Analisis Text

(Akmal Setiawan Wijaya), (Faishal Basbeth)

(20917005), (20917015)

1. Pada percobaan ini, di gunakanlah corpus tentang berita – berita, berita ini menggunakan Bahasa Inggris.

2. Proses pembuatan *Topic Modeling*, ada beberapa tahap, tahap yang pertama pastinya dibutuhkannya sebuah *library* yang akan digunakan pada percobaan ini, setelah itu import data tersebut ke dalam sistem yang digunakan untuk membuat model tersebut, disini digunakanlah *Jupyter*, lalu dilakukanlah *preprocessing* data, contoh pada *preprocesing* disini adalah pemberian token pada masing – masing kata. Lakukanlah Penentuan *Bow* dari dataset tersebut, setelah itu dilakukan proses TF-IDF, dan lakukanlah *training* model, lebih jelasnya bisa dilihat pada Gambar 1 – 7.

   - *Topic Modeling*

```
In [1]:
import pandas as pd
import gensim
from nltk.stem import WordNetLemmatizer, SnowballStemmer
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer,HashingVectorizer
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns
from nltk.stem.porter import *
import numpy as np
np.random.seed(2018)
import re
from sklearn.cluster import KMeans
```

Gambar 1. *Import library* yang dibutuhkan

```
In [3]:
# Load Data
data = pd.read_csv('abcnews-date-text.csv', error_bad_lines=False);
data_text = data[['headline_text']]
data_text['index'] = data_text.index
documents = data_text
```

Gambar 2. *Import* data dari csv

```
In [4]:
#  lemmatize dan stem preprocessing
stemmer = SnowballStemmer(language='english',ignore_stopwords=True)
def lemmatize_stemming(text):
    return stemmer.stem(WordNetLemmatizer().lemmatize(text, pos='v'))
def preprocess(text):
    result = []
    for token in gensim.utils.simple_preprocess(text):
        if token not in gensim.parsing.preprocessing.STOPWORDS and len(token) > 3:
            result.append(lemmatize_stemming(token))
    return result

In [5]:
# Preprocess  headline text, lalu simpan ke'processed_docs'
processed_docs = documents['headline_text'].map(preprocess)
processed_docs.head(10)

Out[5]: 0           [decid, communiti, broadcast, licenc]
        1                         [wit, awar, defam]
        2          [call, infrastructur, protect, summit]
        3                   [staff, aust, strike, rise]
        4            [strike, affect, australian, travel]
        5              [ambiti, olsson, win, tripl, jump]
        6          [antic, delight, record, break, barca]
        7    [aussi, qualifi, stosur, wast, memphi, match]
        8           [aust, address, secur, council, iraq]
        9                      [australia, lock, timet]
        Name: headline_text, dtype: object
```

Gambar 3. *Preprocesing* data

```
In [8]:
bow_corpus = [dictionary.doc2bow(doc) for doc in processed_docs]
bow_corpus[4310]

Out[8]: [(162, 1), (240, 1), (292, 1), (589, 1), (838, 1), (3570, 1), (3571, 1)]

In [9]:
# Preview Bag Of Words
bow_doc_4310 = bow_corpus[4310]
for i in range(len(bow_doc_4310)):
    print("Word {} (\"{}\") appears {} time.".format(bow_doc_4310[i][0],
                                                     dictionary[bow_doc_4310[i][0]],
bow_doc_4310[i][1]))

Word 162 ("govt") appears 1 time.
Word 240 ("group") appears 1 time.
Word 292 ("vote") appears 1 time.
Word 589 ("local") appears 1 time.
Word 838 ("want") appears 1 time.
Word 3570 ("compulsori") appears 1 time.
Word 3571 ("ratepay") appears 1 time.
```

Gambar 4. Menentukan *Bow* dari dataset

```
word 3571 ( ratepay ) appears 1 time.

In [10]:
tfidf = models.TfidfModel(bow_corpus)
corpus_tfidf = tfidf[bow_corpus]
for doc in corpus_tfidf:
    print(doc)
    break

[(0, 0.5842699484464488), (1, 0.38798859072167835), (2, 0.5008422243250992), (3, 0.5071987254965034)]
```

Gambar 5. mencari TF-IDF

```
In [11]:
# proses training
lda_model = gensim.models.LdaMulticore(bow_corpus, num_topics=10, id2word=dictionary, passes=2, workers=2)
```

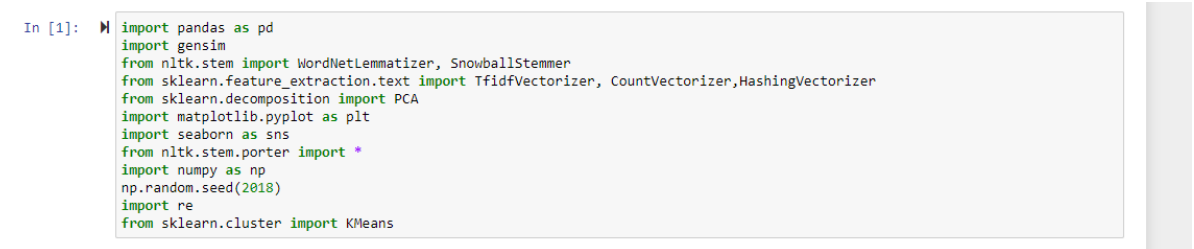
Gambar 6. Proses *Training*

Gambar 7. Hasil *topic modeling*

Pada gambar 7 dapat dilihat hasil dari topic modeling. Pada *topic* pertama dapat dilihat hasil nya polisi, duggaan, kematian, case, pembunuhan. Kemudan pada topic kedua dapat dilihat hasilnya pulau, australia, berita, pasar, dunia

3. Pada proses *Clustering* ini, seperti deskripsinya clustering adalah sebuah proses pengelompokan data dimana data tersebut tidak memiliki label, dari deskripsi yang di dapat, *Clustering* ini melakukan proses beberapa tahap yaitu seperti biasa di butuhkanlah *Import* masing – masing *library* yang di gunakan, dan juga upload data yang digunakan / yang akan di kelola, dan juga lakukanlah *preprocesing* data, perbedaan disini, disini ada proses perhitungan *vectorizer* daripada proses *Topic Modeling* di atas, lakukanlah *training*, dan *training* tersebut digunakanlah dengan algoritma *k-Means*, setelah itu didapatkanlah *plot clusternya*.

- *Clustering*



Gambar 8. *Import Library* yang dibutuhkan

```
In [5]:  # read data
         data = pd.read_csv('bbc-text.csv');
```

Gambar 9. *Load* data

```
In [3]:  #  lemmatize dan stem preprocessing
         stemmer = SnowballStemmer(language='english',ignore_stopwords=True)
         def lemmatize_stemming(text):
             return stemmer.stem(WordNetLemmatizer().lemmatize(text, pos='v'))
         def preprocess(text):
             result = []
             for token in gensim.utils.simple_preprocess(text):
                 if token not in gensim.parsing.preprocessing.STOPWORDS and len(token) > 3:
                     result.append(lemmatize_stemming(token))
             return ' '.join(result)

In [4]:  # Preprocess  headline text dan simpan ke 'processed_docs'
         processed_docs = data['text'].map(preprocess)
         processed_docs.head(10)

Out[4]: 0    futur hand viewer home theatr system plasma hi...
```

Gambar 10. *Preprocessing* data

```
In [5]:  # menghitung Vectors sebagai fitur
         # membuat dan menghitung vectorizer object
         count_vect = CountVectorizer(analyzer='word', token_pattern=r'\w{1,}')
         count_vect.fit(processed_docs)

         # transform  training dan validation data menggunakan  vectorizer object
         count_vec =  count_vect.transform(processed_docs)
```

Gambar 11. menghitung *vectorizer*

```
In [6]:  # parameter cluster K =3
         final_model=KMeans(3)
         final_model.fit(count_vec)
         prediction=final_model.predict(count_vec)
```

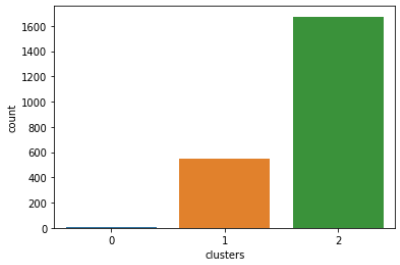
Gambar 12. *Training* dengan *Kmeans*

```
In [7]:  data["clusters"] = prediction

In [8]:  # plot total data pada cluster
         sns.countplot(data["clusters"])

         d:\phyton\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From
         version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword wil
         l result in an error or misinterpretation.
           FutureWarning

Out[8]: <AxesSubplot:xlabel='clusters', ylabel='count'>
```



Gambar 13. *Plot* data *cluster*

Dari plot pada gambar 13 dapat disimpulkan pada *cluster* 3 memiliki jumlah data paling banyak sebesar 1600.

```
In [9]:  data[data["clusters"] == 0].head()
```

Out[9]:

	category	text	clusters
408	politics	terror powers expose tyranny the lord chance...	0
865	entertainment	brits debate over urban music joss stone a ...	0
1604	politics	kilroy launches veritas party ex-bbc chat sh...	0
1615	entertainment	scissor sisters triumph at brits us band sciss...	0

```
In [10]:  data[data["clusters"] == 1].head()
```

Out[10]:

	category	text	clusters
0	tech	tv future in the hands of viewers with home th...	1
5	politics	howard hits back at mongrel jibe michael howar...	1
21	tech	halo 2 heralds traffic explosion the growing p...	1
24	tech	mobile audio enters new dimension as mobile ph...	1
28	politics	terror suspects face house arrest uk citizens ...	1

```
In [11]:  data[data["clusters"] == 2].head()
```

Out[11]:

	category	text	clusters
1	business	worldcom boss left books alone former worldc...	2
2	sport	tigers wary of farrell gamble leicester say ...	2
3	sport	yeading face newcastle in fa cup premiership s...	2
4	entertainment	ocean s twelve raids box office ocean s twelve...	2
6	politics	blair prepares to name poll date tony blair is...	2

```
In [ ]:
```

Gambar 14. Hasil *output* dari masing masing *cluster*

Dari gambar 14 dapat dilihat hasil dari masing masing *cluster*. Pada *cluster* pertama masuk ke katagori politik dan entertaiment. Kemudian pada *cluster* kedua ada katagori teknologi dan politik. Pada kluster terakhir terdapat 3 campuran katagori bisnis, olahraga dan politik.