# HW 5: Clustering and Topic Modeling

Each assignment needs to be completed independently. Never ever copy others' work (even with minor modification, e.g. changing variable names). Anti-Plagiarism software will be used to check all submissions.

In this assignment, you'll need to use the following dataset:

- text_train.json: This file contains a list of documents. It's used for training models
- text_test.json: This file contains a list of documents and their ground-truth labels. It's used for testing performance. This file is in the format shown below. Note, a document may have multiple labels.

**Note: due to randomness, every time you run your clustering models, you may get different results. To ease the grading process, once you get satisfactory results, please save your notebook as a pdf file (Jupyter notebook menu File -> Print -> Save as pdf), and submit this pdf along with your .py code.**

In [24]:

```
import pandas as pd

# Add your import statement

from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn import metrics
from nltk.corpus import stopwords
from nltk.cluster import KMeansClusterer, cosine_distance
from sklearn import metrics
from sklearn.metrics import classification_report
import numpy as np
from sklearn.cluster import KMeans
from sklearn import mixture
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.metrics.pairwise import cosine_similarity
```

In [2]:

```
train_data = pd.read_csv("hw5_train.csv")
train_data.head()

test_data = pd.read_csv("hw5_test.csv")
test_data.head()
```

Out[2]:

| | text |
|---|---|
| **0** | blm in wyo to begin deciding on backlogged lea... |
| **1** | report amtrak loss comes to per passenger u s ... |
| **2** | medicare key in races washington an upset vict... |
| **3** | sunnyvale bicyclist dies of injuries suffered ... |
| **4** | mozambique upbeat on debt crisis investors not... |

Out[2]:

| | text | T1 | T2 | T3 |
|---|---|---|---|---|
| **0** | child asylum seekers targeted in home office b... | 0 | 1 | 0 |
| **1** | obama acknowledges economic stress not so long... | 0 | 1 | 0 |
| **2** | help not soonbyline by ted ralltime fri jul pm... | 0 | 1 | 0 |
| **3** | un pakistan flood misery exceeds tsunami haiti... | 1 | 0 | 0 |
| **4** | new home sales plunge new home sales plunge we... | 0 | 1 | 0 |

# Q1: K-Mean Clustering (5 points)

Define a function `cluster_kmean(train_data, test_data, num_clusters, min_df = 1, stopwords = None, metric = 'cosine')` as follows:

- Take two dataframes as inputs: `train_data` is the dataframe loaded from `hw5_train.csv`, and `test_data` is the dataframe loaded from `hw5_test.csv`
- Use **KMeans** to cluster documents in `train_data` into 3 clusters by the distance metric specified. Tune the following parameters carefully:
  - `min_df` and `stopword` options in generating TFIDF matrix. You may need to remove corpus-specific stopwords in addition to the standard stopwords.
  - distance metric: `cosine` or `Euclidean` distance
  - sufficient iterations with different initial centroids to make sure clustering converges
- Test the clustering model performance using `test_data`:
  - Predict the cluster ID for each document in `test_data`.
  - Apply `majority vote` rule to dynamically map each cluster to a ground-truth label in `test_data`.
    - Note a small percentage of documents have multiple labels. For these cases, you can randomly pick a label during the match
    - Be sure `not to hardcode the mapping`, because a cluster may corrspond to a different topic in each run. (hint: if you use pandas, look for `idxmax` function)
  - Calculate `precision/recall/f-score` for each label. Your best F1 score on the test dataset should be around `80%`.
- Assign a meaninful name to each cluster based on the `top keywords` in each cluster. You can print out the keywords and write the cluster names as markdown comments.

- This function has no return. Print out confusion matrix, precision/recall/f-score.

**Analysis**:

- Comparing the clustering with cosine distance and that with Euclidean distance, do you notice any difference? Which metric works better here?
- How would the stopwords and min_df options affect your clustering results?

In [47]:
```python
def cluster_kmean(train_data, test_data, num_clusters, min_df = 1,\
                  stop_words = None, metric = 'cosine'):

    if stop_words:
        stopwords_list = stopwords.words('english') + \
        ["said","says","please","well","year"]
        tfidf_vect = TfidfVectorizer(stop_words=stopwords_list,\
                                     min_df = min_df)
    else:
        tfidf_vect = TfidfVectorizer(min_df = min_df)

    # generate tfidf matrix
    X_train = tfidf_vect.fit_transform(train_data["text"])
    # generate tifid for new documents
    X_test = tfidf_vect.transform(test_data["text"])

    if metric == 'cosine':
        clusterer = KMeansClusterer(num_clusters, cosine_distance,\
                                    repeats=15)
        clusters = clusterer.cluster(X_train.toarray(), \
                             assign_clusters=True)
        predicted = [clusterer.classify(v) for v in X_test.toarray()]

        # find top words at centroid of each cluster
        centroids=np.array(clusterer.means())
        # the matrix in ascending order
        sorted_centroids = centroids.argsort()[:, ::-1]

    else:
        clusterer = KMeans(n_clusters=num_clusters, n_init=20).fit(X_train)
        predicted = clusterer.predict(X_test)
        centroids = clusterer.cluster_centers_
        sorted_centroids = centroids.argsort()[:, ::-1]


    voc_lookup= tfidf_vect.get_feature_names_out()

    truth_label = test_data[["T1","T2","T3"]].idxmax(axis = 1)
    confusion_df = pd.DataFrame(list(zip(truth_label.values, predicted)),\
                        columns = ["label", "cluster"])

    # generate crosstab between clusters and true labels
    matrix = pd.crosstab(index=confusion_df.cluster, \
                      columns=confusion_df.label)
    cluster_dict = dict(matrix.idxmax(axis=1))
    print(matrix)
    for i in range(num_clusters):
     # get words with top 20 tf-idf weight in the centroid
        top_words=[voc_lookup[word_index] \
                   for word_index in sorted_centroids[i, :10]]
```

```
        print(f"Cluster {i} -> Topic {cluster_dict[i]}\ntop words: {top_words}'

    # Map true label to cluster id
    predicted_target=[cluster_dict[i] \
                      for i in predicted]

    print(metrics.classification_report\
          (truth_label, predicted_target))
```

In [44]:
```
# Clustering by cosine distance
cluster_kmean(train_data, test_data, num_clusters=3, \
              min_df = 1, stop_words = True, metric = 'cosine')
```

```
label      T1   T2   T3
cluster
0          56    8  169
1         143    3    4
2          15  195    7
Cluster 0 -> Topic T3
 top words: ['crash', 'bus', 'rail', 'plane', 'train', 'passengers', 'police',
'airlines', 'cruise', 'car']
Cluster 1 -> Topic T1
 top words: ['oil', 'people', 'bp', 'japan', 'fire', 'water', 'spill', 'gulf',
'disaster', 'nuclear']
Cluster 2 -> Topic T2
 top words: ['percent', 'tax', 'economy', 'rate', 'obama', 'comment', 'governm
ent', 'economic', 'would', 'billion']
              precision    recall  f1-score   support

          T1       0.95      0.67      0.79       214
          T2       0.90      0.95      0.92       206
          T3       0.73      0.94      0.82       180

    accuracy                           0.84       600
   macro avg       0.86      0.85      0.84       600
weighted avg       0.87      0.84      0.84       600
```

Assign a meaninful name to each cluster based on the top keywords:

- Topic 1: Disaster
- Topic 2: Economic
- Topic 3: Transportation

In [46]:
```
# Clustering by Euclidean distance

cluster_kmean(train_data, test_data, num_clusters=3, min_df = 2,\
              stop_words = True, metric = 'euclidean')
```

```
label      T1   T2   T3
cluster
0          38    9    0
1           4  149    1
2         172   48  179
Cluster 0 -> Topic T1
 top words: ['oil', 'bp', 'comment', 'spill', 'gulf', 'users', 'sign', 'rate', 
'news', 'disliked']
Cluster 1 -> Topic T2
 top words: ['percent', 'tax', 'economy', 'obama', 'economic', 'government', 
'billion', 'would', 'debt', 'bank']
Cluster 2 -> Topic T3
 top words: ['people', 'police', 'crash', 'bus', 'fire', 'plane', 'rail', 'cit
y', 'train', 'new']
              precision    recall  f1-score   support

          T1       0.81      0.18      0.29       214
          T2       0.97      0.72      0.83       206
          T3       0.45      0.99      0.62       180

    accuracy                           0.61       600
   macro avg       0.74      0.63      0.58       600
weighted avg       0.76      0.61      0.57       600
```

- When comparing two models, we observe that the clustering model using cosine distance takes a longer time to run but provides better results, while the model using Euclidean distance runs relatively fast but produces unstable results.
- The stopwords and min_df options affect my clustering results:
  - Stopword: Stopwords are common words that do not add much meaning to the text, such as "the," "and," "a," etc. Keeping them in the clustering model can lead to them being identified as the most important words, which is not helpful for identifying meaningful cluster names.
  - min_df: this parameter will sets the minimum frequency for words to be considered in the clustering model, it will focus on the main topic for each document. We can focus on the words that are more frequent and important in the documents, which can help us to identify the most meaningful topics for each cluster.

# Q2: GMM Clustering (5 points)

Define a function `cluster_gmm(train_data, test_data, num_clusters, min_df = 10, stopwords = stopwords)` to redo Q1 using the Gaussian mixture model.

**Requirements**:

- To save time, you can specify the covariance type as `diag`.
- Be sure to run the clustering with different initiations to get stabel clustering results
- Your F1 score on the test set should be around `70%` or higher.

In [61]:
```
def cluster_gmm(train_data, test_data, num_clusters, min_df = 10,\
                stopwords = stopwords):
```

```
    if stopwords:
        stopwords_list = stopwords.words('english') + \
        ["said","says","please","well","year","would","years","may",\
         "last","one","two","people"]
        tfidf_vect = TfidfVectorizer(stop_words=stopwords_list, \
                                     min_df = min_df)
    else:
        tfidf_vect = TfidfVectorizer(min_df = min_df)

    # generate tfidf matrix
    X_train = tfidf_vect.fit_transform(train_data["text"])
    # generate tifid for new documents
    X_test = tfidf_vect.transform(test_data["text"])

    gmm = mixture.GaussianMixture\
    (n_components=num_clusters, covariance_type='diag',n_init=15)\
    .fit(X_train.toarray())
    predicted = gmm.predict(X_test.toarray())

    voc_lookup= tfidf_vect.get_feature_names_out()

    truth_label = test_data[["T1","T2","T3"]].idxmax(axis = 1)
    confusion_df = pd.DataFrame(list(zip(truth_label.values, predicted)),\
                                columns = ["label", "cluster"])

    # generate crosstab between clusters and true labels
    matrix = pd.crosstab(index=confusion_df.cluster, \
                         columns=confusion_df.label)
    cluster_dict = dict(matrix.idxmax(axis=1))
    print(matrix)

    predicted_target=[cluster_dict[i] for i in predicted]

    print(metrics.classification_report(truth_label, predicted_target))
```

In [65]:
```
cluster_gmm(train_data, test_data, num_clusters=3, \
            min_df = 10, stopwords = stopwords)
```

```
label      T1   T2   T3
cluster
0          29   36  118
1         139    3   59
2          46  167    3
              precision    recall  f1-score   support

          T1       0.69      0.65      0.67       214
          T2       0.77      0.81      0.79       206
          T3       0.64      0.66      0.65       180

    accuracy                           0.71       600
   macro avg       0.70      0.71      0.70       600
weighted avg       0.71      0.71      0.71       600
```

# Q3: LDA Clustering (5 points)

**Q3.1.** Define a function `cluster_lda(train_data, test_data, num_clusters, min_df = 5, stopwords = stopwords)` to redo Q1 using the LDA model. Note, for LDA, you need to use `CountVectorizer` instead of `TfidfVectorizer`.

**Requirements**:

- Your F1 score on the test set should be around `80%` or higher
- Print out top-10 words in each topic
- Return the topic mixture per document matrix for the test set(denoted as `doc_topics`) and the trained LDA model.

**Q3.2**. Find similar documents

- Define a function `find_similar_doc(doc_id, doc_topics)` to find `top 3 documents` that are the most thematically similar to the document with `doc_id` using the `doc_topics`. (1 point)
- Return the IDs of these similar documents.
- Print the text of these documents to check if their thematic similarity.

**Analysis**:

You already learned how to find similar documents by using TFIDF weights. Can you comment on the difference between the approach you just implemented with the one by TFID weights?

In [66]:
```
def cluster_lda(train_data, test_data, num_clusters, min_df = 5,\
                stopwords = stopwords):

    model, doc_topic = None, None
    # generate a new list of stopword
    specific_stopword = ["said","says","please","well","would","www",\
                         "com","bp","percent","also","may","year","new",\
                         "last","one","two","people","rate","sign"]

    if stopwords:
        stopwords_list = list(stopwords.words('english')) + specific_stopword
        tf_vectorizer = CountVectorizer(stop_words=stopwords_list, \
                                        min_df = min_df)
    else:
        tf_vectorizer = CountVectorizer(min_df = min_df)

    X_train = tf_vectorizer.fit_transform(train_data["text"])
    X_test = tf_vectorizer.transform(test_data["text"])
    # Train LDA model
    model = LatentDirichletAllocation(n_components=num_clusters, \
                                      max_iter=20,
                                      evaluate_every=1, n_jobs=1,
                                      random_state=0).fit(X_train)
    # Generate topic assignment of each document in the test set
    doc_topic = model.transform(X_test)
    # to take the position of the highest value
    predicted = doc_topic.argmax(axis=1)
    # to get the name of words
```

```
    voc_lookup= tf_vectorizer.get_feature_names_out()
    # Create a dataframe with cluster id and
    # ground truth label
    truth_label = test_data[["T1","T2","T3"]].idxmax(axis = 1)
    confusion_df = pd.DataFrame(list(zip(truth_label.values, predicted)),\
                                columns = ["label", "cluster"])

    # generate crosstab between clusters and true labels
    matrix = pd.crosstab(index=confusion_df.cluster, \
                         columns=confusion_df.label)
    # Map cluster id to true labels by "majority vote"
    cluster_dict = dict(matrix.idxmax(axis=1))
    print(matrix)

    num_top_words=10

    for topic_idx, topic in enumerate(model.components_):
        # print out top 10 words per topic
        top_words=[voc_lookup[i] \
                   for i in topic.argsort()[::-1][0:num_top_words]]
        print(f"Topic {topic_idx}: {cluster_dict[topic_idx]}\ntop words:{top_wo
        print("\n")
    # Map true label to cluster id
    predicted_target=[cluster_dict[i] \
                      for i in predicted]

    print(metrics.classification_report\
          (truth_label, predicted_target))

    return model, doc_topic
```

In [67]:

```
# Test LDA model
model, doc_topics = cluster_lda(train_data, test_data, num_clusters = 3 ,\
                                min_df = 5, stopwords = stopwords)
```

```
label      T1   T2   T3
cluster
0          57   15  151
1         139    1   11
2          18  190   18
Topic 0: T3
top words:['comment', 'news', 'users', 'rail', 'travel', 'crash', 'passenger
s', 'service', 'car', 'plane']


Topic 1: T1
top words:['oil', 'japan', 'water', 'fire', 'officials', 'government', 'disast
er', 'city', 'nuclear', 'could']


Topic 2: T2
top words:['tax', 'government', 'obama', 'economy', 'billion', 'economic', 'mi
llion', 'market', 'money', 'president']


              precision    recall  f1-score   support

          T1       0.92      0.65      0.76       214
          T2       0.84      0.92      0.88       206
          T3       0.68      0.84      0.75       180

    accuracy                           0.80       600
   macro avg       0.81      0.80      0.80       600
weighted avg       0.82      0.80      0.80       600
```

In [68]:

```
def find_similar(doc_id, doc_topics):

    # Get the row of the document with doc_id
    id_topic = doc_topics[doc_id]

    # Compute the cosine of angle between two vectors,
    # it means between d_topic with each row of doc_topics

    similarities = np.dot(doc_topics, id_topic)\
    / (np.linalg.norm(doc_topics, axis=1) * np.linalg.norm(id_topic))
    # Get the top 3 highest score from similarities
    docs = np.argsort(similarities)[::-1][1:4]

    return docs
```

In [69]:

```
doc_topics[10:15]

doc_id = 11
idx = find_similar(doc_id, doc_topics)

print(test_data.text.iloc[doc_id])
print("Similar documents: \n")
for i in idx:
    print(i,"-", test_data.iloc[i].text)
```

Out[69]:
```
array([[0.37309574, 0.1535742 , 0.47333006],
       [0.01706364, 0.34609139, 0.63684497],
       [0.57436741, 0.0008928 , 0.42473979],
       [0.05198249, 0.77565863, 0.17235888],
       [0.73932828, 0.00158627, 0.25908545]])
```

obama says he s finding out whose ass to kick over gulf disasterbyline time tu
e jun am et is president obama bowing to criticism that he hasn t shown enough
emotion and outrage about the gulf of mexico oil spill in an interview with th
e today show s matt lauer this morning the president offered his most candid r
esponse yet about the disaster bluntly telling lauer he s been talking to expe
rts about whose ass to kick when it comes to responsibility for the mess i was
down there a month ago before most of these talking heads were even paying att
ention to the gulf a month ago i was meeting with fishermen down there standin
g in the rain talking about what a potential crisis this could be obama said d
efending his administration s handling of the spill and i don t sit around jus
t talking to experts because this is a college seminar we talk to these folks
because they potentially have the best answers so i know whose ass to kick tha
t s a pretty sharp response for a president known for his cool headed approach
to situations in recent weeks as obama was assailed by critics for not being e
xpressive enough in his response to the spill white house officials defended h
is reaction by suggesting voters would prefer to see concrete actions over emp
ty method acting yet administration officials are not ignorant of polls showin
g the nation less than thrilled with obama s handling of the gulf according to
the latest abc washington post poll more than two thirds of those polled perce
nt disapprove of the federal government s handling of the spill that s higher
than the outrage over the bush administration s handling of hurricane katrina
holly bailey is a senior political writer for yahoo news

Similar documents:

169 - feds bp agrees to expedite oil spill payments the obama administration s
ays bp has agreed to expedite the payment of claims to businesses and individu
als whose livelihoods have been disrupted by the gulf of mexico oil spill trac
y wareing wehr ing who is with the national incident command office told repor
ters in washington that the understanding on payment of claims came in a meeti
ng wednesday with bp executives including ceo tony hayward wareing said admini
stration officials raised a pressing concern about the time bp has been taking
to provide relief payments particularly to businesses in the stricken area she
said the company will change the way it processes such claims and will expedit
e payments among other things it will drop the current practice of waiting to
make such payments until businesses have closed their books for each month

474 - feds bp agrees to expedite oil spill payments washington the obama admin
istration says bp has agreed to expedite the payment of claims to businesses a
nd individuals whose livelihoods have been disrupted by the gulf of mexico oil
spill tracy wareing wehr ing who is with the national incident command office
told reporters in washington that the understanding on payment of claims came
in a meeting wednesday with bp executives including ceo tony hayward wareing s
aid administration officials raised a pressing concern about the time bp has b
een taking to provide relief payments particularly to businesses in the strick
en area she said the company will change the way it processes such claims and
will expedite payments among other things it will drop the current practice of
waiting to make such payments until businesses have closed their books for eac
h month

167 - world bank waives haiti debt payments the world bank said thursday it wa
s waiving haiti s debt payments for the next five years due to the devastation
caused by the earthquake and is studying efforts to cancel the nation s remain
ing debt in a statement the washington based multilateral lender said haiti s
debt to the world bank which is interest free is about million dollars or arou
nd four percent of haiti s total external debt due to the crisis caused by the
earthquake we are waiving any payments on this debt for the next five years an
d at the same time we are working to find a way forward to cancel the remainin
g debt the statement said last week the world bank said it planned to provide
an additional million dollars in emergency aid to haiti after the january eart

```
hquake ravaged the impoverished nation officials fear up to people were killed
since the development lender said it has provided grants interest free aid of
million dollars to the caribbean nation the poorest country in the western hem
isphere that amount does not include the million dollars in grants announced o
n january it said the world bank and its sibling institution the international
monetary fund classify haiti among heavily indebted poor countries that are el
igible for debt forgiveness haiti was granted billion dollars in debt relief l
ast june
```

### Analysis:

- To identify similar documents, I plan to utilize the topic mixture of each document. Each row can be considered as a vector, and the cosine similarity between each document vector will be calculated. If the similarity value is high, it indicates that the documents are similar. $$cos(\alpha)=\frac{a.b}{|a||b|}$$

## Q4 (Bonus): Find the most significant topics in a document

A small portion of documents in our dataset have multiple topics. For instace, consider the following document which has topic T2 and T3. The LDA model returns two significant topics with probabilities 0.355 and 0.644. Can you describe a way to find out most significant topics in documents but ignore the insignificant ones? In this example, you should ignore the first topic but keep the last two.

- Implement your ideas
- Test your ideas with the test set
- Recalculate the precision/recall/f1 score for each label.

In [135…
```
(test_data.reset_index()).iloc[12:13]
doc_topics[12]
```

Out[135]:

| | index | text | T1 | T2 | T3 |
|---|---|---|---|---|---|
| **12** | 12 | white house to dole out billion for fast train... | 0 | 1 | 1 |

Out[135]:
```
array([0.00091134, 0.35500994, 0.64407872])
```

In my oppinion, I will use a threshold approach to identify the most significant topics in a document, particularly to determine whether a document has two topics or not.

- We know the sum of topic mixtures for each document is $\theta_1+\theta_2+\theta_3 =1$, presenting the proportion of each topic per document. I assume that if a document has 2 topics, the two topic mixtures will have higher values compared to the remaining topic. Therefore, I set the threshold at 1/3, where 3 is the total number of topics.
- I utilize the result from the majority vote technique in the 3rd question, which provides the topic distribution of each document in a fixed order (e.g., T3 in position 0, T1 in position 1, and T2 in position 2).

- Using np.where, I can identify the positions that have values higher than the threshold and assign 1 to those positions, and 0 otherwise. It is important to note that the assigned values are based on the position of topics from the third question.

In [70]:
```
y_pred = []
threshold = 1/3
# Iterate over each document's topic mixture
for doc in doc_topics:
    each = [0,0,0]
    # Find the topics with mixtures greater than the threshold
    compar_thres = list(np.where(doc>threshold)[0])
    # Assign 1 to the topic with the mixture value higher than threshold
    for index in compar_thres:
        if index==0:
            each[2] = 1
        elif index==1:
            each[0] = 1
        else:
            each[1] = 1
    y_pred.append(each)
# Get the true labels from the test data
y_test = test_data[["T1","T2","T3"]].values
# Print the precision/recall/f1 score for each label.
print(classification_report(y_test, np.array(y_pred)))
```

```
              precision    recall  f1-score   support

           0       0.82      0.79      0.80       214
           1       0.78      0.98      0.87       207
           2       0.67      0.92      0.77       197

   micro avg       0.75      0.89      0.82       618
   macro avg       0.76      0.90      0.82       618
weighted avg       0.76      0.89      0.82       618
 samples avg       0.80      0.90      0.83       618
```

In [73]:
```
if __name__ == "__main__":

    # Due to randomness, you won't get the exact result
    # as shown here, but your result should be close
    # if you tune the parameters carefully

    # Q1
    print("----Question 1----\n")
    print("With cosine distance:\n")
    cluster_kmean(train_data, test_data, num_clusters=3, min_df = 1,\
                  stop_words = True, metric = 'cosine')
    print("\nWith Euclidean distance:\n")
    cluster_kmean(train_data, test_data, num_clusters=3, min_df = 2,\
                  stop_words = True, metric = 'euclidean')
    print("\n")
    # Q2
    print("----Question 2----\n")
    cluster_gmm(train_data, test_data, num_clusters=3, min_df = 10,\
                stopwords = stopwords)
    # Q3
    print("\n----Question 3----\n")
    print("Q.3a:\n")
```

```
    model, doc_topics = cluster_lda(train_data, test_data, num_clusters = 3 ,\
                                    min_df = 5, stopwords = stopwords)
    doc_topics[10:15]
    print("\nQ.3b:\n")
    doc_id = 11
    idx = find_similar(doc_id, doc_topics)

    print(test_data.text.iloc[doc_id])
    print("Similar documents: \n")
    for i in idx:
        print(i,"-", test_data.iloc[i].text)
```

```
----Question 1----

With cosine distance:

label     T1   T2   T3
cluster
0         82   10  165
1        117    1    3
2         15  195   12
Cluster 0 -> Topic T3
top words: ['crash', 'bus', 'police', 'plane', 'rail', 'train', 'fire', 'passe
ngers', 'cruise', 'airlines']
Cluster 1 -> Topic T1
top words: ['oil', 'bp', 'people', 'japan', 'spill', 'water', 'gulf', 'disaste
r', 'nuclear', 'earthquake']
Cluster 2 -> Topic T2
top words: ['percent', 'tax', 'economy', 'rate', 'obama', 'comment', 'governme
nt', 'would', 'economic', 'billion']
              precision    recall  f1-score   support

          T1       0.97      0.55      0.70       214
          T2       0.88      0.95      0.91       206
          T3       0.64      0.92      0.76       180

    accuracy                           0.80       600
   macro avg       0.83      0.80      0.79       600
weighted avg       0.84      0.80      0.79       600


With Euclidean distance:

label     T1   T2   T3
cluster
0        134   59  179
1         12  147    1
2         68    0    0
Cluster 0 -> Topic T3
top words: ['crash', 'police', 'people', 'bus', 'fire', 'plane', 'rail', 'trai
n', 'passengers', 'new']
Cluster 1 -> Topic T2
top words: ['percent', 'tax', 'rate', 'obama', 'economy', 'comment', 'economi
c', 'government', 'would', 'billion']
Cluster 2 -> Topic T1
top words: ['oil', 'bp', 'japan', 'spill', 'gulf', 'people', 'pakistan', 'nucl
ear', 'tsunami', 'water']
              precision    recall  f1-score   support

          T1       1.00      0.32      0.48       214
          T2       0.92      0.71      0.80       206
          T3       0.48      0.99      0.65       180

    accuracy                           0.66       600
   macro avg       0.80      0.68      0.64       600
weighted avg       0.82      0.66      0.64       600



----Question 2----

label     T1   T2   T3
```

```
cluster
0          48  170    4
1          26   33  117
2         140    3   59
              precision    recall  f1-score   support

          T1       0.69      0.65      0.67       214
          T2       0.77      0.83      0.79       206
          T3       0.66      0.65      0.66       180

    accuracy                           0.71       600
   macro avg       0.71      0.71      0.71       600
weighted avg       0.71      0.71      0.71       600


----Question 3----

Q.3a:

label      T1   T2   T3
cluster
0          57   15  151
1         139    1   11
2          18  190   18
Topic 0: T3
top words:['comment', 'news', 'users', 'rail', 'travel', 'crash', 'passenger
s', 'service', 'car', 'plane']


Topic 1: T1
top words:['oil', 'japan', 'water', 'fire', 'officials', 'government', 'disast
er', 'city', 'nuclear', 'could']


Topic 2: T2
top words:['tax', 'government', 'obama', 'economy', 'billion', 'economic', 'mi
llion', 'market', 'money', 'president']


              precision    recall  f1-score   support

          T1       0.92      0.65      0.76       214
          T2       0.84      0.92      0.88       206
          T3       0.68      0.84      0.75       180

    accuracy                           0.80       600
   macro avg       0.81      0.80      0.80       600
weighted avg       0.82      0.80      0.80       600
```

Out[73]:

```
array([[0.37309574, 0.1535742 , 0.47333006],
       [0.01706364, 0.34609139, 0.63684497],
       [0.57436741, 0.0008928 , 0.42473979],
       [0.05198249, 0.77565863, 0.17235888],
       [0.73932828, 0.00158627, 0.25908545]])
```

Q.3b:

obama says he s finding out whose ass to kick over gulf disasterbyline time tu
e jun am et is president obama bowing to criticism that he hasn t shown enough
emotion and outrage about the gulf of mexico oil spill in an interview with th
e today show s matt lauer this morning the president offered his most candid r
esponse yet about the disaster bluntly telling lauer he s been talking to expe
rts about whose ass to kick when it comes to responsibility for the mess i was
down there a month ago before most of these talking heads were even paying att
ention to the gulf a month ago i was meeting with fishermen down there standin
g in the rain talking about what a potential crisis this could be obama said d
efending his administration s handling of the spill and i don t sit around jus
t talking to experts because this is a college seminar we talk to these folks
because they potentially have the best answers so i know whose ass to kick tha
t s a pretty sharp response for a president known for his cool headed approach
to situations in recent weeks as obama was assailed by critics for not being e
xpressive enough in his response to the spill white house officials defended h
is reaction by suggesting voters would prefer to see concrete actions over emp
ty method acting yet administration officials are not ignorant of polls showin
g the nation less than thrilled with obama s handling of the gulf according to
the latest abc washington post poll more than two thirds of those polled perce
nt disapprove of the federal government s handling of the spill that s higher
than the outrage over the bush administration s handling of hurricane katrina
holly bailey is a senior political writer for yahoo news

Similar documents:

169 - feds bp agrees to expedite oil spill payments the obama administration s
ays bp has agreed to expedite the payment of claims to businesses and individu
als whose livelihoods have been disrupted by the gulf of mexico oil spill trac
y wareing wehr ing who is with the national incident command office told repor
ters in washington that the understanding on payment of claims came in a meeti
ng wednesday with bp executives including ceo tony hayward wareing said admini
stration officials raised a pressing concern about the time bp has been taking
to provide relief payments particularly to businesses in the stricken area she
said the company will change the way it processes such claims and will expedit
e payments among other things it will drop the current practice of waiting to
make such payments until businesses have closed their books for each month

474 - feds bp agrees to expedite oil spill payments washington the obama admin
istration says bp has agreed to expedite the payment of claims to businesses a
nd individuals whose livelihoods have been disrupted by the gulf of mexico oil
spill tracy wareing wehr ing who is with the national incident command office
told reporters in washington that the understanding on payment of claims came
in a meeting wednesday with bp executives including ceo tony hayward wareing s
aid administration officials raised a pressing concern about the time bp has b
een taking to provide relief payments particularly to businesses in the strick
en area she said the company will change the way it processes such claims and
will expedite payments among other things it will drop the current practice of
waiting to make such payments until businesses have closed their books for eac
h month

167 - world bank waives haiti debt payments the world bank said thursday it wa
s waiving haiti s debt payments for the next five years due to the devastation
caused by the earthquake and is studying efforts to cancel the nation s remain
ing debt in a statement the washington based multilateral lender said haiti s
debt to the world bank which is interest free is about million dollars or arou
nd four percent of haiti s total external debt due to the crisis caused by the
earthquake we are waiving any payments on this debt for the next five years an
d at the same time we are working to find a way forward to cancel the remainin

```
g debt the statement said last week the world bank said it planned to provide
an additional million dollars in emergency aid to haiti after the january eart
hquake ravaged the impoverished nation officials fear up to people were killed
since the development lender said it has provided grants interest free aid of
million dollars to the caribbean nation the poorest country in the western hem
isphere that amount does not include the million dollars in grants announced o
n january it said the world bank and its sibling institution the international
monetary fund classify haiti among heavily indebted poor countries that are el
igible for debt forgiveness haiti was granted billion dollars in debt relief l
ast june
```

In [ ]: