

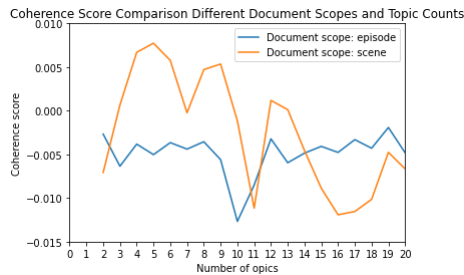
## Coherences Scores of LDA With Different Documents Scopes and NMF

```
In [1]: 1 from pickle import load#, dump
2 from my_lda_utils import my_lda_model
3 import pandas as pd
```

### Number of Topics and Coherence LDA Document Scope Episode and Scene

```
In [65]: 1 with open("c_nmpi_coherence_approach3.pkl", "rb") as f1, open("c_nmpi_coherence_approach2.pkl", "rb") as f2:
2 coherence_topic_counts_approach3 = load(f1)
3 coherence_topic_counts_approach2 = load(f2)
4
5 compare_coherence = pd.merge(coherence_topic_counts_approach2['coherence'],coherence_topic_counts_approach3['coherence'],
6                             suffixes = ('_2', '_3'),right_index=True, left_index=True)
7
8 compare_coherence.columns = ['Document scope: episode', 'Document scope: scene']
9 compare_coherence.plot(kind='line',
10                        xticks=[x for x in range(21)],
11                        yticks=[round(x*0.001, 5) for x in range(-15,15,5)],
12                        xlim=[0,20],
13                        xlabel="Number of opics",
14                        ylabel="Coherence score",
15                        title="Coherence Score Comparison Different Document Scopes and Topic Counts"
16                        )
```

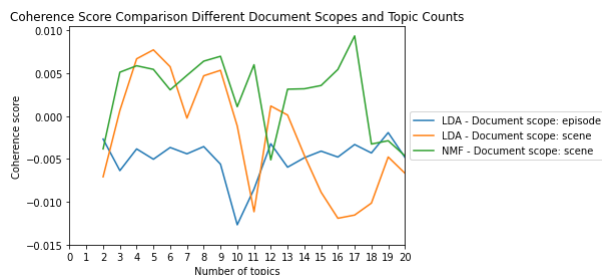
```
Out[65]: <AxesSubplot:title={'center':'Coherence Score Comparison Different Document Scopes and Topic Counts'}, xlabel='Number of opics', ylabel='Coherence score'>
```



### Number of Topics and General Coherence LDA Document Scope Episode and Scene and NMF Document Scope Scene

```
In [68]: 1 with open("c_nmpi_coherence_approach3.pkl", "rb") as f1, \
2         open("c_nmpi_coherence_approach2.pkl", "rb") as f2, \
3         open("c_nmpi_coherence_approach3_nmf.pkl", "rb") as f3:
4 coherence_topic_counts_approach3 = load(f1)
5 coherence_topic_counts_approach2 = load(f2)
6 coherence_topic_counts_approach3_nmf = load(f3)
7
8
9 compare_coherence1 = pd.merge(coherence_topic_counts_approach2['coherence'],coherence_topic_counts_approach3['coherence'],
10                             suffixes = ('_2', '_3'),right_index=True, left_index=True)
11 compare_coherence = pd.merge(compare_coherence1,coherence_topic_counts_approach3_nmf['coherence'], suffixes = ('', ''),
12                             right_index=True, left_index=True)
13 compare_coherence.columns = ['LDA - Document scope: episode', 'LDA - Document scope: scene', 'NMF - Document scope: scene']
14
15 compare_coherence.plot(kind='line',
16                        xticks=[x for x in range(21)],
17                        yticks=[round(x*0.001, 5) for x in range(-15,15,5)],
18                        xlim=[0,20],
19                        xlabel="Number of topics",
20                        ylabel="Coherence score",
21                        title="Coherence Score Comparison Different Document Scopes and Topic Counts",
22                        ).legend(loc='center left',bbox_to_anchor=(1.0, 0.5))
```

```
Out[68]: <matplotlib.legend.Legend at 0x20983c137f0>
```

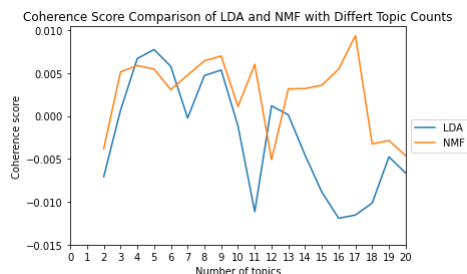


```

In [71]: 1 with open("c_nmpi_coherence_approach3.pkl", "rb") as f1, \
2         open("c_nmpi_coherence_approach3_nmf.pkl", "rb") as f3:
3         coherence_topic_counts_approach3 = load(f1)
4         coherence_topic_counts_approach3_nmf = load(f3)
5
6
7 compare_coherence = pd.merge(coherence_topic_counts_approach3['coherence'],
8                             coherence_topic_counts_approach3_nmf['coherence'],
9                             suffixes = ('_LDA', '_NMF'),right_index=True, left_index=True)
10 compare_coherence.columns = ['LDA', 'NMF']
11
12 compare_coherence.plot(kind='line',
13                        xticks=[x for x in range(21)],
14                        yticks=[round(x*0.001, 5) for x in range(-15,15)],
15                        xlim=[0,20],
16                        xlabel="Number of topics",
17                        ylabel="Coherence score",
18                        title="Coherence Score Comparison of LDA and NMF with Differt Topic Counts",
19                        ).legend(loc='center left',bbox to anchor=(1.0, 0.5))

```

Out[71]: <matplotlib.legend.Legend at 0x2098a09dbe0>



```

In [76]: 1 with open("c_nmpi_coherence_approach3.pkl", "rb") as f1, \
2         open("c_nmpi_coherence_approach2.pkl", "rb") as f2, \
3         open("c_nmpi_coherence_approach3_nmf.pkl", "rb") as f3:
4         coherence_topic_counts_approach3 = load(f1)
5         coherence_topic_counts_approach2 = load(f2)
6         coherence_topic_counts_approach3_nmf = load(f3)
7
8
9 compare_coherence1 = pd.merge(coherence_topic_counts_approach2['coherence'],
10                             coherence_topic_counts_approach3['coherence'],
11                             suffixes = ('_2', '_3'),right_index=True, left_index=True)
12 compare_coherence = pd.merge(compare_coherence1,coherence_topic_counts_approach3_nmf['coherence'],
13                             suffixes = ('', ''),right_index=True, left_index=True)
14 compare_coherence.columns = ['del', 'LDA', 'NMF']
15 compare_coherence['del'] = 10
16 compare_coherence.plot(kind='line',
17                        xticks=[x for x in range(21)],
18                        yticks=[round(x*0.001, 5) for x in range(-15,15)],
19                        xlim=[0,20],
20                        ylim=[-0.015,0.01],
21                        xlabel="Number of topics",
22                        ylabel="Coherence score",
23                        title="Coherence Score Comparison Different Document Scopes and Topic Counts",
24                        ).legend(loc='center left',bbox to anchor=(1.0, 0.5))

```

Out[76]: <matplotlib.legend.Legend at 0x2096fa7b040>

