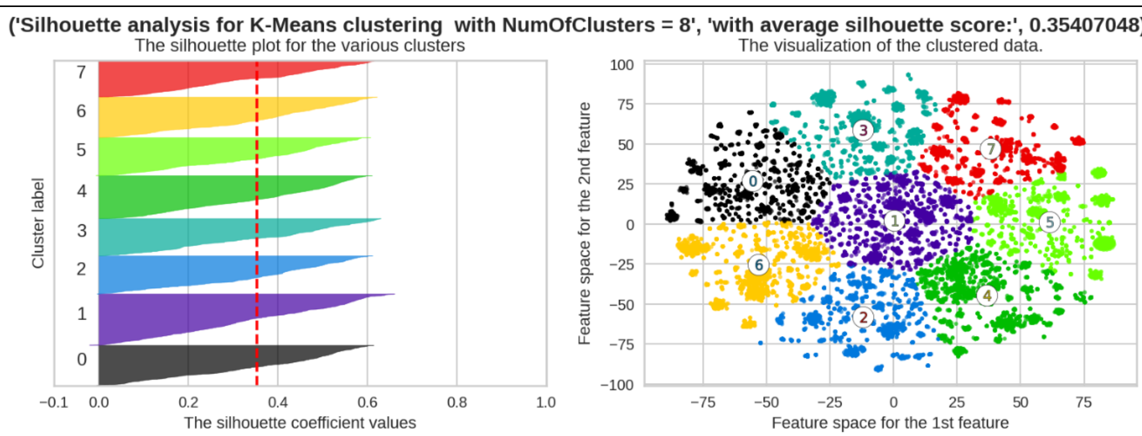
Silhouette score when K = 7

0.36285824



Silhouette score when K = 8

0.35407048



So, the best number of K is when K = 4

8- Perform LSA Text Summarization:

Latent Semantic Analysis is a robust algebraic-Statistical method which extracts hidden semantic structures of words and sentences. it extracts the features that cannot be directly mentioned. These features are essential to data, but are not original features of the dataset. It is an unsupervised approach along with the usage of Natural Language Processing (NLP).

1- Get the data for each topic:

```
#get data foreach topic
def get_data(n):
  topic=[]
  lab=np.array(cleaned_df['labelID'])
  data=np.array(cleaned_df['cleaned_text'])
  for i in range(len(cleaned_df)):
    if lab[i] == n:
      topic.append(data[i])


  topic_sum=[]

  data_sum=np.array(cleaned_df['cleaned_Text_summary'])
  for i in range(len(cleaned_df)):
    if lab[i] == n:
      topic_sum.append(data[i])

  return  topic , topic_sum
```

2- Build LSA function:

```
#LSA function
def lSa(topic):
  summarize_topic=[]
  for i in range(len(topic)):
    parser = PlaintextParser.from_string(topic[i]  ,Tokenizer("english"))
    for sentens in summarizer_lsa(parser.document, 1 ) :
      summarize_topic.append(sentens)
  return  summarize_topic
```

3- Build LSA data frame:

```
summarize_data = lSa(data)
```

```
summarize_data_lsa= pd.DataFrame(summarize_data)
summarize_data_lsa
```

	0
0	ally's plan approve federal reserve find bank ...
1	duke's recent environmental problem suggest se...
2	photo take earlier month show north carolina r...
3	accord ap report, cull information document in...
4	spokesman dave scanzoni say team begin work so...
...	...
35175	happen please make sure browser support javasc...
35176	mountain view base internet giant say titan ce...
35177	google purchase new mexico base unmanned aeria...
35178	atmospheric satellite could help bring interne...
35179	still early days, atmospheric satellite could ...

35180 rows × 1 columns

4- Let's compare between the human summary, and LSA summary:

	0	0
0	ally's plan approve federal reserve find bank ...	federal reserve approve ally financial inc.'s ...
1	duke's recent environmental problem suggest se...	major shareholder duke energy corp. call compa...
2	photo take earlier month show north carolina r...	photo take earlier month show north carolina r...
3	accord ap report, cull information document in...	thanks dog report associate press, know active...
4	spokesman dave scanzoni say team begin work so...	energy giant say committed clean dan river spi...
...	...	...
35175	happen please make sure browser support javasc...	happen please make sure browser support javasc...
35176	mountain view base internet giant say titan ce...	google inc nasdaq googl googl . nasdaq goog go...
35177	google purchase new mexico base unmanned aeria...	google purchase new mexico base unmanned aeria...
35178	atmospheric satellite could help bring interne...	google beaten world's large social network pur...
35179	still early days, atmospheric satellite could ...	google plan titan dronestitan aerospace's dron...

35180 rows × 2 columns

5- Count the Rouge score for the LSA summary:

```
dfs = [res[i][0] for i in range(len(res))]
dfscore = pd.DataFrame.from_records(dfs)
dfscore
```

	rouge-1	rouge-2	rouge-l
0	{'r': 0.5, 'p': 0.5, 'f': 0.4999999950000001}	{'r': 0.14285714285714285, 'p': 0.142857142857...	{'r': 0.375, 'p': 0.375, 'f': 0.37499999500000...
1	{'r': 0.125, 'p': 0.14285714285714285, 'f': 0....	{'r': 0.0, 'p': 0.0, 'f': 0.0}	{'r': 0.125, 'p': 0.14285714285714285, 'f': 0....
2	{'r': 1.0, 'p': 1.0, 'f': 0.999999995}	{'r': 1.0, 'p': 1.0, 'f': 0.999999995}	{'r': 1.0, 'p': 1.0, 'f': 0.999999995}
3	{'r': 0.125, 'p': 0.125, 'f': 0.1249999950000002}	{'r': 0.0, 'p': 0.0, 'f': 0.0}	{'r': 0.125, 'p': 0.125, 'f': 0.1249999950000002}
4	{'r': 0.2222222222222222, 'p': 0.2222222222222...	{'r': 0.0, 'p': 0.0, 'f': 0.0}	{'r': 0.2222222222222222, 'p': 0.2222222222222...
...	...	...	...
35175	{'r': 1.0, 'p': 1.0, 'f': 0.999999995}	{'r': 1.0, 'p': 1.0, 'f': 0.999999995}	{'r': 1.0, 'p': 1.0, 'f': 0.999999995}
35176	{'r': 0.14285714285714285, 'p': 0.111111111111...	{'r': 0.0, 'p': 0.0, 'f': 0.0}	{'r': 0.14285714285714285, 'p': 0.111111111111...
35177	{'r': 1.0, 'p': 1.0, 'f': 0.999999995}	{'r': 1.0, 'p': 1.0, 'f': 0.999999995}	{'r': 1.0, 'p': 1.0, 'f': 0.999999995}
35178	{'r': 0.125, 'p': 0.14285714285714285, 'f': 0....	{'r': 0.0, 'p': 0.0, 'f': 0.0}	{'r': 0.125, 'p': 0.14285714285714285, 'f': 0....
35179	{'r': 0.14285714285714285, 'p': 0.142857142857...	{'r': 0.0, 'p': 0.0, 'f': 0.0}	{'r': 0.14285714285714285, 'p': 0.142857142857...

35180 rows × 3 columns

6- Calculate the f1 score:

```
r=pd.DataFrame()
r['summarize_data_lsa']=summarize_data_lsa
r['cleaned_Text_summary']=cleaned_Text_summary
r['fscore_rouge_L']=flist
r['fscore_rouge_1']=flist1
r['fscore_rouge_2']=flist2
```

	summarize_data_lsa	cleaned_Text_summary	fscore_rouge_L	fscore_rouge_1	fscore_rouge_2
0	ally's plan approve federal reserve find bank ...	federal reserve approve ally financial inc.'s ...	0.375000	0.500000	0.142857
1	duke's recent environmental problem suggest se...	major shareholder duke energy corp. call compa...	0.133333	0.133333	0.000000
2	photo take earlier month show north carolina r...	photo take earlier month show north carolina r...	1.000000	1.000000	1.000000
3	accord ap report, cull information document in...	thanks dog report associate press, know active...	0.125000	0.125000	0.000000
4	spokesman dave scanzoni say team begin work so...	energy giant say committed clean dan river spi...	0.222222	0.222222	0.000000
...	...	...	...	...	...
35175	happen please make sure browser support javasc...	happen please make sure browser support javasc...	1.000000	1.000000	1.000000
35176	mountain view base internet giant say titan ce...	google inc nasdaq googl googl . nasdaq goog go...	0.125000	0.125000	0.000000
35177	google purchase new mexico base unmanned aeria...	google purchase new mexico base unmanned aeria...	1.000000	1.000000	1.000000
35178	atmospheric satellite could help bring interne...	google beaten world's large social network pur...	0.133333	0.133333	0.000000
35179	still early days, atmospheric satellite could ...	google plan titan dronestitan aerospace's dron...	0.142857	0.142857	0.000000

35180 rows × 5 columns

7- Let's see a sample of the LSA summary, which is closely to the human summary:

```
cleaned_df['cleaned_Text_summary'][4]
```

```
energy giant says committed clean dan river spill site increase
customer rate cover expense. duke energy add new element deal coal
ash basins. wednesday, company announce hire independent team
engineer review condition sites. spokesman dave scanzoni say team
begin work soon north carolina's governor pat mccrory approves
company's proposal. see issue need addressed, we'll take care
immediately. effort response dan river spill february
```

9- Perform BERT summarization:

BERT (Bidirectional transformer) is a transformer used to overcome the limitations of RNN and other neural networks as Long term dependencies. It is a pre-trained model that is naturally bidirectional. This pre-trained model can be tuned to easily to perform the NLP tasks as specified, Summarization in our case.

1- Divide the text into sentences.
2- Make a word embedding for each sentence.
3- Every sentence have a point.
4- Put every point in a cluster.
5- Calculate the distance between the point and the centroid.
6- Take the closest point to the centroid.
7- Extract the summary.

1- **Implement a function to tokenize text into sentences and get embedding for every sentence using sentence transformer:**

```
#function to tokenize text into sentences and get embeding for every sentence using sentence transformer
def text_to_sent_list(text,
                      embedder = SentenceTransformer('distilbert-base-nli-mean-tokens'),
                      min_len=2):

    ''' Returns cleaned article sentences and BERT sentence embeddings'''

    #convert to list of sentences

    sents = sent_tokenize(text)
    #remove short sentences by threshhold
    sents_clean = [sentence for sentence in sents if len(sentence)> min_len]
    #remove entries with empty list
    sents_clean = [sentence for sentence in sents_clean if len(sentence)!=0]
    #embed sentences (deafult uses BERT SentenceTransformer)
    sents_embedding= np.array(embedder.encode(sents_clean, convert_to_tensor=True))

    return sents_clean, sents_embedding
```

```
sents_clean, sents_embedding = text_to_sent_list(cleaned_df['cleaned_text'][1]) #trying on second row of dataset
print(sents_clean)
print(len(sents_clean)) #tokenized into 13 sentences
print(sents_embedding.shape) # each sentence is a 768 feature vector
```

2- Make the clustering:

```
from sklearn.cluster import KMeans
model = KMeans(n_clusters=5)
pred = model.fit_predict(X = sents_embedding)
print(pred) # cluster for k = 5 and  predict which sentence belongs to which cluster
print(len(model.cluster_centers_))
```

```
[2 2 1 2 2 2 1 1 0 4 2 3 1]
5
```

3- Calculate the distance between the points and the centroid:

```
#get eculidean distance between every sentence and each centroid
# get the index of the sentence closest to the centroid of every cluster
min_list=[]
from sklearn.metrics.pairwise import euclidean_distances
for j in range(5): # match this number with k
  min = euclidean_distances(model.cluster_centers_[j].reshape(1, -1), sents_embedding[0].reshape(1, -1))
  idx = 0
  for i in range(len(pred)):
    dist =  euclidean_distances(model.cluster_centers_[j].reshape(1, -1), sents_embedding[i].reshape(1, -1))
    if dist<min:
      idx = i
  min_list.append(idx)
print(min_list)
```

```
[10, 12, 10, 11, 9]
```

4- Generate the summary based on the clustering:

```
import sentence_transformers # get the indices of these same sentences from the original text and concatenate forming the summary
org_text = sent_tokenize(cleaned_df['text'][1])
print(len(org_text))
index_summ = list(set(min_list))
text =""
for i in range(len(index_summ)):
  text += org_text[index_summ[i]]
print(text)
```

5- Prepare for BERT summary:

```
cleaned_df['len_clean'] = [len(sent_tokenize(x)) for x in cleaned_df['cleaned_text']]
cleaned_df['len_org'] = [len(sent_tokenize(x)) for x in cleaned_df['text']]
```

```
cleaned_df_right = cleaned_df[cleaned_df['len_clean'] > cleaned_df['len_org']]
cleaned_df_left =  cleaned_df[cleaned_df['len_clean'] < cleaned_df['len_org']]
cleaned_df_special = cleaned_df[cleaned_df['len_clean'] == cleaned_df['len_org']]
```

6- Make a function for BERT summary:

```
def bert_with_kmeans(row, cleaned_df, k):
  sents_clean, sents_embedding = text_to_sent_list(cleaned_df['cleaned_text'][row]) #trying on second row of dataset
  print(sents_clean)
  print(len(sents_clean)) #tokenized into 13 sentences
  if len(sents_clean) < k:
        k = 1
  print(sents_embedding.shape) # each sentence is a 768 feature vector
  from sklearn.cluster import KMeans
  model = KMeans(n_clusters=k)
  pred = model.fit_predict(X = sents_embedding)
  print(pred) # cluster for k = 5 and  predict which sentence belongs to which cluster
  print(len(model.cluster_centers_))
  X = set(pred)
  print(X)
  #get eculidean distance between every sentence and each centroid
  #get the index of the sentence closest to the centroid of every cluster
  min_list=[]
  from sklearn.metrics.pairwise import euclidean_distances
  for j in range(k): # match this number with k
    min = euclidean_distances(model.cluster_centers_[j].reshape(1, -1), sents_embedding[0].reshape(1, -1))
    idx = 0
    for i in range(len(pred)):
      dist =  euclidean_distances(model.cluster_centers_[j].reshape(1, -1), sents_embedding[i].reshape(1, -1))
      if dist<min:
        idx = i
    min_list.append(idx)
  print(min_list)
```

```
  import sentence_transformers # get the indices of these same sentences from the original text and concatenate forming the summary
  org_text = sent_tokenize(cleaned_df['text'][row])
  print(len(org_text))
  index_summ = list(set(min_list))
  text =""
  if len(org_text) == len(sents_clean):
    for i in range(len(index_summ)):
      text += org_text[index_summ[i]]

  elif len(sents_clean) > len(org_text):
    for i in range(len(index_summ)):
      if index_summ[i] < len(org_text):
        text += org_text[index_summ[i]]
      else:
        adj_index = index_summ[i] - (len(sents_clean) - len(org_text))
        text += org_text[adj_index]
  else:
    for i in range(len(index_summ)):
      text += org_text[index_summ[i]]

  print(text)
  return text
```

So, the special_df is:

	label	text	Text_summary	labelID	cleaned_text	tokenized
0	business	The Federal Reserve approved Ally Financial In...	The Federal Reserve approved Ally Financial In...	0	federal reserve approve ally financial inc.'s ...	[federal, reserve, approve, ally, financial, i...
1	business	— Major shareholders of Duke Energy Corp. have...	— Major shareholders of Duke Energy Corp. have...	0	major shareholder duke energy corp. call compa...	[major, shareholder, duke, energy, corp., call...
2	business	Photos taken earlier this month show that Nort...	Photos taken earlier this month show that Nort...	0	photo take earlier month show north carolina r...	[photo, take, earlier, month, show, north, car...
3	business	Thanks to dogged reporting by the Associated P...	Thanks to dogged reporting by the Associated P...	0	thanks dog report associate press, know active...	[thanks, dog, report, associate, press, ,, kno...
5	business	TribLIVE's Daily and Weekly email newsletters ...	RALEIGH — Duke Energy Corp. said on Wednesday ...	0	triblive's daily weekly email newsletter deliv...	[triblive, 's, daily, weekly, email, newslette...

cleaned_Text_summary	summary_len	text_len	sentences	len_clean	len_org
federal reserve approve ally financial inc.'s ...	382	383	[federal reserve approve ally financial inc.'s...	2	2
major shareholder duke energy corp. call compa...	1037	2796	[major shareholder duke energy corp. call comp...	13	13
photo take earlier month show north carolina r...	799	3563	[photo take earlier month show north carolina ...	22	22
thanks dog report associate press, know active...	681	3269	[thanks dog report associate press, know acti...	18	18
raleigh duke energy corp. say wednesday move c...	899	1812	[triblive's daily weekly email newsletter deli...	12	12

Let's see the cleaned_df_right:

```
cleaned_df_right.head()
```

	label	text	Text_summary	labelID	cleaned_text	tokenized
4	business	The energy giant says it is committed to clean...	The energy giant says it is committed to clean...	0	energy giant say committed clean dan river spi...	[energy, giant, say, committed, clean, dan, ri...
11	business	By Suttinee Yuvejwattana and Michael Sin\n\nMa...	The Japanese satellite detected about a dozen ...	0	suttinee yuvejwattana michael sin march bloomb...	[suttinee, yuvejwattana, michael, sin, march, ...
21	business	Bangkok/Tokyo/Canberra: Over 300 new objects w...	According to a report from Tokyo, a Japanese s...	0	bangkok tokyo canberra new object spot satelli...	[bangkok, tokyo, canberra, new, object, spot, ...
24	business	BANGKOK: A Thai satellite has detected floatin...	BANGKOK: A Thai satellite has detected floatin...	0	bangkok thai satellite detect floating object ...	[bangkok, thai, satellite, detect, floating, o...
36	business	BANGKOK, Thailand – Thai satellite images have...	BANGKOK, Thailand – Thai satellite images have...	0	bangkok, thailand thai satellite image show fl...	[bangkok, ,, thailand, thai, satellite, image,...

cleaned_Text_summary	summary_len	text_len	sentences	len_clean	len_org
energy giant say committed clean dan river spi...	613	1392	[energy giant say committed clean dan river sp...	9	8
japanese satellite detect dozen piece possible...	647	4990	[suttinee yuvejwattana michael sin march bloom...	38	35
accord report tokyo, japanese satellite also s...	653	4114	[bangkok tokyo canberra new object spot satell...	27	26
bangkok thai satellite detect floating object ...	746	1004	[bangkok thai satellite detect floating object...	8	7
bangkok, thailand thai satellite image show fl...	781	1611	[bangkok, thailand thai satellite image show ...	12	11

Let's see the cleaned_df_left:

```
cleaned_df_left.head()
```

	label	text	Text_summary	labelID	cleaned_text	tokenized
7	business	Thank you for reading!\n\nPlease log in, or si...	Thank you for reading!\nPlease log in, or sign...	0	thank read please log in, sign new account pur...	[thank, read, please, log, in, ,, sign, new, a...
17	business	Colorado Springs, CO (80903)\n\nToday\n\nPartl...	Colorado Springs, CO (80903)TodayPartly cloudy...	0	colorado springs, co today partly cloudy. high...	[colorado, springs, ,, co, today, partly, clou...
22	business	The delivery schedule is part of a comprehensi...	The delivery schedule is part of a comprehensi...	0	delivery schedule part comprehensive recovery ...	[delivery, schedule, part, comprehensive, reco...
30	business	Do you support Iran's measure to reduce JCPOA ...	Do you support Iran's measure to reduce JCPOA ...	0	support iran's measure reduce jcpoa commitment...	[support, iran, 's, measure, reduce, jcpoa, co...
33	business	This transcript has been automatically generat...	This transcript has been automatically generat...	0	transcript automatically generate may accurate	[transcript, automatically, generate, may, acc

cleaned_Text_summary	summary_len	text_len	sentences	len_clean	len_org
thank read please log in, sign new account pur...	403	405	[thank read please log in, sign new account p...	4	5
colorado springs, co todaypartly cloudy. high ...	143	160	[colorado springs, co today partly cloudy. hi...	3	5
delivery schedule part comprehensive recovery ...	819	826	[delivery schedule part comprehensive recovery...	3	5
support iran's measure reduce jcpoa commitment...	267	274	[support iran's measure reduce jcpoa commitmen...	1	2
transcript automatically generate may accurate...	3832	3833	[transcript automatically generate may accurat...	2	3

10- Error Analysis for BERT summarization:

1- See the rows, where the model can't get it's summary as well.

```
Summaries = []
Scores = []
c = 1
for i in cleaned_df_test.index:
    print(c)
    text = bert_with_kmeans(i, cleaned_df_test, 5)
    human_summary = cleaned_df_test['Text_summary'][i]

    scorer = rouge_scorer.RougeScorer(['rouge1', 'rougeL'], use_stemmer=True)
    scores = scorer.score(text,human_summary)

    Summaries.append(text)
    Scores.append(scores)
    c += 1
```

2- Calculate the scores of the summary:

```
cleaned_df_test['Summaries'] = Summaries
cleaned_df_test['Scores'] = Scores
cleaned_df_test.head()
```

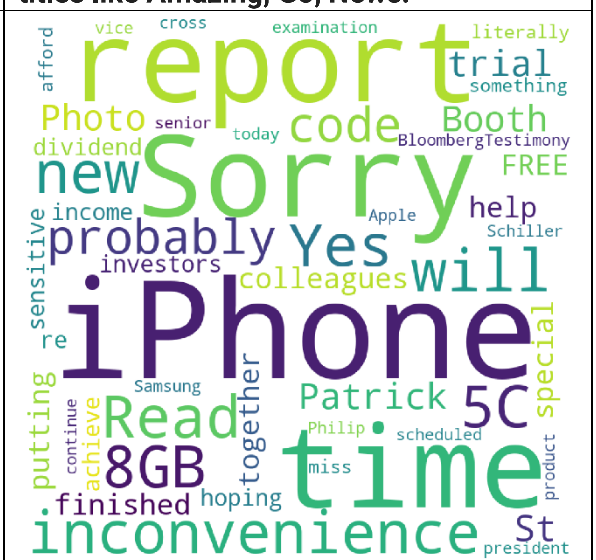
Summaries	Scores
"It's time for Gering and Minatare to get on b...	{'rouge1': (0.5398230088495575, 0.586538461538...
The Associated Press contributed to this repor...	{'rouge1': (0.05263157894736842, 0.24, 0.08633...
The airline had warned that should pilots reje...	{'rouge1': (0.5405405405405406, 0.5, 0.5194805...
Messina, who agreed to sell Intesa's Ukrainian...	{'rouge1': (0.17557251908396945, 0.33333333333...
Along with Colorado, Washington state has also...	{'rouge1': (0.28888888888888886, 0.70909090909...

3- Let’s see the Wordcloud for the error analysis:

According to human summary word clouds these rows to have relevant information each according to its own class

Let’s see what the machine tried to capture:

```
summary_words = ""
for i in df_sub_0.index:
    summary_words += df_sub_0['Summaries'][i]
    summary_words += " "
```

	
machine is focused more on names of news outlets and names of stock markets more than words correlated with the business world.	**The human summary focused on what is going to happen and when while the machine was distracted to flashy news titles like Amazing, Go, News.**
	
The human summary focused on food safety and the diseases related to corruption of food crops while the machine was distracted by where the news was found like twitter, podcast or wesite, but still managed to capture scary and rights which are in a way related to the topics in this rows but not enough.	**Machine seemed to have focused on unnecessary details here,that may be included in an article but focusing on the importance of technology instead of the real news like google did so or Apple released something new.**

11- Question and Answering System:

1- **Use the BERT pre-trained models for question and answering for extracting the answer from the summary:**

```
tokenizer = AutoTokenizer.from_pretrained('bert-base-cased')
tokenizer = BertTokenizer.from_pretrained('bert-large-uncased-whole-word-masking-finetuned-squad')
model = BertForQuestionAnswering.from_pretrained('bert-large-uncased-whole-word-masking-finetuned-squad')
```

2- **Make a function using torch for QA:**

```
import torch
def answer_question(question, answer_text):
    input_ids = tokenizer.encode(question, answer_text)
    print('Query has {:,} tokens.\n'.format(len(input_ids)))
    sep_index = input_ids.index(tokenizer.sep_token_id)
    num_seg_a = sep_index + 1
    num_seg_b = len(input_ids) - num_seg_a
    segment_ids = [0]*num_seg_a + [1]*num_seg_b
    assert len(segment_ids) == len(input_ids)
    outputs = model(torch.tensor([input_ids]),
                    token_type_ids=torch.tensor([segment_ids]),
                    return_dict=True)

    start_scores = outputs.start_logits
    end_scores = outputs.end_logits
    answer_start = torch.argmax(start_scores)
    answer_end = torch.argmax(end_scores)
    tokens = tokenizer.convert_ids_to_tokens(input_ids)
    answer = tokens[answer_start]
    for i in range(answer_start + 1, answer_end + 1):
        if tokens[i][0:2] == '##':
            answer += tokens[i][2:]
        else:
            answer += ' ' + tokens[i]
    print('Answer: "' + answer + '"')
    return answer
```

So, let's ask a question:

```
question = " what does Scanzoni says?"
answer=answer_question(question, y)
```

```
Query has 28 tokens.

Answer: "it will be several years before the state ' s basins are dismantled"
```

12- Innovation:

Our Innovation is a language translation for the summary:

1- From English language to German.

```
src = "en"
dst = "de"
task_name = f"translation_{src}_to_{dst}"
model_name = f"Helsinki-NLP/opus-mt-{src}-{dst}"
translator  = pipeline(task_name, model=model_name, tokenizer=model_name)
```

Input
"The energy giant says it is committed to cleaning up the Dan River spill site and will not increase customer rates to cover the expense.\nDuke Energy is adding a new element to dealing with its coal ash basins.\nWednesday, the company announced it will hire an independent team of engineers to review the condition of its sites.\nSpokesman Dave Scanzoni says the team will begin working as soon as North Carolina's Governor Pat McCrory approves the company's proposal.\n If we see any issues that need to be addressed, we'll take care of them immediately."This effort is in response the Dan River spill on February 2."
Output
Der Energieriese sagt, dass es sich zur Reinigung der Dan River Spill Website verpflichtet und wird nicht die Kundenpreise erhöhen, um die Kosten zu decken. Duke Energy fügt ein neues Element in den Umgang mit seinen Kohleasche Becken. Mittwoch, das Unternehmen kündigte an, es wird ein unabhängiges Team von Ingenieuren zu mieten, um den Zustand seiner Standorte zu überprüfen. Sprecher Dave Scanzoni sagt, dass das Team beginnt zu arbeiten, sobald North Carolina Gouverneur Pat McCrory genehmigt den Vorschlag des Unternehmens. Wenn wir irgendwelche Probleme sehen, die behandelt werden müssen, werden wir uns sofort um sie kümmern.

2- From English language to Chinese.

```
src = "en"
dst = "zh"
model, tokenizer = get_translation_model_and_tokenizer(src, dst)
```

```
inputs = tokenizer.encode(text_to_translate, return_tensors="pt", max_length=512, truncation=True)
print(inputs)
```

output
能源巨人说,该公司致力于清理丹河溢漏点,不会提高客户费率以支付费用。 杜克能源公司正在为其煤灰盆地的处理增加一个新的要素。 星期三,该公司宣布它将雇用一个独立的工程师小组来审查其矿址的状况。发言人戴夫·斯坎佐尼说,该小组将在北卡罗来纳州州长帕特·麦克罗里批准该公司提案后立即开始工作。如果我们看到任何需要解决的问题,我们将立即处理。 "这一努力是为了应对2月2日丹河泄漏事件。 ”

3- From English language to Arabic.

```
src = "en"
dst = "ar"
model, tokenizer = get_translation_model_and_tokenizer(src, dst)
```

```
# tokenize the text
inputs = tokenizer.encode(text_to_translate, return_tensors="pt", max_length=512, truncation=True)
beam_outputs = model.generate(
    inputs,
    num_beams=5,
    num_return_sequences=1,
    early_stopping=True,
)
for i, beam_output in enumerate(beam_outputs):
  print(tokenizer.decode(beam_output, skip_special_tokens=True))
```

output
يقول العامل في مجال الطاقة إنه ملتزم بتنظيف موقع انسكاب نهر دان ولن يزيد أسعار العملاء لتغطية النفقات. إن شركة الدوق للطاقة تضيف عنصراً جديداً للتعامل مع أحواض رماد الفحم لديها. يوم الأربعاء، أعلنت الشركة أنها سوف تستأجر فريقاً مستقلاً من المهندسين لاستعراض حالة مواقعها. والناطق باسم ديف سكانزوني يقول إن الفريق سوف يبدأ العمل بمجرد أن يوافق محافظ كارولينا الشمالية بات ماكروري على اقتراح الشركة. وإذا رأينا أي قضايا تحتاج إلى معالجة فسوف نعتني بها على الفور". وهذا الجهد يأتي استجابة لانسكاب نهر دان في الثاني من فبراير/شباط.

Conclusion:

In this project, we applied data pre-processing, classification techniques, and clustering. Then we applied LSA and Bert summarization models, after that we made a comparison between them.
The LSA model had a good sores and summary close to the human summary than the BERT model.
After that we made the error analysis to see what the machine tried to predict.
Then we made a simple question and answering system, to extract the answer from the summary.
Finally, we made a different language translation from English to French, Chinese, and Arabic languages.

References:

- https://ieeexplore-ieee-org.proxy.bib.uottawa.ca/document/9395976/figures#figures
- https://www.codegrepper.com/code-examples/python/NLP+text+summarization+with+LSA
- https://pypi.org/project/bert-extractive-summarizer/
- https://towardsdatascience.com/how-to-detect-and-translate-languages-for-nlp-project-dfd52af0c3b5