Predictions on Topics of UN Debates at UN Assembly New York with Topic Modelling

- Charita Ramnathsing

- April 29th, 2020

Build Dictionary

```
# Create a dictionary representation of the documents
dictionary = gensim.corpora.Dictionary(data_corpus_bigram)
```

```
dictionary.filter_extremes(no_below=10, no_above=0.6)
```

```
#Transform corpus into bag of words vectors
bow_corpus = [dictionary.doc2bow(text) for text in data_corpus_bigram]
```

Initialize and run LDA Mallet

```
from gensim.test.utils import common_corpus, common_dictionary
from gensim.models.wrappers import LdaMallet
from gensim.models import CoherenceModel
import os
```

```
os.environ.update({'MALLET_HOME':r'my_folder/mallet-2.0.8/bin'})
MALLET_PATH = 'my_folder/mallet-2.0.8/bin/mallet'
```

```
lda_mallet = gensim.models.wrappers.LdaMallet(mallet_path=MALLET_PATH, corpus=bow_corpus, id2word=dictionary,
                                              num_topics=8,
                                              iterations=500, workers=4)
```

Topic Modelling with ’topic_nums’ topics

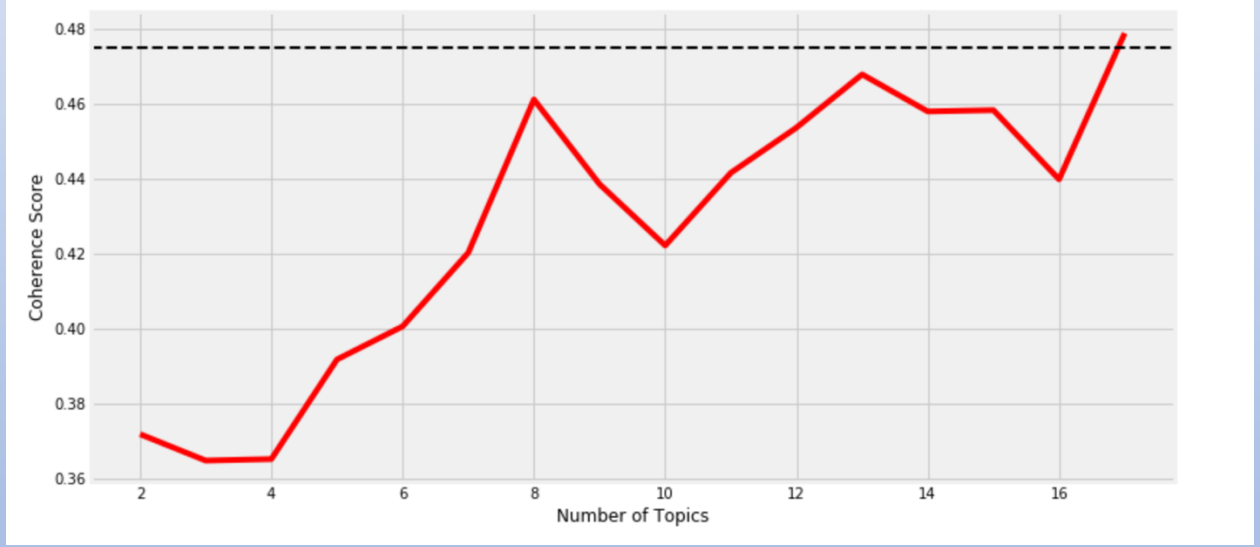
```
# Finding the optimal numbers of topics
def topic_model_coherence_generator(corpus, texts, dictionary,
                                    start_topic_count=2, end_topic_count=17, step=1,
                                    cpus=1):

    models = []
    coherence_scores = []
    for topic_nums in tqdm.tqdm(range(start_topic_count, end_topic_count+1, step)):
        mallet_lda_model = gensim.models.wrappers.LdaMallet(mallet_path=MALLET_PATH, corpus=corpus,
                                                            num_topics=topic_nums, id2word=dictionary,
                                                            iterations=150, workers=cpus)
        cv_coherence_model_mallet_lda = gensim.models.CoherenceModel(model=mallet_lda_model, corpus=corpus,
                                                                     texts=texts, dictionary=dictionary,
                                                                     coherence='c_v')
        coherence_score = cv_coherence_model_mallet_lda.get_coherence()
        coherence_scores.append(coherence_score)
        models.append(mallet_lda_model)

    return models, coherence_scores
```

```
lda_models, coherence_scores = topic_model_coherence_generator(corpus=bow_corpus, texts=data_corpus_bigram,
                                                               dictionary=dictionary , start_topic_count=2,
                                                               end_topic_count=17, step=1, cpus=4)
```

Chosen number of Topics in the Model



How did the model perform?

```
cv_coherence_model_lda_mallet = gensim.models.CoherenceModel(model=lda_mallet, corpus=bow_corpus,
                                                             texts=data_corpus_bigram,
                                                             dictionary=dictionary,
                                                             coherence='c_v')
avg_coherence_cv = cv_coherence_model_lda_mallet.get_coherence()

umass_coherence_model_lda_mallet = gensim.models.CoherenceModel(model=lda_mallet, corpus=bow_corpus,
                                                                texts=data_corpus_bigram,
                                                                dictionary=dictionary,
                                                                coherence='u_mass')
avg_coherence_umass = umass_coherence_model_lda_mallet.get_coherence()


print('Avg. Coherence Score (Cv):', avg_coherence_cv)
print('Avg. Coherence Score (UMass):', avg_coherence_umass)
```

```
Avg. Coherence Score (Cv): 0.47228869280709124
Avg. Coherence Score (UMass): -0.9522068656727839
```

Feature Engineering after Modelling

```
topics = [[(term, round(wt,3))
               for term, wt in lda_mallet.show_topic(n, topn=10)]
                     for n in range(0, lda_mallet.num_topics)]
```

```
tm_results = lda_mallet[bow_corpus]
```

```
corpus_topics = [sorted(topics_local, key=lambda record: -record[1])[0]
                        for topics_local in tm_results]
```

```
topics_df = pd.DataFrame([','.join([term for term, wt in topic])
                              for topic in topics],
                         columns = ['Terms per Topic'],
                         index=['Topic'+str(t) for t in range(1, lda_mallet.num_topics+1)])
```

```
corpus_topic_df = pd.DataFrame()
corpus_topic_df['Document'] = range(0, len(df.text))
corpus_topic_df['Dominant Topic'] = [item[0]+1 for item in corpus_topics]
corpus_topic_df['Contribution %'] = [round(item[1]*100, 2) for item in corpus_topics]
corpus_topic_df['Topic Description'] = [topics_df.iloc[t[0]]['Terms per Topic'] for t in corpus_topics]
corpus_topic_df['Country'] = df.country
```

```
pd.set_option('display.max_colwidth', 250)
corpus_topic_df.groupby('Dominant Topic').apply(lambda topic_set: (topic_set.sort_values(by=['Contribution %'],
                                                                                         ascending=False)
                                                                   .iloc[0]))
```

Prediction of Topic After modelling

Document	Dominant Topic	Contribution %	Topic Description	Country
0	6	30.28	south_africa,delegation,african,co_operation,namibia,independence,south,africa,apartheid,developed	Maldives
1	8	47.47	disarmament,nuclear,europe,nuclear_weapon,proposal,european,co_operation,treaty,field,basis	Finland
2	6	31.14	south_africa,delegation,african,co_operation,namibia,independence,south,africa,apartheid,developed	Niger
3	3	41.61	law,solidarity,society,trade,justice,mankind,latin_america,reason,case,freedom	Uruguay
4	6	33.52	south_africa,delegation,african,co_operation,namibia,independence,south,africa,apartheid,developed	Zimbabwe
...	...	...	...	...
7502	8	29.42	disarmament,nuclear,europe,nuclear_weapon,proposal,european,co_operation,treaty,field,basis	Kazakhstan
7503	5	25.45	africa,democracy,african,republic,democratic,election,solidarity,continent,child,programme	Liberia
7504	5	38.48	africa,democracy,african,republic,democratic,election,solidarity,continent,child,programme	Burundi
7505	1	30.83	terrorism,iraq,palestinian,law,afghanistan,israel,comprehensive,council,humanitarian,party	Hungary
7506	1	45.83	terrorism,iraq,palestinian,law,afghanistan,israel,comprehensive,council,humanitarian,party	Kuwait

7507 rows × 4 columns

Bag of Words per Topic

Word use per Topic