Nama : Fuad Maulana
Email : fuadmaulana0812@gmail.com

Tugas Topic Modelling

- **Text Preprocessing**

  Sebelum melakukan modelling, perlu dilakukan pembersihan data sehingga analisis yang dilakukan dapat menjadi semakin lebih baik. Pembersihan yang dilakukan adalah case folding dan menghilang stopwords.

```
def case_folding(data):
    data = data.lower()
    data = ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)"," ",data).split())
    data = re.sub(r"\d+", "", data)
    data = data.translate(str.maketrans("","",string.punctuation))
    data = re.sub(r"\n","",data)
    data = re.sub(r"\t","",data)
    return data

def stopword_cleaner(data):
    sw_indonesia = stopwords.words("indonesian")
    data  = [word for word in data if word not in sw_indonesia]
    data = ' '.join(data)
    return data
```

```
[6] for index in range (len(df["judul"])):
      df["judul"].iloc[index] = case_folding(df["judul"].iloc[index])
      df["judul"].iloc[index] = word_tokenize(df["judul"].iloc[index])
      df["judul"].iloc[index] = stopword_cleaner(df["judul"].iloc[index])

    data_berita = [berita.split() for berita in df["judul"]]
```

- **Mencari jumlah topik optimum berdasarkan Coherence Value**

  Tahap pertama adalah membuat sebuah variabel yang isinya melakukan maping kata dalam satu dokumen terhadap semua kata pada dataset. hasilnya adalah nilai seperti matriks dengan nilai biner.

```
dictionary = corpora.Dictionary(data_berita)
doc_term_matrix = [dictionary.doc2bow(doc) for doc in data_berita]
```

  Kemudian membuat fungsi untuk melakukan hyper tuning parameter pada model yang akan dibuat untuk memperbaiki coherence value.

```
[9] def compute_coherence_values(corpus, dictionary, k, a, b):
        lda_model = gensim.models.LdaMulticore(corpus=doc_term_matrix,
                                               id2word=dictionary,
                                               num_topics=k,
                                               random_state=100,
                                               chunksize=100,
                                               passes=10,
                                               alpha=a,
                                               eta=b)
        coherence_model_lda = CoherenceModel(model=lda_model, texts=data_berita, dictionary=dictionary, coherence='c_v')
        return coherence_model_lda.get_coherence()
```

Kemudian dilakukan proses pencarian nilai koheren terhadap parameter yang dimiliki dengan mencari nilai k, alpha, dan beta terbaik.

```
# Topics range
min_topics = 2
max_topics = 11
step_size = 1
topics_range = range(min_topics, max_topics, step_size)
# Alpha parameter
alpha = list(np.arange(0.01, 1, 0.3))
alpha.append('symmetric')
alpha.append('asymmetric')
# Beta parameter
beta = list(np.arange(0.01, 1, 0.3))
beta.append('symmetric')
# Validation sets
num_of_docs = len(doc_term_matrix)
corpus_sets = [gensim.utils.ClippedCorpus(doc_term_matrix, num_of_docs*0.75),
               doc_term_matrix]
corpus_title = ['75% Corpus', '100% Corpus']
model_results = {'Validation_Set': [],
                 'Topics': [],
                 'Alpha': [],
                 'Beta': [],
                 'Coherence': []
                }
```

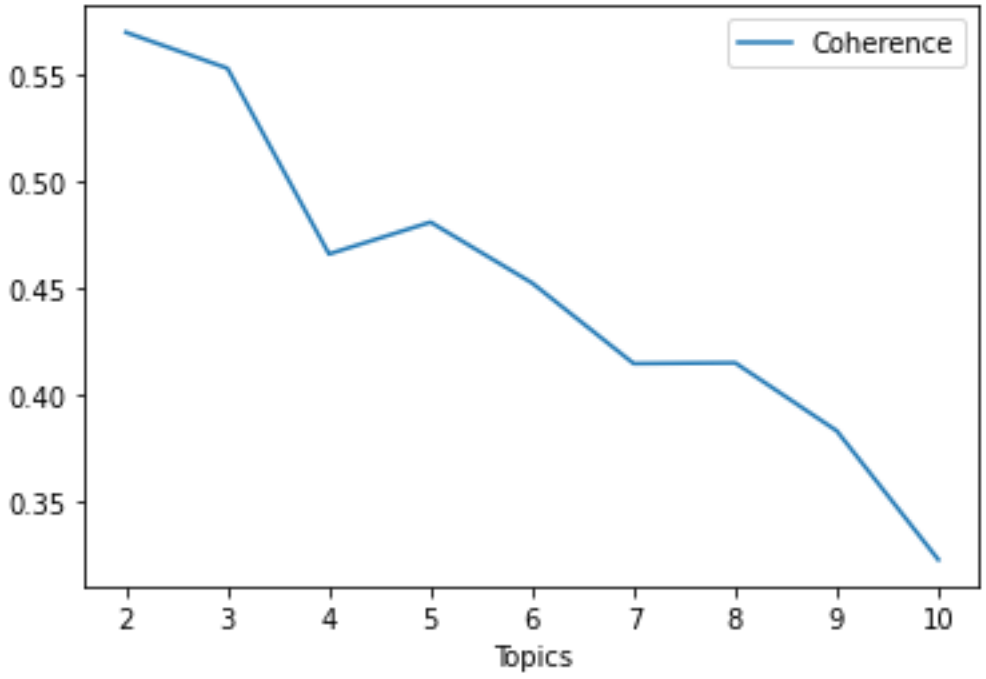
```
if 1 == 1:
    pbar = tqdm.tqdm(total=540)
    for i in range(len(corpus_sets)):
        for k in topics_range:
            for a in alpha:
                for b in beta:
                    cv = compute_coherence_values(corpus=corpus_sets[i], dictionary=dictionary,
                                                  k=k, a=a, b=b)
                    # Save the model results
                    model_results['Validation_Set'].append(corpus_title[i])
                    model_results['Topics'].append(k)
                    model_results['Alpha'].append(a)
                    model_results['Beta'].append(b)
                    model_results['Coherence'].append(cv)
                    pbar.update(1)
    pd.DataFrame(model_results).to_csv('lda_tuning_results.csv', index=False)
    pbar.close()
```

```
df_tuning_result = pd.read_csv("lda_tuning_results.csv")
df_tuning_result
```

	Validation_Set	Topics	Alpha	Beta	Coherence
0	75% Corpus	2	0.01	0.01	0.563620
1	75% Corpus	2	0.01	0.31	0.569274
2	75% Corpus	2	0.01	0.61	0.562670
3	75% Corpus	2	0.01	0.9099999999999999	0.537393
4	75% Corpus	2	0.01	symmetric	0.561362
...	...	...	...	...	...
535	100% Corpus	10	asymmetric	0.01	0.328445
536	100% Corpus	10	asymmetric	0.31	0.301677
537	100% Corpus	10	asymmetric	0.61	0.304176
538	100% Corpus	10	asymmetric	0.9099999999999999	0.297632

Selanjutnya untuk melihat k terbaik, saya akan memilih salah satu nilai alpha dan beta yang ada pada table tersebut yaitu alpha = 0.01 dan beta = 0.31. Kemudian mencari nilai k dengan menggunakan plot terhadap hasilnya

```
[12] _plot = df_tuning_result[(df_tuning_result.Alpha == "0.01") & (df_tuning_result.Beta == "0.31") & (df_tuning_result.Validation_Set == "100% Corpus")].copy()
     _plot.plot(x='Topics', y = "Coherence", kind="line")
```



Sehingga dapat disimpulkan bahwa jumlah topik optimum adalah sebanyak 2 topik.

- **Penjelasan tentang topik optimum yang didapatkan**

  Selanjutnya mencari topik apa saja yang merupakan topik optimum.

```
[14] # Berdasarkan plot, maka topik optimum sebanyak 2 topik
     df_plot_2 = df_tuning_result[(df_tuning_result.Topics == 2) & (df_tuning_result.Validation_Set == "100% Corpus")].copy()
     df_plot_2.sort_values(by="Coherence", ascending=False)
```

	Validation_Set	Topics	Alpha	Beta	Coherence
278	100% Corpus	2	0.31	0.9099999999999999	0.577405
271	100% Corpus	2	0.01	0.31	0.569274
279	100% Corpus	2	0.31	symmetric	0.568200
276	100% Corpus	2	0.31	0.31	0.567383
270	100% Corpus	2	0.01	0.01	0.563620
272	100% Corpus	2	0.01	0.61	0.562670
274	100% Corpus	2	0.01	symmetric	0.561362
277	100% Corpus	2	0.31	0.61	0.557231
273	100% Corpus	2	0.01	0.9099999999999999	0.537393
289	100% Corpus	2	0.9099999999999999	symmetric	0.526101

Dapat diketahui bahwa coherence value terbaik ada pada saat alpha = 0.31 dan beta = 0.9. Setelah mendapatkan parameter terbaik, maka akan dibuat modelnya

```
[17] # Coherence value terbaik ada saat alpha 0.31 dan beta 0.9
     lda_model = gensim.models.LdaMulticore(corpus=doc_term_matrix,
                                            id2word=dictionary,
                                            num_topics=2,
                                            random_state=100,
                                            chunksize=100,
                                            passes=10,
                                            alpha=0.31,
                                            eta=0.9)
```

Dari model tersebut, maka topik yang didapatkan adalah sebagai berikut

```
[18] pprint(lda_model.print_topics())

     [(0,
       '0.010*"kpk" + 0.008*"ott" + 0.006*"vaksinasi" + 0.006*"bnn" + '
       '0.006*"korupsi" + 0.005*"ajukan" + 0.005*"hakim" + 0.005*"surabaya" + '
       '0.004*"juta" + 0.004*"pelaku"'),
      (1,
       '0.016*"arteria" + 0.012*"covid" + 0.010*"dahlan" + 0.010*"pdi" + 0.010*"p" '
       '+ 0.010*"sunda" + 0.008*"polisi" + 0.007*"dki" + 0.006*"fraksi" + '
       '0.006*"jakarta"')]
```

Analisis terhadap setiap topik adalah :

Topik 0 – kata kpk, vaksinasi, korupsi, hakim, surabaya, pelaku maka dapat disimpulkan bahwa topik ini menjelaskan tentang **pelaku korupsi vaksinasi di Surabaya**

Topik 1 – kata arteria, dahlan, pdi, p, sunda, polisi, dki, fraksi, jakarta maka dapat disimpulkan bahwa topik ini menjelaskan tentang **kasus Arteria Dahlan seorang dari fraksi PDIP tentang penggunaan bahasa sunda di DKI Jakarta**