```
!pip install pyLDAvis -qq
!pip install -qq -U gensim
!pip install spacy -qq
!pip install matplotlib -qq
!pip install seaborn -qq
```

```
import pandas as pd
import spacy
import seaborn as sns
sns.set()
import spacy
import pyLDAvis.gensim_models
pyLDAvis.enable_notebook()# Visualise inside a notebook
import en_core_web_md
from gensim.corpora.dictionary import Dictionary
from gensim.models import LdaMulticore
from gensim.models import CoherenceModel
```

```
C:\Users\Vaishali\anaconda3\lib\site-packages\scipy\sparse\sparsetools.py:21: DeprecationWarning: `scipy.sparse.sparsetools` is
scipy.sparse.sparsetools is a private module for scipy.sparse, and should not be used.
  _deprecated()
```

```
reports = pd.read_csv("../FYP/processed_data.csv")
reports.head()
```

	news_date	tokens
0	1/6/2011 2:45:49 PM	நாலு,ஆள்,உயரம்,முறுக்கு,மீசை,கையில்,வீச்சரிவாள...
1	1/6/2011 2:56:51 PM	அமானுஷ்யமான,சம்பவம்,நம்,சுற்றி,ஆங்காங்கே,நட,கொ...
2	1/6/2011 3:02:00 PM	காமன்வெல்த்,போட்டி,ஏற்பாட்டில்,நடைபெறு,முறைகேட...
3	1/6/2011 3:08:15 PM	தென்அமெரிக்க,நாடான,பெருவில்,காடுகள்,பயங்கரமானவ...
4	1/6/2011 3:09:20 PM	கடந்த,ம்,தேதி,சாயங்காலம்,அடைமழையை,கிழித்தபடி,ச...

```
reports.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126746 entries, 0 to 126745
Data columns (total 2 columns):
 #   Column     Non-Null Count   Dtype 
---  ------     --------------   ----- 
 0   news_date  126746 non-null  object
 1   tokens     126746 non-null  object
dtypes: object(2)
memory usage: 1.9+ MB
```

```
reports["tokens"] = reports["tokens"].map(lambda x: x.split(","))

reports.head()
```

	news_date	tokens
0	1/6/2011 2:45:49 PM	[நாலு, ஆள், உயரம், முறுக்கு, மீசை, கையில், வீச...
1	1/6/2011 2:56:51 PM	[அமானுஷ்யமான, சம்பவம், நம், சுற்றி, ஆங்காங்கே,...
2	1/6/2011 3:02:00 PM	[காமன்வெல்த், போட்டி, ஏற்பாட்டில், நடைபெறு, மு...
3	1/6/2011 3:08:15 PM	[தென்அமெரிக்க, நாடான, பெருவில், காடுகள், பயங்க...
4	1/6/2011 3:09:20 PM	[கடந்த, ம், தேதி, சாயங்காலம், அடைமழையை, கிழித்...

```
len(reports)
```

```
126746
```

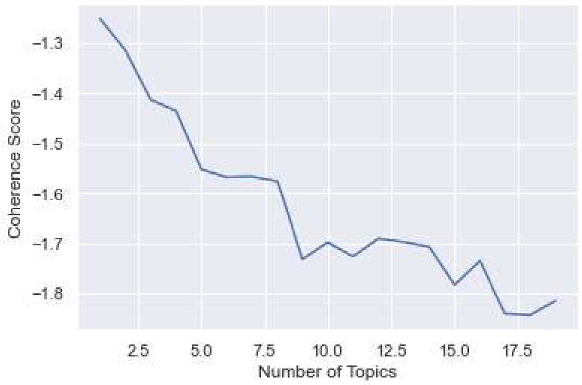
```
from gensim import corpora

text_data = reports["tokens"]
dictionary = corpora.Dictionary(text_data)
dictionary.filter_extremes(no_below=5, no_above=0.5, keep_n=1000)
corpus = [dictionary.doc2bow(text) for text in text_data]
```

```
lda_model = LdaMulticore(corpus=corpus, id2word=dictionary, iterations=50, num_topics=10, workers = 4, passes=10)
```

```
topics = []
score = []
```

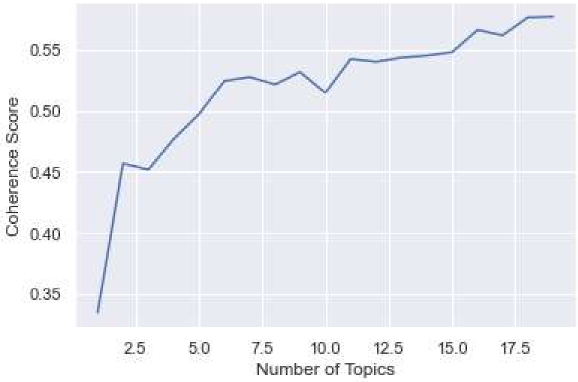
```
#C_umass score
for i in range(1,20,1):
    lda_model = LdaMulticore(corpus=corpus, id2word=dictionary, iterations=10, num_topics=i, workers = 4, passes=10, random_state
    cm = CoherenceModel(model=lda_model, corpus=corpus, dictionary=dictionary, coherence='u_mass')
    topics.append(i)
    score.append(cm.get_coherence())
```

```
import matplotlib.pyplot as plt
plt.plot(topics, score)
plt.xlabel('Number of Topics')
plt.ylabel('Coherence Score')
plt.show()
```



```
#C_v score
topics = []
score = []
for i in range(1,20,1):
    lda_model = LdaMulticore(corpus=corpus, id2word=dictionary, iterations=10, num_topics=i, workers = 4, passes=10, random_state
    cm = CoherenceModel(model=lda_model, texts = reports['tokens'], corpus=corpus, dictionary=dictionary, coherence='c_v')
    topics.append(i)
    score.append(cm.get_coherence())
```

```
plt.plot(topics, score)
plt.xlabel('Number of Topics')
plt.ylabel('Coherence Score')
plt.show()
```



```
lda_model = LdaMulticore(corpus=corpus, id2word=dictionary, iterations=100, num_topics=7, workers = 4, passes=100)
```

```
lda_model.print_topics(-1)
```

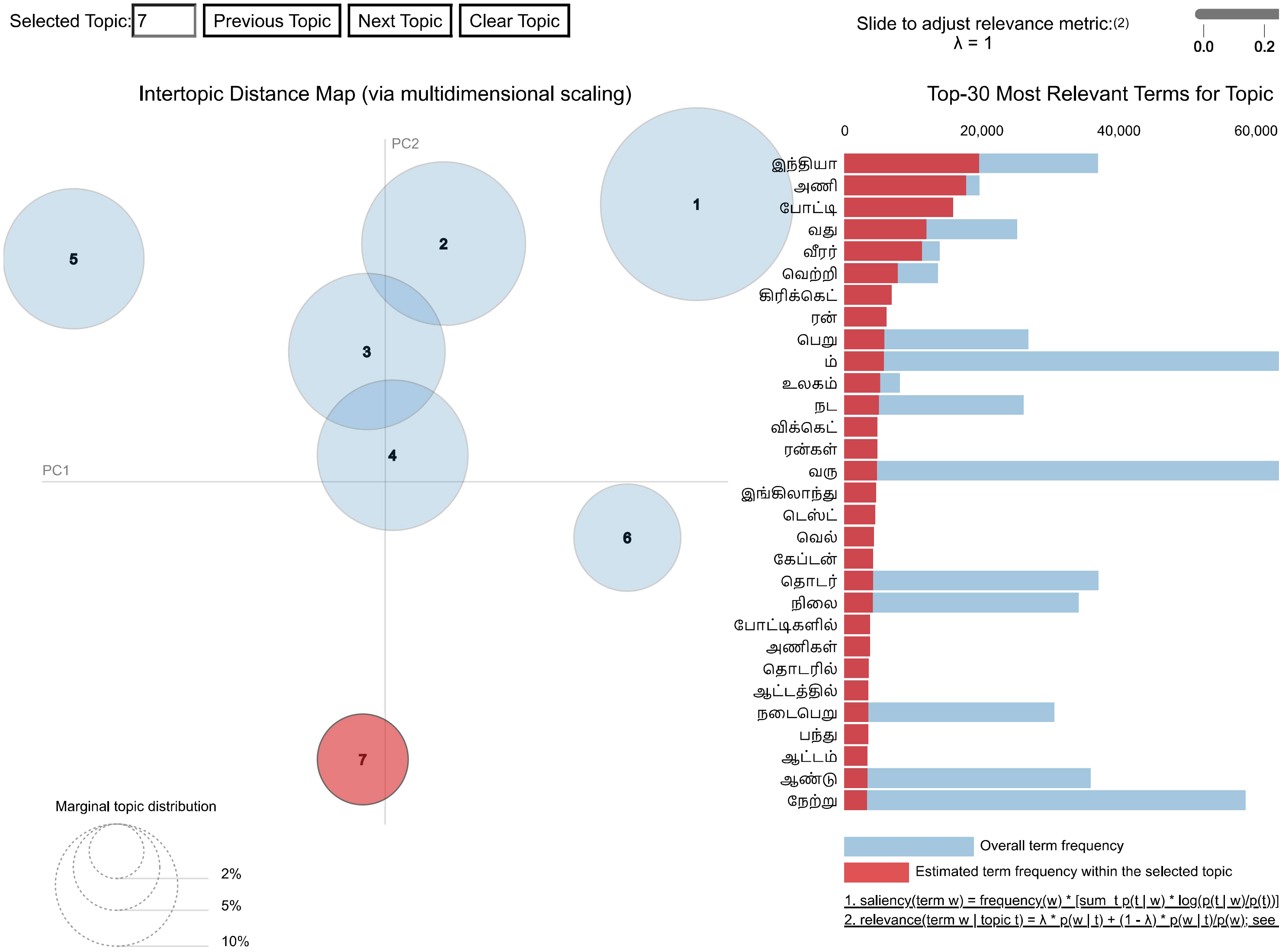
```
[(0,
  '0.024*"வரு" + 0.021*"பகுதி" + 0.019*"மக்கள்" + 0.019*"மணி" + 0.017*"செல்" + 0.015*"மாவட்டம்" + 0.015*"சென்னை" + 0.
*"காலை"'),
 (1,
  '0.041*"போலீசார்" + 0.028*"வா" + 0.028*"சேர்" + 0.019*"செல்" + 0.015*"விசாரணை" + 0.014*"நேற்று" + 0.014*"கைது" + 0.
ருகே"'),
 (2,
  '0.034*"ரூ" + 0.029*"ம்" + 0.028*"அரசு" + 0.018*"வேண்டு" + 0.017*"வழக்கு" + 0.016*"தேதி" + 0.015*"ஆண்டு" + 0.015*"கடந்
 (3,
  '0.019*"வேண்டு" + 0.017*"ஆனால்" + 0.016*"வா" + 0.015*"இல்" + 0.015*"படம்" + 0.015*"படத்தில்" + 0.014*"கூறு" + 0.012*"
 (4,
  '0.030*"அரசு" + 0.028*"மத்திய" + 0.024*"வரு" + 0.018*"வேண்டு" + 0.016*"நடத்து" + 0.015*"இந்தியா" + 0.014*"நாடு" + 0.01
*"குறி"'),
 (5,
  '0.042*"தேர்தல்" + 0.038*"தலைவர்" + 0.029*"கட்சி" + 0.023*"காங்கிரஸ்" + 0.023*"மாவட்டம்" + 0.021*"திமுக" + 0.020*"பாஜ
+ 0.017*"அதிமுக"'),
 (6,
  '0.051*"இந்தியா" + 0.046*"அணி" + 0.041*"போட்டி" + 0.031*"வது" + 0.030*"வீரர்" + 0.020*"வெற்றி" + 0.018*"கிரிக்கெட்" + 0
```

```
lda_model[corpus][0]
```

```
[(0, 0.48286268), (1, 0.35839072), (3, 0.14729519)]
```

```
lda_display = pyLDAvis.gensim_models.prepare(lda_model, corpus, dictionary)
pyLDAvis.display(lda_display)
```

```
C:\Users\Vaishali\anaconda3\lib\site-packages\pyLDAvis\_prepare.py:243: FutureWarning: In a future version of pandas all argument
'labels' will be keyword-only
  default_term_info = default_term_info.sort_values(
```


Selected Topic: 7
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
5
6
7
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic
0
20,000
40,000
60,000
இந்தியா
அணி
போட்டி
வது
வீரர்
வெற்றி
கிரிக்கெட்
ரன்
பெறு
ம்
உலகம்
நட
விக்கெட்
ரன்கள்
வரு
இங்கிலாந்து
டெஸ்ட்
வெல்
கேப்டன்
தொடர்
நிலை
போட்டிகளில்
அணிகள்
தொடரில்
ஆட்டத்தில்
நடைபெறு
பந்து
ஆட்டம்
ஆண்டு
நேற்று
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))]
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see