LDA Tuning Approach 3

Document = part between interludes

Imports

In [1]:
```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from wordcloud import WordCloud, STOPWORDS
from gensim.utils import simple_preprocess,SaveLoad
#nltk.download('stopwords')
#nltk.download('wordnet')
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer, SnowballStemmer
from nltk.stem.porter import *
import time
from pickle import load, dump
import my_lda_utils
from my_lda_utils import my_lda_model
from gensim.models import CoherenceModel
import pyLDAvis
```

Initializations

In [2]:
```python
np.random.seed(2018)
stemmer = SnowballStemmer('english')
pyLDAvis.enable_notebook()
```

In [3]:
```python
stop_words = stopwords.words('english')
stop_words.extend(STOPWORDS)
stop_words2 = stop_words
```

Helpers

In [4]:
```python
# Cleaning functions
def remove_stopwords(texts):
    return [word for word in texts if word not in stop_words ]

def stem_and_lemmatize(texts):
    return [stemmer.stem(WordNetLemmatizer().lemmatize(word, pos='v')) for word in texts] #try to lematize first: source https://towardsdatascience.com/tf-idf-for-document-ranking-from-s

#     return [[word for word in simple_preprocess(str(doc))
#              if word not in stop_words] for doc in texts]
```

Import Data

In [5]:
```python
clean_data_folder = "../clean_data/"
with open("../clean_data/"+"characters_with_gender.pckl", "rb") as f2:
    characters = load(f2)
```

In [6]:
```python
clean_data_folder_lda = "../clean_data_lda"
with open(clean_data_folder_lda+"/all_data.pkl", "rb") as f:
    all_scripts = load(f)
data_lda = pd.merge(all_scripts, characters, on="character", how="inner")
```

In [7]:
```python
all_scripts.reset_index(inplace=True, drop=True)
```

In [8]:
```python
data = pd.merge(all_scripts, characters, on="character", how="inner")
```

In [9]:
```python
data2 = data_lda[['text', 'doc_id']]
data3 = data2.groupby(['doc_id'])['text'].agg(sum).reset_index()
documents = data3.text.apply(stem_and_lemmatize)
```

Tuning

Number of Topics

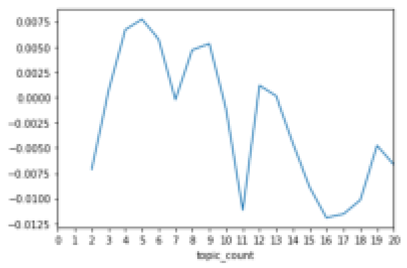
In [10]:
```python
coherence_scores_nmpi = []
```

In [11]:
```python
for i in range(2,21,1):
    lda_model_x = my_lda_model(documents)
    lda_model_x.train(num_topics = i, passes = 5, tfidf_threshold = 0.0)
    coherence_model_lda = CoherenceModel(model=lda_model_x.lda_model, texts=lda_model_x.documents, dictionary=lda_model_x.dictionary, coherence='c_npmi')
    coherence_lda = coherence_model_lda.get_coherence()
    coherence_scores_nmpi.append([i, coherence_lda,lda_model_x])
    print("coherence: ",[i, coherence_lda])
```

...

In [10]:
```python
#df = pd.DataFrame(coherence_scores_nmpi)
#df.columns = ["topic_count", "coherence", "model"]
#df = df.set_index("topic_count")
df.coherence.plot(kind='line',
                  xticks=[x for x in range(21)],
                 xlim=[0,20])
df.coherence.idxmax()
```

Out[10]:
```
5
```



In [13]:
```python
df.coherence[9:].idxmax()
```

Out[13]:
```
12
```

In [14]:
```python
#df.to_pickle("c_nmpi_coherence_approach3.pkl")
```

In [7]:
```python
with open("c_nmpi_coherence_approach3.pkl", "rb") as f:
    coherence_scores_topic_counts = load(f)
```

In [8]:
```python
coherence_scores_topic_counts.head(2)
```

Out[8]:

topic_count	coherence	model
2	-0.007080	<my_lda_utils.my_lda_model object at 0x000001C...
3	0.000657	<my_lda_utils.my_lda_model object at 0x000001C...

In [9]:
```python
df = coherence_scores_topic_counts
```

5, 9, 12 maxima

In [ ]:
```python
df.model[4].visual
```

In [ ]:
```python
df.model[7].visual
```

In [ ]:
```python
df.model[12].visual
```

pick topic count 8 for better comparisoin

TF-IDF - Theshold

8 Topics

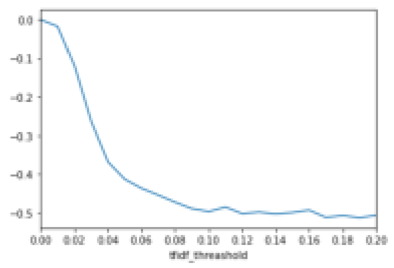
In [17]:
```python
coherence_scores_tfidf = []
```

In [31]:
```python
tfidf_range = [round(x*0.01, 2) for x in range(21)]
for i in tfidf_range:
    lda_model_x = my_lda_model(documents)
    lda_model_x.train(num_topics = 8, passes = 5, tfidf_threshold = i)
    coherence_model_lda = CoherenceModel(model=lda_model_x.lda_model, texts=lda_model_x.documents, dictionary=lda_model_x.dictionary, coherence='c_npmi')
    coherence_lda = coherence_model_lda.get_coherence()
    coherence_scores_tfidf.append([i, coherence_lda,lda_model_x])
    print("coherence: ",[i, coherence_lda])
```

...

In [40]:
```python
df = pd.DataFrame(coherence_scores_tfidf)
df.columns = ["tfidf_threashold", "coherence", "model"]
df = df.set_index("tfidf_threashold")
df.coherence.plot(kind='line',
                  xticks=[i for i in tfidf_range if round(i*100)%2 == 0], # get every second tick
                 xlim=[0,0.2])
df.coherence.idxmax()
```

Out[40]:
```
0.0
```



In [41]:
```python
#df.to_pickle("tfidf_coherence-topics-8-passes-5_approach3.pkl")
#todo: compare to approach 2
```

In [99]:
```python
with open("tfidf_coherence-topics-8-passes-5_approach3.pkl", "rb") as f:
    df = load(f)
df.head(2)
```

Out[99]:

tfidf_threashold	coherence	model
0.00	-0.000705	<my_lda_utils.my_lda_model object at 0x0000028...
0.01	-0.017902	<my_lda_utils.my_lda_model object at 0x0000028...

In [108]:
```python
df.model[0.11].visual
```

Out[108]:

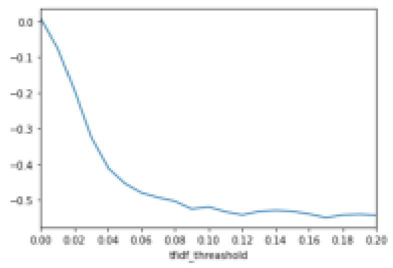
4 Topics

In [46]:
```python
coherence_scores_tfidf = []
```

In [47]:
```python
tfidf_range = [round(x*0.01, 2) for x in range(21)]
for i in tfidf_range:
    lda_model_x = my_lda_model(documents)
    lda_model_x.train(num_topics = 4, passes = 5, tfidf_threshold = i)
    coherence_model_lda = CoherenceModel(model=lda_model_x.lda_model, texts=lda_model_x.documents, dictionary=lda_model_x.dictionary, coherence='c_npmi')
    coherence_lda = coherence_model_lda.get_coherence()
    coherence_scores_tfidf.append([i, coherence_lda,lda_model_x])
    print("coherence: ",[i, coherence_lda])
```

...

In [48]:
```python
df = pd.DataFrame(coherence_scores_tfidf)
df.columns = ["tfidf_threashold", "coherence", "model"]
df = df.set_index("tfidf_threashold")
df.coherence.plot(kind='line',
                  xticks=[i for i in tfidf_range if round(i*100)%2 == 0], # get every second tick
                 xlim=[0,0.2])
df.coherence.idxmax()
```
Out[48]: 0.0



In [49]:
```python
#df.to_pickle("tfidf_coherence-topics-4-passes-5_approach3.pkl")
```

In [8]:
```python
with open("tfidf_coherence-topics-4-passes-5_approach3.pkl", "rb") as f:
    df2 = load(f)
df2.head(2)
```
Out[8]:

	coherence	model
tfidf_threashold		
0.00	0.007221	<my_lda_utils.my_lda_model object at 0x000001E...
0.01	-0.074569	<my_lda_utils.my_lda_model object at 0x000001E...

In [49]:
```python
df2.model[0.2].visual
```
Out[49]:

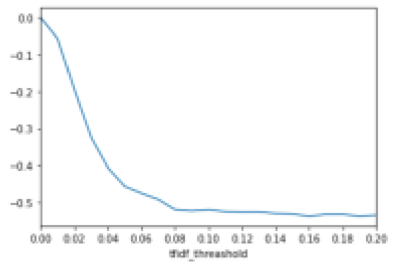
5 Topics

In [26]:
```python
coherence_scores_tfidf = []
```

In [27]:
```python
tfidf_range = [round(x*0.01, 2) for x in range(21)]
for i in tfidf_range:
    lda_model_x = my_lda_model(documents)
    lda_model_x.train(num_topics = 5, passes = 5, tfidf_threshold = i)
    coherence_model_lda = CoherenceModel(model=lda_model_x.lda_model, texts=lda_model_x.documents, dictionary=lda_model_x.dictionary, coherence='c_npmi')
    coherence_lda = coherence_model_lda.get_coherence()
    coherence_scores_tfidf.append([i, coherence_lda,lda_model_x])
    print("coherence: ",[i, coherence_lda])
```

```
C:\Users\debor\anaconda3\lib\site-packages\pyLDAvis\_prepare.py:246: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be
keyword-only.
  default_term_info = default_term_info.sort_values(

coherence:  [0.0, -0.0005813933019606164]

C:\Users\debor\anaconda3\lib\site-packages\pyLDAvis\_prepare.py:246: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be
keyword-only.
  default_term_info = default_term_info.sort_values(

coherence:  [0.01, -0.05626118088835745]

C:\Users\debor\anaconda3\lib\site-packages\pyLDAvis\_prepare.py:246: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be
keyword-only.
  default_term_info = default_term_info.sort_values(

coherence:  [0.02, -0.19691455879759376]

C:\Users\debor\anaconda3\lib\site-packages\pyLDAvis\_prepare.py:246: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be
keyword-only.
  default_term_info = default_term_info.sort_values(
```

In [28]:
```python
df = pd.DataFrame(coherence_scores_tfidf)
df.columns = ["tfidf_threashold", "coherence", "model"]
df = df.set_index("tfidf_threashold")
df.coherence.plot(kind='line',
                  xticks=[i for i in tfidf_range if round(i*100)%2 == 0], # get every second tick
                 xlim=[0,0.2])
df.coherence.idxmax()
```
Out[28]: 0.0



In [29]:
```python
#df.to_pickle("tfidf_coherence-topics-5-passes-5_approach3.pkl")
```

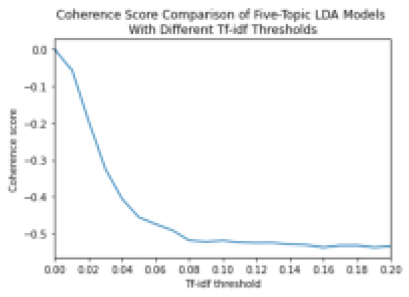
In [10]:
```python
with open("tfidf_coherence-topics-5-passes-5_approach3.pkl", "rb") as f:
    df2 = load(f)
df2.head(2)
```

Out[10]:

	coherence	model
tfidf_threashold		
0.00	-0.000581	<my_lda_utils.my_lda_model object at 0x0000025...
0.01	-0.056261	<my_lda_utils.my_lda_model object at 0x0000025...

In [22]:
```python
df2.columns = ["Coherence", "Model"]
df2.index.names = ["Tf-idf threshold"]
df2.Coherence.plot(kind='line',
                   xticks=[round(x*0.01, 2) for x in range(0, 21, 2)], # get every second tick
                   xlim=[0,0.2],
                   ylabel="Coherence score",
                   title="Coherence Score Comparison of Five-Topic LDA Models \n With Different Tf-idf Thresholds "
                  )
```

Out[22]:
```
<AxesSubplot:title={'center':'Coherence Score Comparison of Five-Topic LDA Models \n With Different Tf-idf Thresholds '}, xlabel='Tf-idf threshold', ylabel='Coherence score'>
```



In [19]:
```python

```

Out[19]:
```
[0.0, 0.02, 0.04, 0.06, 0.08, 0.1, 0.12, 0.14, 0.16, 0.18, 0.2]
```

In [16]:
```python
df2
```

Out[16]:

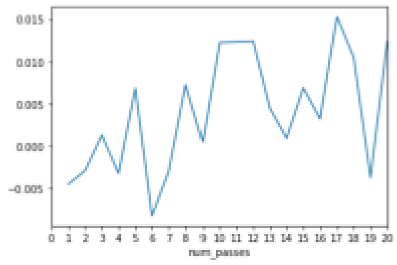
	Coherence	Model
Tf-idf threshold		
0.00	-0.000581	<my_lda_utils.my_lda_model object at 0x0000025...
0.01	-0.056261	<my_lda_utils.my_lda_model object at 0x0000025...
0.02	-0.196915	<my_lda_utils.my_lda_model object at 0x0000025...
0.03	-0.324870	<my_lda_utils.my_lda_model object at 0x0000025...
0.04	-0.406348	<my_lda_utils.my_lda_model object at 0x0000025...
0.05	-0.455895	<my_lda_utils.my_lda_model object at 0x0000025...
0.06	-0.474568	<my_lda_utils.my_lda_model object at 0x0000025...
0.07	-0.491303	<my_lda_utils.my_lda_model object at 0x0000025...
0.08	-0.518814	<my_lda_utils.my_lda_model object at 0x0000025...
0.09	-0.522374	<my_lda_utils.my_lda_model object at 0x0000025...
0.10	-0.519249	<my_lda_utils.my_lda_model object at 0x0000025...
0.11	-0.524537	<my_lda_utils.my_lda_model object at 0x0000025...
0.12	-0.525298	<my_lda_utils.my_lda_model object at 0x0000025...
0.13	-0.525205	<my_lda_utils.my_lda_model object at 0x0000025...
0.14	-0.529430	<my_lda_utils.my_lda_model object at 0x0000025...
0.15	-0.530739	<my_lda_utils.my_lda_model object at 0x0000025...
0.16	-0.537099	<my_lda_utils.my_lda_model object at 0x0000025...
0.17	-0.531854	<my_lda_utils.my_lda_model object at 0x0000025...
0.18	-0.531787	<my_lda_utils.my_lda_model object at 0x0000025...
0.19	-0.537047	<my_lda_utils.my_lda_model object at 0x0000025...
0.20	-0.534179	<my_lda_utils.my_lda_model object at 0x0000025...

Number of Passes

In [40]:
```python
coherence_scores_no_passes = []
```

In [41]:
```python
lds_train = df2.model[0.01].lda_model
for i in range(1,21,1):
    lda_model_x = my_lda_model(documents)
    lda_model_x.train(num_topics = 8, passes = i, tfidf_threshold = 0)#todo
    coherence_model_lda = CoherenceModel(model=lda_model_x.lda_model, texts=lda_model_x.documents, dictionary=lda_model_x.dictionary, coherence='c_npmi')
    coherence_lda = coherence_model_lda.get_coherence()
    coherence_scores_no_passes.append([i, coherence_lda,lda_model_x])
    print("coherence: ",[i, coherence_lda])
```

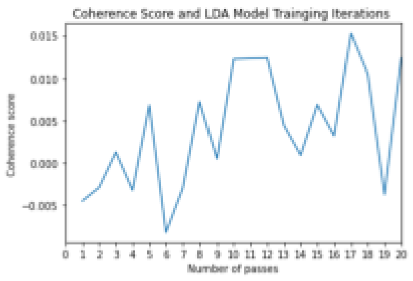
...

In [43]:
```python
df = pd.DataFrame(coherence_scores_no_passes)
df.columns = ["num_passes", "coherence", "model"]
df = df.set_index("num_passes")
df.coherence.plot(kind='line',
                  xticks=list(range(21)),
                 xlim=[0,20])
df.coherence.idxmax()
```
Out[43]:
```
17
```


In [44]:
```python
#df.to_pickle("num_passes_coherence_approach3-topic8-tfidf-0.pkl")
```

In [12]:
```python
with open("num_passes_coherence_approach3-topic8-tfidf-0.pkl", "rb") as f:
    df = load(f)
df.head(2)
```
Out[12]:

	coherence	model
num_passes		
1	-0.004455	<my_lda_utils.my_lda_model object at 0x000001B...
2	-0.002866	<my_lda_utils.my_lda_model object at 0x000001B...

In [14]:
```python
# df = pd.DataFrame(coherence_scores_no_passes)
# df.columns = ["num_passes", "coherence", "model"]
# df = df.set_index("num_passes")
df.coherence.plot(kind='line',
                  xticks=list(range(21)),
                 xlim=[0,20],
                 xlabel="Number of passes",
                 ylabel="Coherence score",
                 title="Coherence Score and LDA Model Trainging Iterations ")
```
Out[14]:
```
<AxesSubplot:title={'center':'Coherence Score and LDA Model Trainging Iterations '}, xlabel='Number of passes', ylabel='Coherence score'>
```


In [ ]:
```python

```