EDA_code

December 6, 2024

[41]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter
from wordcloud import WordCloud


data= pd.read_csv('data/bbc_news_text_complexity_summarization.csv')
data.head()
```

[41]:

```
                                              text    labels  no_sentences  \
0  Ad sales boost Time Warner profit\n\nQuarterly…  business            26   
1  Dollar gains on Greenspan speech\n\nThe dollar…  business            17   
2  Yukos unit buyer faces loan claim\n\nThe owner…  business            14   
3  High fuel prices hit BA's profits\n\nBritish A…  business            24   
4  Pernod takeover talk lifts Domecq\n\nShares in…  business            17   

   Flesch Reading Ease Score  Dale-Chall Readability Score  \
0                      62.17                          9.72   
1                      65.56                          9.09   
2                      69.21                          9.66   
3                      62.98                          9.86   
4                      70.63                         10.23   

                                 text_rank_summary  \
0  It hopes to increase subscribers by offering t…   
1  The dollar has hit its highest level against t…   
2  The owners of embattled Russian oil giant Yuko…   
3  Looking ahead to its full year results to Marc…   
4  Reports in the Wall Street Journal and the Fin…   

                                       lsa_summary  
0  Its profits were buoyed by one-off gains which…  
1  "I think the chairman's taking a much more san…  
2  Yukos' owner Menatep Group says it will ask Ro…  
3  Rod Eddington, BA's chief executive, said the …  
4  Shares in UK drinks and food firm Allied Domec…  
```

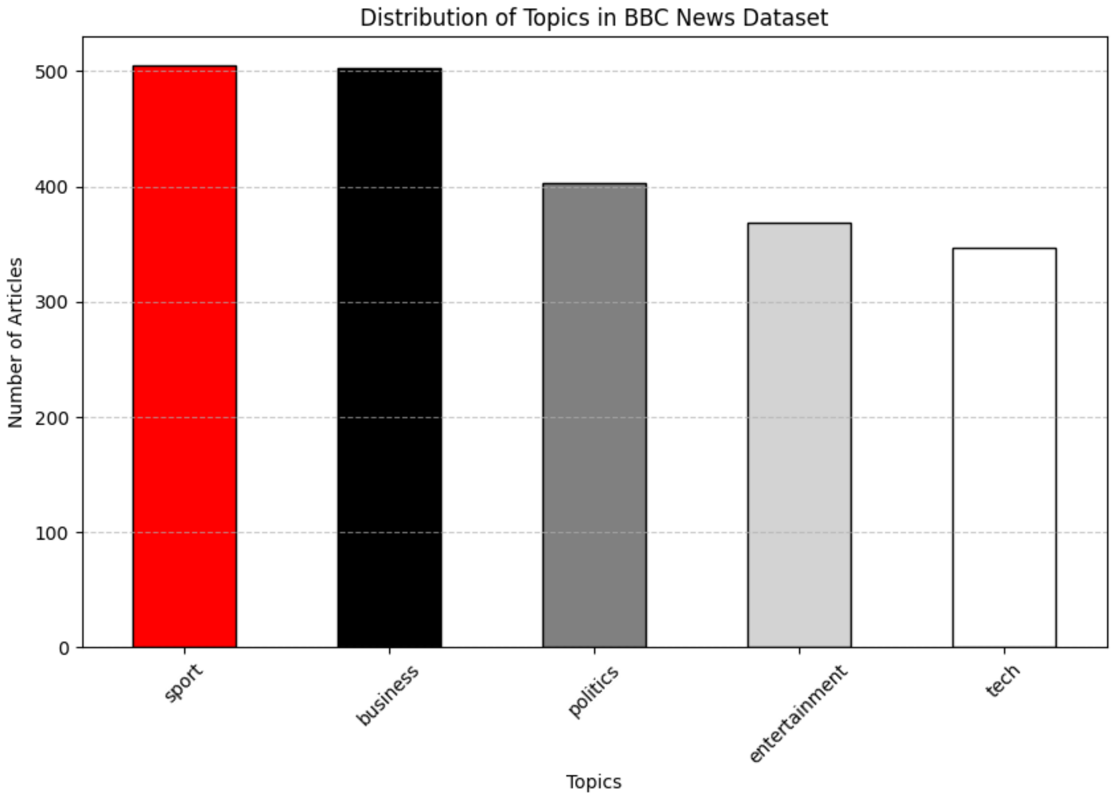
[42]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2127 entries, 0 to 2126
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   text                          2127 non-null   object
 1   labels                        2127 non-null   object
 2   no_sentences                  2127 non-null   int64
 3   Flesch Reading Ease Score     2127 non-null   float64
 4   Dale-Chall Readability Score  2127 non-null   float64
 5   text_rank_summary             2127 non-null   object
 6   lsa_summary                   2127 non-null   object
dtypes: float64(2), int64(1), object(4)
memory usage: 116.4+ KB
```

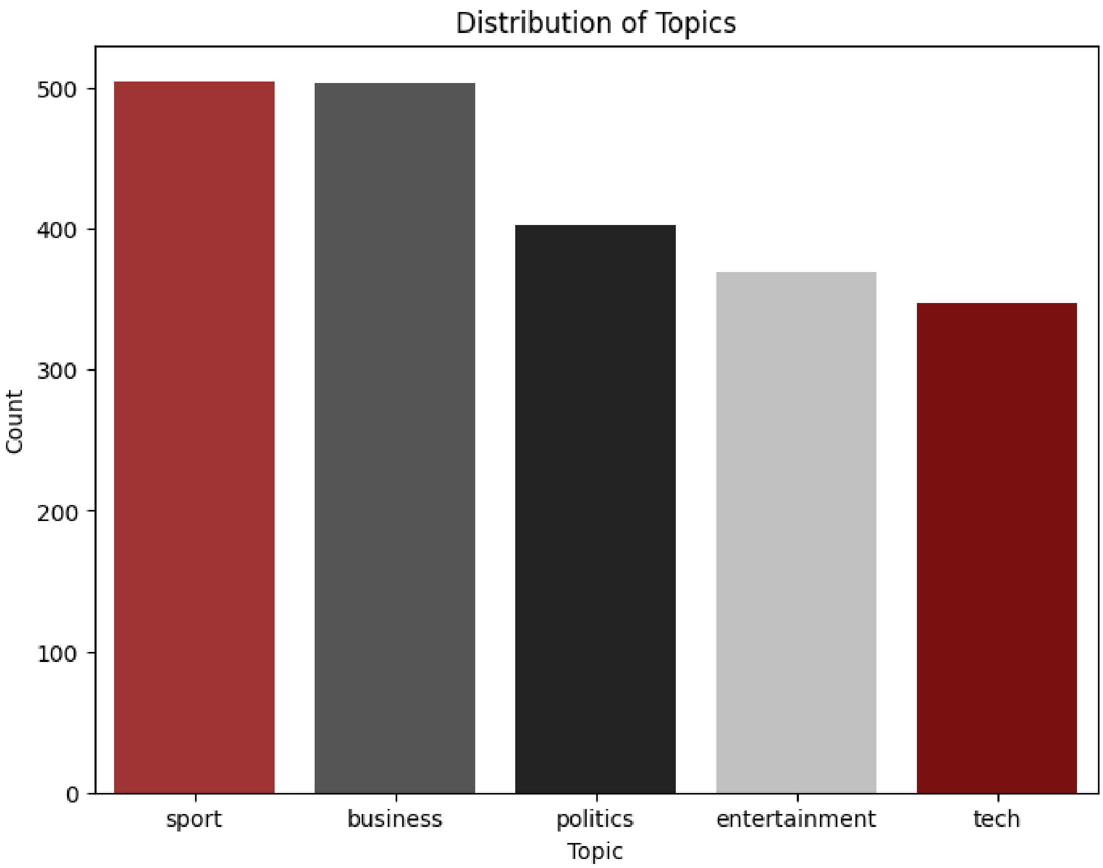
[43]:

```
import matplotlib.pyplot as plt

# Define colors based on user preference
colors = ['#FF0000', '#000000', '#808080', '#D3D3D3', '#FFFFFF']  # red, black,␣
 ↪gray, light gray, white

# Plot topic distribution
plt.figure(figsize=(10, 6))
data['labels'].value_counts().plot(kind='bar', color=colors, edgecolor='black')
plt.title('Distribution of Topics in BBC News Dataset')
plt.xlabel('Topics')
plt.ylabel('Number of Articles')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()
```



[44]:

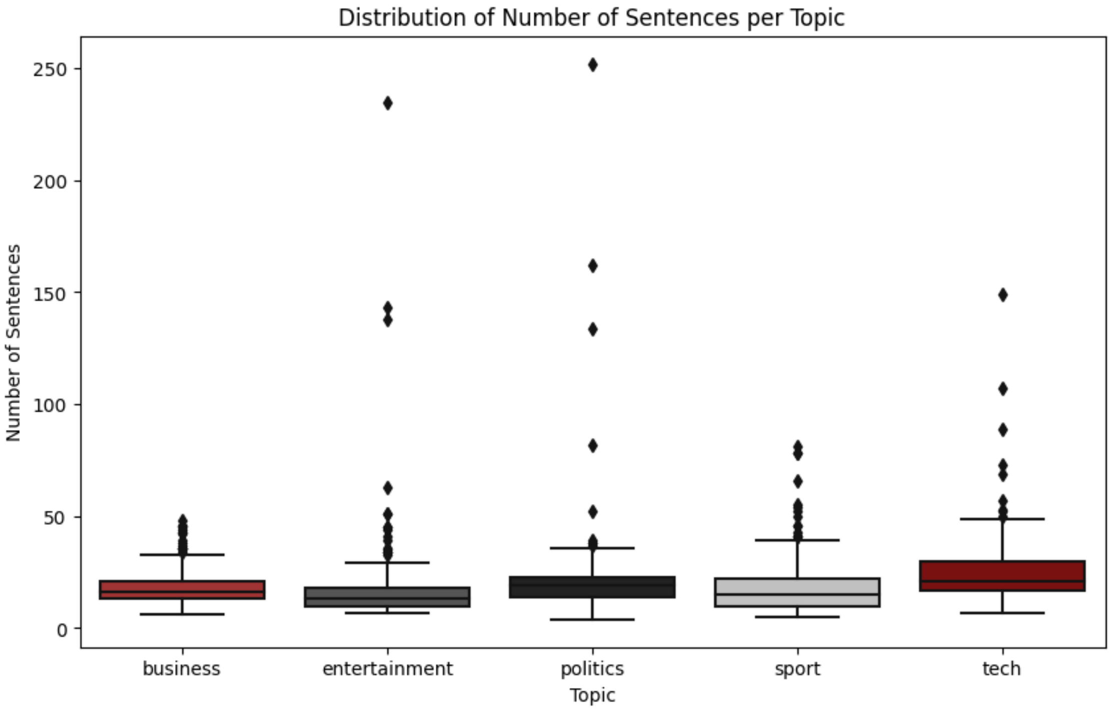
```
# Adjusting the colors for the distribution of topics plot
plt.figure(figsize=(8, 6))
sns.countplot(x='labels', data=data, order=data['labels'].value_counts().index,␣
  ↪palette=['#B22222', '#555555', '#222222', '#C0C0C0', '#8B0000'])
plt.title('Distribution of Topics', color='black')
plt.xlabel('Topic', color='black')
plt.ylabel('Count', color='black')
plt.xticks(color='black')
plt.yticks(color='black')
plt.savefig("/Users/marwahfaraj/Desktop/ms_degree_application_and_doc/
  ↪final_projects/502_final_project/BBC_new_topic_classification/image/
  ↪topice_dis.jpg")
plt.show()
```



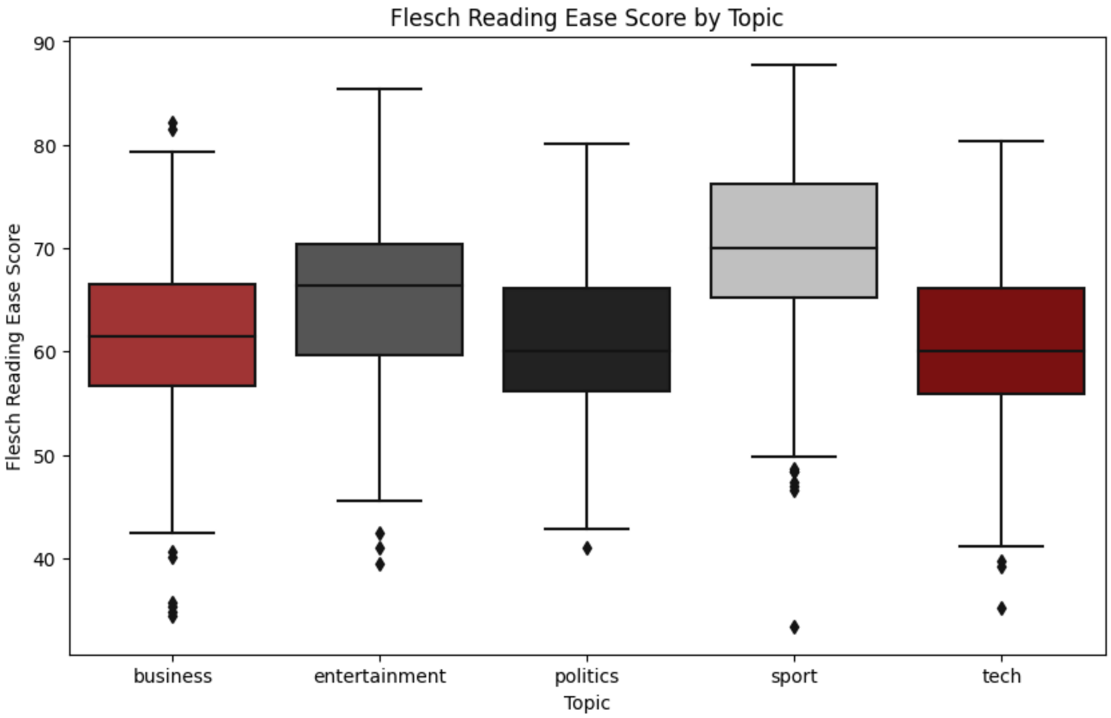
The bar chart illustrates the distribution of articles across five topics in the BBC News dataset: Sport, Business, Politics, Entertainment, and Tech. The dataset is relatively balanced, with Sport and Business topics having the highest representation, each around 500 articles. Politics follows closely, while Entertainment and Tech have fewer entries. This balanced distribution suggests that the dataset is well-suited for classification tasks, as each topic has a comparable number of samples. However, slight imbalances could impact the model's accuracy for less-represented categories like Tech.

[45]:

```
# Adjusting the colors for the boxplot of number of sentences per topic
plt.figure(figsize=(10, 6))
sns.boxplot(x='labels', y='no_sentences', data=data, palette=['#B22222',␣
 ↪'#555555', '#222222', '#C0C0C0', '#8B0000'])
plt.title('Distribution of Number of Sentences per Topic', color='black')
plt.xlabel('Topic', color='black')
plt.ylabel('Number of Sentences', color='black')
plt.xticks(color='black')
plt.yticks(color='black')
plt.show()
```



[46]:
```
# Adjusting the colors for Flesch Reading Ease Score per topic
plt.figure(figsize=(10, 6))
sns.boxplot(x='labels', y='Flesch Reading Ease Score', data=data,␣
 ↪palette=['#B22222', '#555555', '#222222', '#C0C0C0', '#8B0000'])
plt.title('Flesch Reading Ease Score by Topic', color='black')
plt.xlabel('Topic', color='black')
plt.ylabel('Flesch Reading Ease Score', color='black')
plt.xticks(color='black')
plt.yticks(color='black')
plt.savefig("/Users/marwahfaraj/Desktop/ms_degree_application_and_doc/
 ↪final_projects/502_final_project/BBC_new_topic_classification/image/
 ↪flesch_reading.jpg")

plt.show()
```

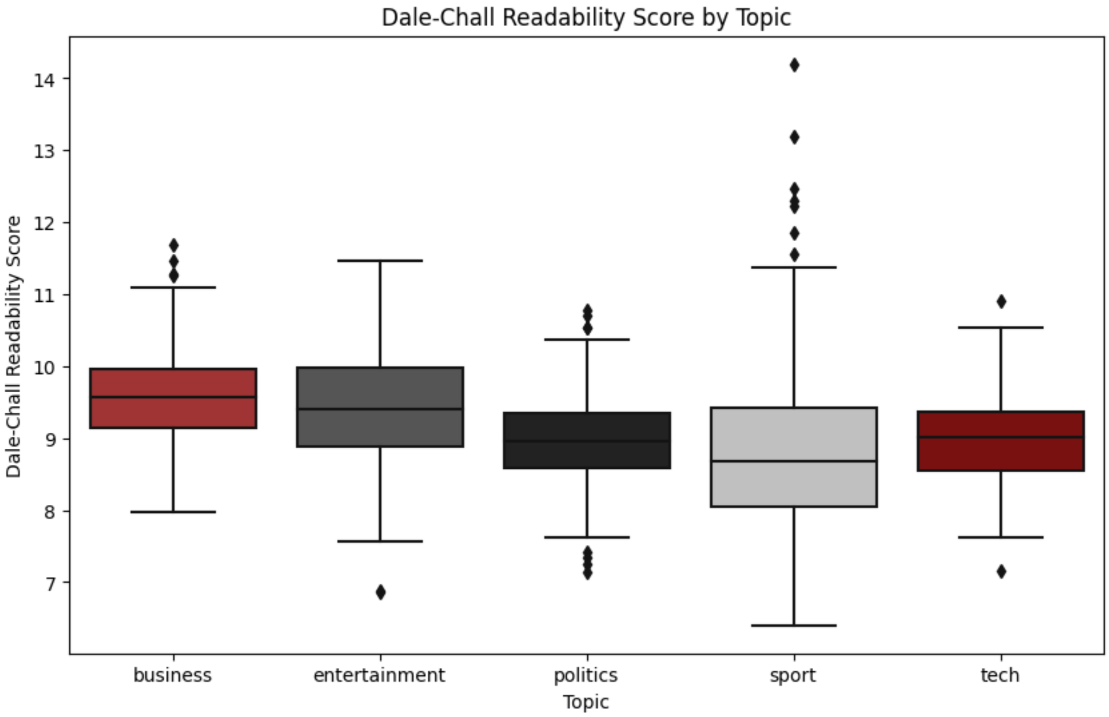


The box plot shows the Flesch Reading Ease Score distribution for each topic, providing insights into readability:

Politics and Business articles have a wider range and tend to have lower readability scores, suggesting they are more complex. Entertainment and Sport articles generally have higher readability scores, indicating simpler language, which might make them more accessible to a broader audience. Tech articles have a mid-range readability, likely due to specific terminology but generally accessible language. These readability variations may affect classification accuracy, as certain topics might have more distinctive linguistic patterns.

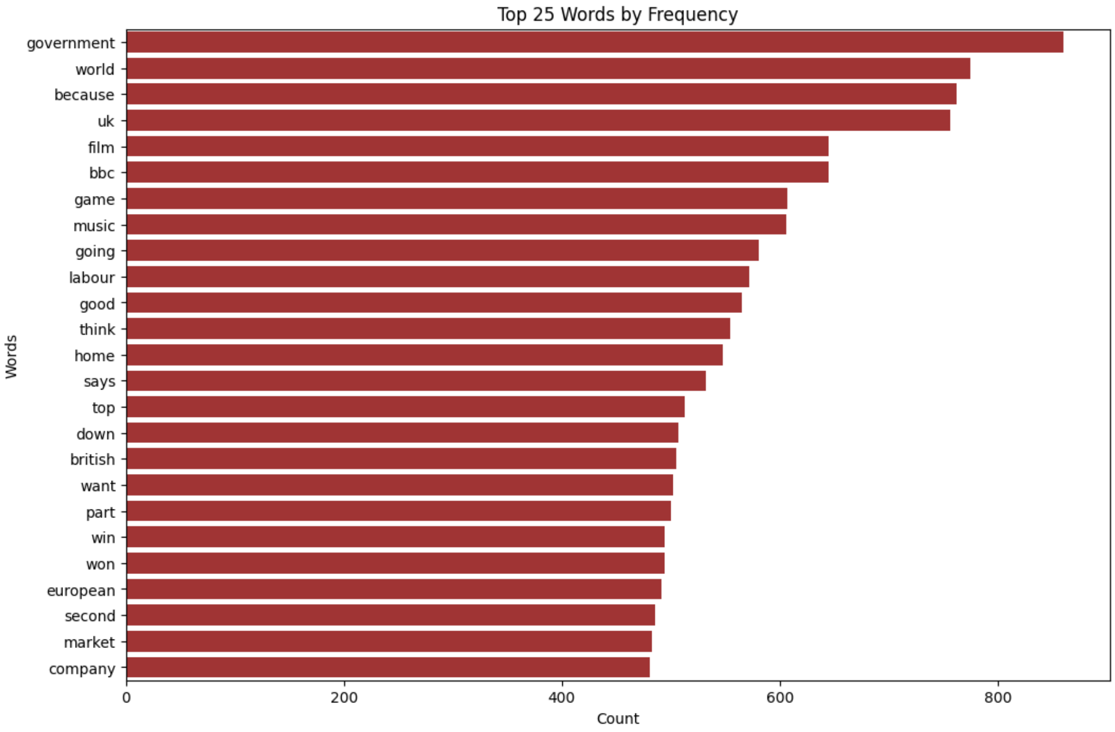
[47]:

```python
# Adjusting the colors for Dale-Chall Readability Score per topic
plt.figure(figsize=(10, 6))
sns.boxplot(x='labels', y='Dale-Chall Readability Score', data=data, palette=['#B22222', '#555555', '#222222', '#C0C0C0', '#8B0000'])
plt.title('Dale-Chall Readability Score by Topic', color='black')
plt.xlabel('Topic', color='black')
plt.ylabel('Dale-Chall Readability Score', color='black')
plt.xticks(color='black')
plt.yticks(color='black')
plt.show()
```



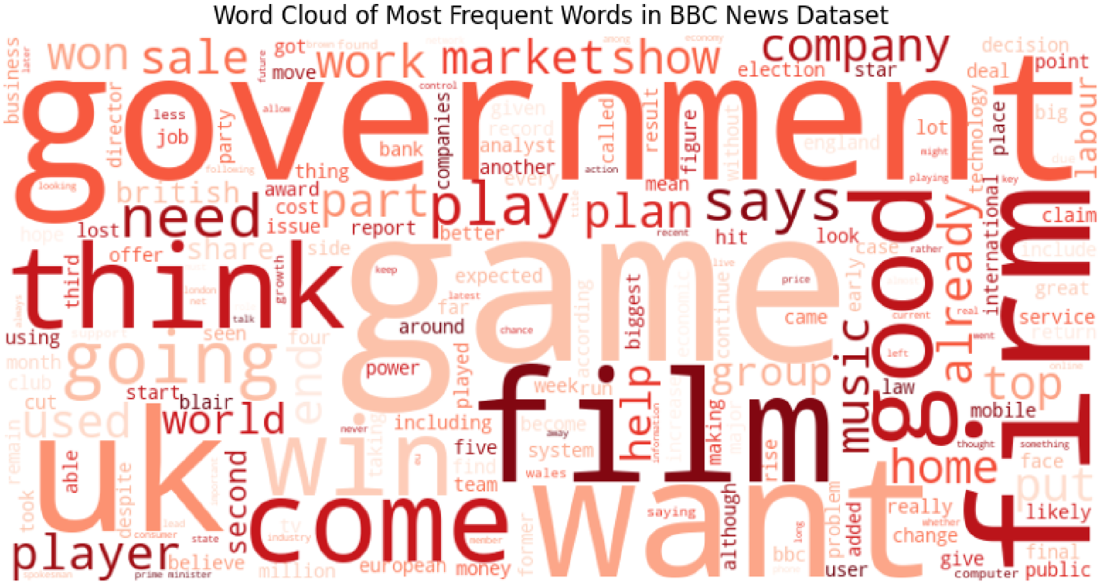
[48]:

```
# Combine all text into a single string and split into words
all_words = " ".join(data['text']).lower().split()

# Define an extended set of stopwords to remove common, meaningless words
extended_stopwords = set([
    'the', 'and', 'to', 'of', 'in', 'for', 'that', 'on', 'with', 'as', 'is',
    'was', 'at', 'by', 'an', 'be', 'this', 'which', 'or', 'from', 'but', 'it',
    'are', 'not', 'have', 'has', 'had', 'we', 'will', 'can', 'more', 'about',
    'their', 'been', 'after', 'they', 'you', 'all', 'out', 'up', 'if', 'who',␣
 ↪'what',
    'he', 'said', 'his', 'mr', 'she', 'her', 'they', 'them', 'its',
    'would', 'one', 'could', 'new', 'also', 'people', 'us', 'over', 'first',
    'last', 'two', 'may', 'many', 'much', 'even', 'make', 'made', 'time',
    'year', 'years', 'still', 'know', 'back', 'way', 'right', 'say', 'day',␣
 ↪'days',
    'now', 'so', 'like', 'a', 'i', 'just', 'only', 'get', 'told', 'no', 'do',␣
 ↪'such',
    'best', 'being', 'other', 'some', 'when', 'were', 'than', 'against',␣
 ↪'should',
    't', 'yes', 'said', 'well', 'there', 'into', 'before', 'while', 'any',␣
 ↪'next', 'most', 'while',
    'take', 'my', 'our', 'set', 'since', 'then', 'go', 'between', 'see',␣
 ↪'very', 'becuase', 'three',
    'number', 'how', 'those', 'under','use'
])

# Filter out stopwords
filtered_words_final = [word for word in all_words if word.isalpha() and word␣
  ↪not in extended_stopwords]
word_counts_final = Counter(filtered_words_final)

# Get the 25 most common words and sort them in descending order of frequency
top_25_words_final = word_counts_final.most_common(25)
words_sorted, counts_sorted = zip(*sorted(top_25_words_final, key=lambda x:␣
  ↪x[1], reverse=True))

# Plot the final refined top 25 words in a horizontal bar chart with the␣
  ↪longest bar at the top
plt.figure(figsize=(12, 8))
sns.barplot(y=list(words_sorted), x=list(counts_sorted), palette=['#B22222']*25)
plt.title('Top 25 Words by Frequency', color='black')
plt.xlabel('Count', color='black')
plt.ylabel('Words', color='black')
plt.xticks(color='black')
plt.yticks(color='black')
plt.savefig("/Users/marwahfaraj/Desktop/ms_degree_application_and_doc/
  ↪final_projects/502_final_project/BBC_new_topic_classification/image/
  ↪top_25_words.jpg")
plt.show()
```



[49]:

```
from collections import Counter

# Recreate all_words by combining all text entries
all_words = " ".join(data['text']).lower().split()

# Define extended stopwords as specified
extended_stopwords = set([
    'the', 'and', 'to', 'of', 'in', 'for', 'that', 'on', 'with', 'as', 'is',
    'was', 'at', 'by', 'an', 'be', 'this', 'which', 'or', 'from', 'but', 'it',
    'are', 'not', 'have', 'has', 'had', 'we', 'will', 'can', 'more', 'about',
    'their', 'been', 'after', 'they', 'you', 'all', 'out', 'up', 'if', 'who',␣
 ↪'what',
    'he', 'said', 'his', 'mr', 'she', 'her', 'they', 'them', 'its',
    'would', 'one', 'could', 'new', 'also', 'people', 'us', 'over', 'first',
    'last', 'two', 'may', 'many', 'much', 'even', 'make', 'made', 'time',
    'year', 'years', 'still', 'know', 'back', 'way', 'right', 'say', 'day',␣
 ↪'days',
    'now', 'so', 'like', 'a', 'i', 'just', 'only', 'get', 'told', 'no', 'do',␣
 ↪'such',
    'best', 'being', 'other', 'some', 'when', 'were', 'than', 'against',␣
 ↪'should',
    't', 'yes', 'said', 'well', 'there', 'into', 'before', 'while', 'any',␣
 ↪'next', 'most', 'while',
    'take', 'my', 'our', 'set', 'since', 'then', 'go', 'between', 'see',␣
 ↪'very', 'becuase', 'three',
    'number', 'how', 'those', 'under','use'
])
# Filter out stopwords
filtered_words_final = [word for word in all_words if word.isalpha() and word␣
 ↪not in extended_stopwords]

# Generate the word cloud
wordcloud = WordCloud(width=800, height=400, background_color='white',␣
 ↪colormap='Reds').generate(" ".join(filtered_words_final))

# Display the word cloud
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud of Most Frequent Words in BBC News Dataset',␣
 ↪color='black')
plt.savefig("/Users/marwahfaraj/Desktop/ms_degree_application_and_doc/
 ↪final_projects/502_final_project/BBC_new_topic_classification/image/world␣
 ↪cloud.jpg")
plt.show()
```



```
[50]: from wordcloud import WordCloud
      import pandas as pd
```

```
# Generate the word cloud
wordcloud = WordCloud(
    width=800,
    height=800,
    background_color='white',
    colormap='Reds',
    max_words=200
).generate(" ".join(filtered_words_final))

# Extract the words and their frequencies
words_frequencies = wordcloud.words_

# Convert relative frequencies to integer weights
max_weight = 100  # Choose a scale for the weights
word_weights = {word: int(freq * max_weight) for word, freq in␣
 ↪words_frequencies.items()}

# Convert to a DataFrame with the correct format
word_freq_df = pd.DataFrame(word_weights.items(), columns=["Word", "Weight"])
word_freq_df = word_freq_df[["Weight", "Word"]]  # Swap columns to match␣
 ↪expected format

# Save to a CSV file
word_freq_df.to_csv('word_cloud_weights.csv', index=False, header=False)

print("CSV created successfully in 'word_cloud_weights.csv'.")
```

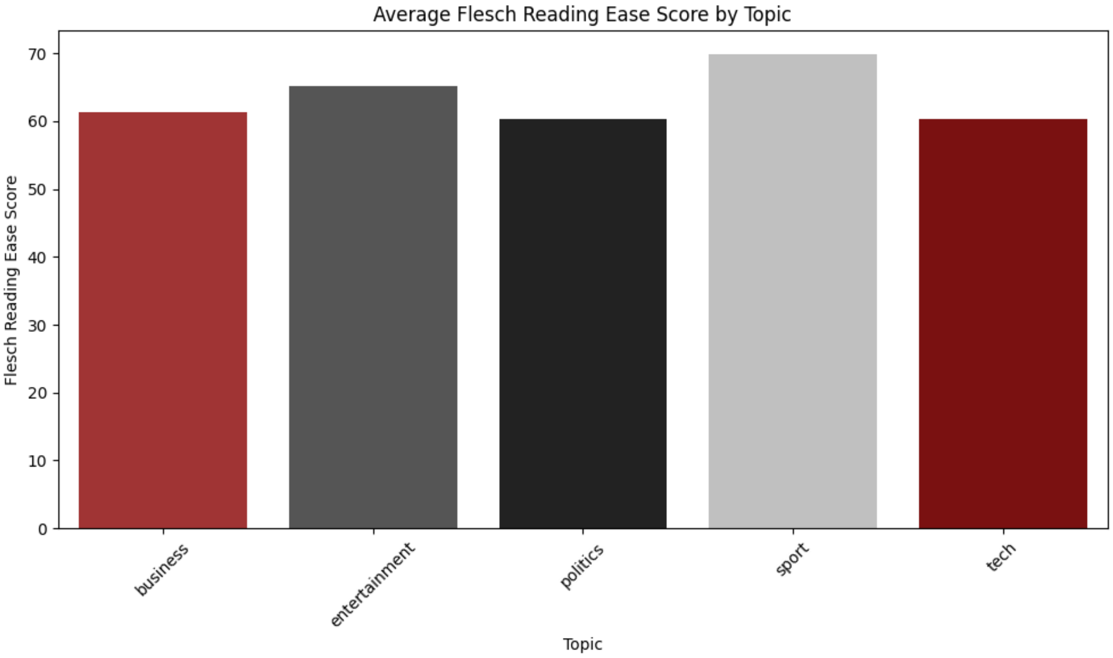
```
CSV created successfully in 'word_cloud_weights.csv'.
```

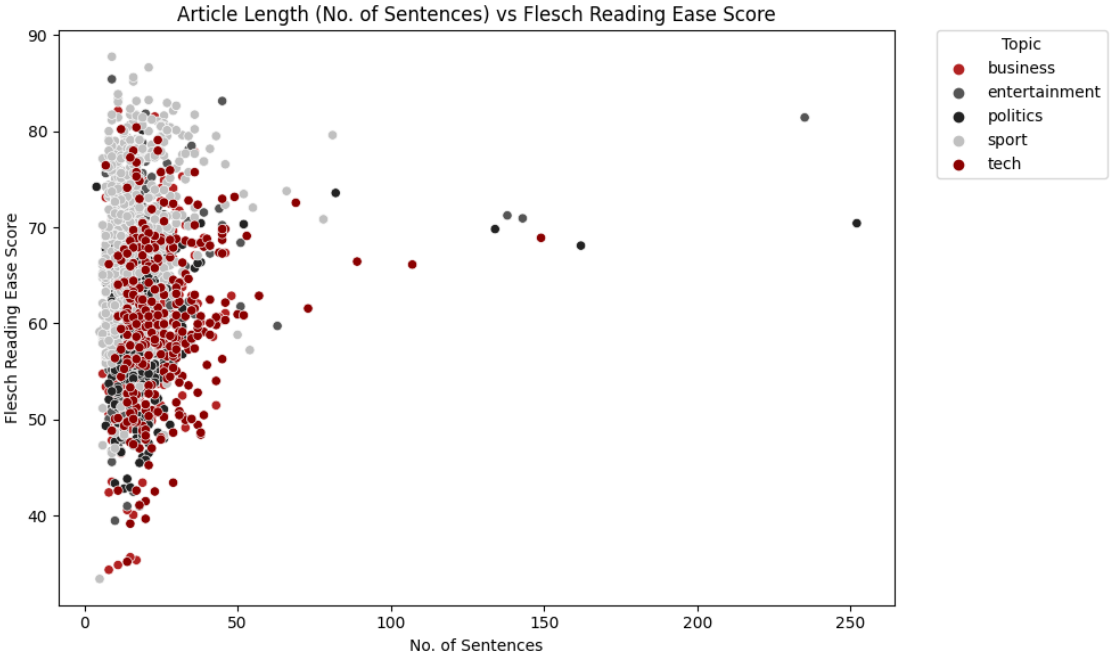
[51]:
```
data.columns
```

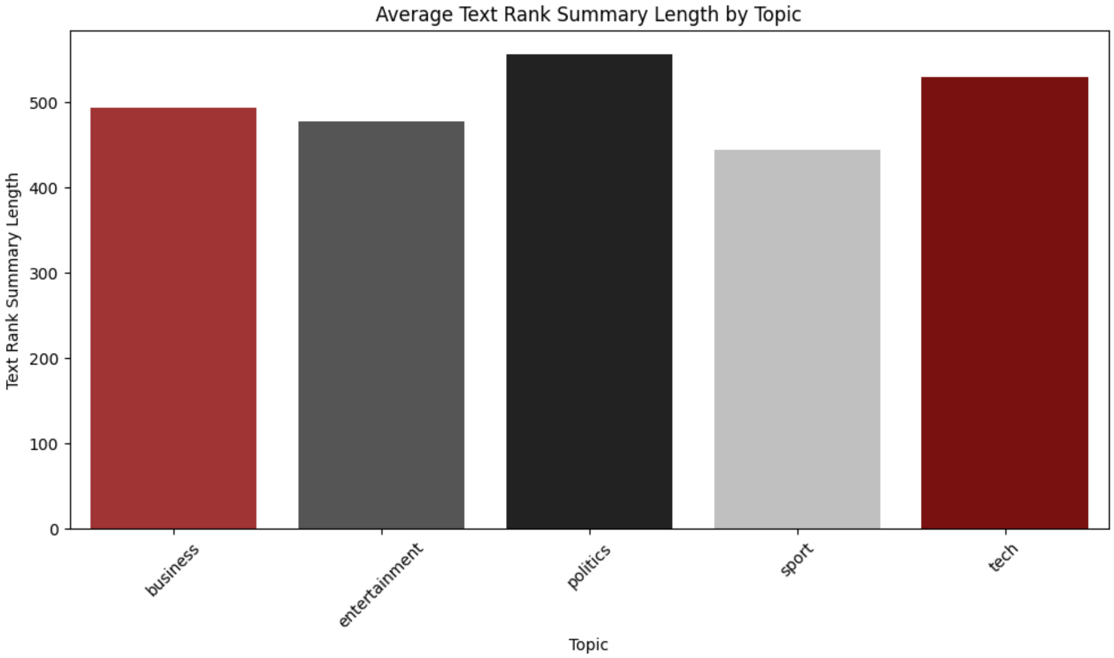
[51]:
```
Index(['text', 'labels', 'no_sentences', 'Flesch Reading Ease Score',
       'Dale-Chall Readability Score', 'text_rank_summary', 'lsa_summary'],
      dtype='object')
```

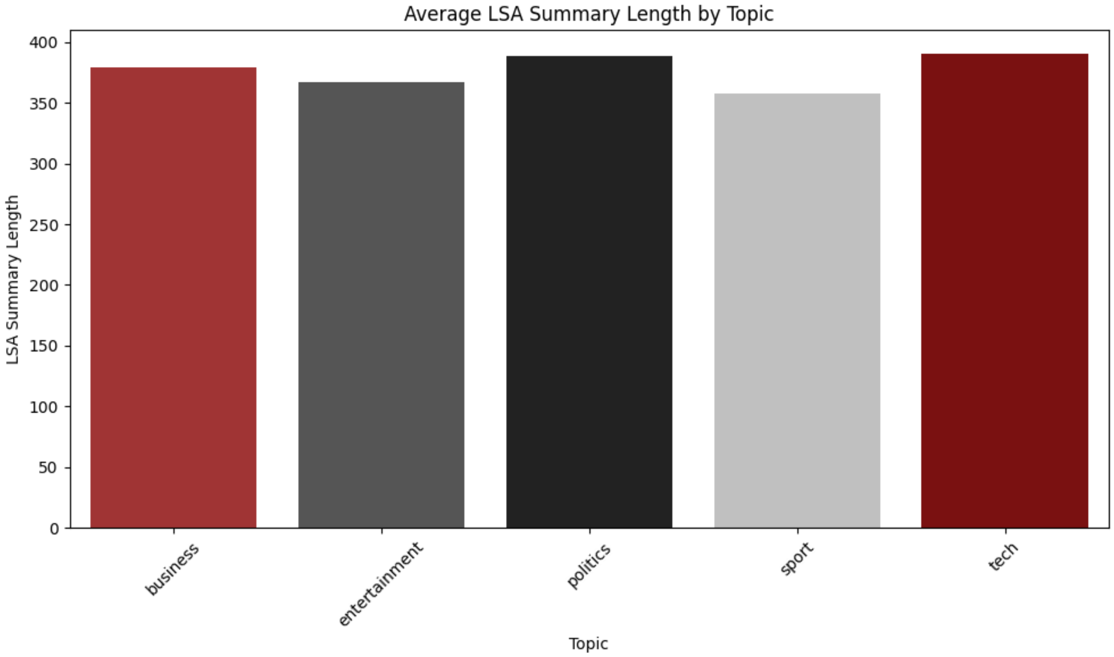
[54]:
```
# Define the palette
custom_palette = ['#B22222', '#555555', '#222222', '#C0C0C0', '#8B0000']
# 1. Topic Complexity Analysis
plt.figure(figsize=(10, 6))
sns.barplot(x='labels', y='Flesch Reading Ease Score', data=topic_complexity,␣
 ↪palette=custom_palette)
plt.title('Average Flesch Reading Ease Score by Topic')
plt.ylabel('Flesch Reading Ease Score')
plt.xlabel('Topic')
plt.xticks(rotation=45)
plt.tight_layout()  # Adjust layout to prevent clipping
plt.savefig("/Users/marwahfaraj/Desktop/ms_degree_application_and_doc/
 ↪final_projects/502_final_project/BBC_new_topic_classification/image/
 ↪ease_of_reading_by_topice.jpg")
plt.show()

# 2. Article Length vs. Readability
plt.figure(figsize=(10, 6))
sns.scatterplot(x='no_sentences', y='Flesch Reading Ease Score', hue='labels',␣
 ↪data=data, palette=custom_palette)
plt.title('Article Length (No. of Sentences) vs Flesch Reading Ease Score')
plt.xlabel('No. of Sentences')
plt.ylabel('Flesch Reading Ease Score')
plt.legend(title='Topic', bbox_to_anchor=(1.05, 1), loc='upper left',␣
 ↪borderaxespad=0)
plt.tight_layout()  # Adjust layout to prevent clipping
plt.savefig("/Users/marwahfaraj/Desktop/ms_degree_application_and_doc/
 ↪final_projects/502_final_project/BBC_new_topic_classification/image/
 ↪news_text_length_vs_ease_of_reading.jpg", bbox_inches='tight')  # Ensure␣
 ↪legend is saved
plt.show()

# 3. Summarization Quality Analysis
plt.figure(figsize=(10, 6))
sns.barplot(x='labels', y='text_rank_summary_length', data=summ_quality_topic,␣
 ↪palette=custom_palette)
plt.title('Average Text Rank Summary Length by Topic')
plt.ylabel('Text Rank Summary Length')
plt.xlabel('Topic')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("/Users/marwahfaraj/Desktop/ms_degree_application_and_doc/
 ↪final_projects/502_final_project/BBC_new_topic_classification/image/
 ↪avg_text_rank_summery_length_by_topic.jpg", bbox_inches='tight')
plt.show()

plt.figure(figsize=(10, 6))
sns.barplot(x='labels', y='lsa_summary_length', data=summ_quality_topic,␣
 ↪palette=custom_palette)
plt.title('Average LSA Summary Length by Topic')
plt.ylabel('LSA Summary Length')
plt.xlabel('Topic')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("/Users/marwahfaraj/Desktop/ms_degree_application_and_doc/
 ↪final_projects/502_final_project/BBC_new_topic_classification/image/
 ↪average_LSA_summary_length_by_topic.jpg", bbox_inches='tight')
```

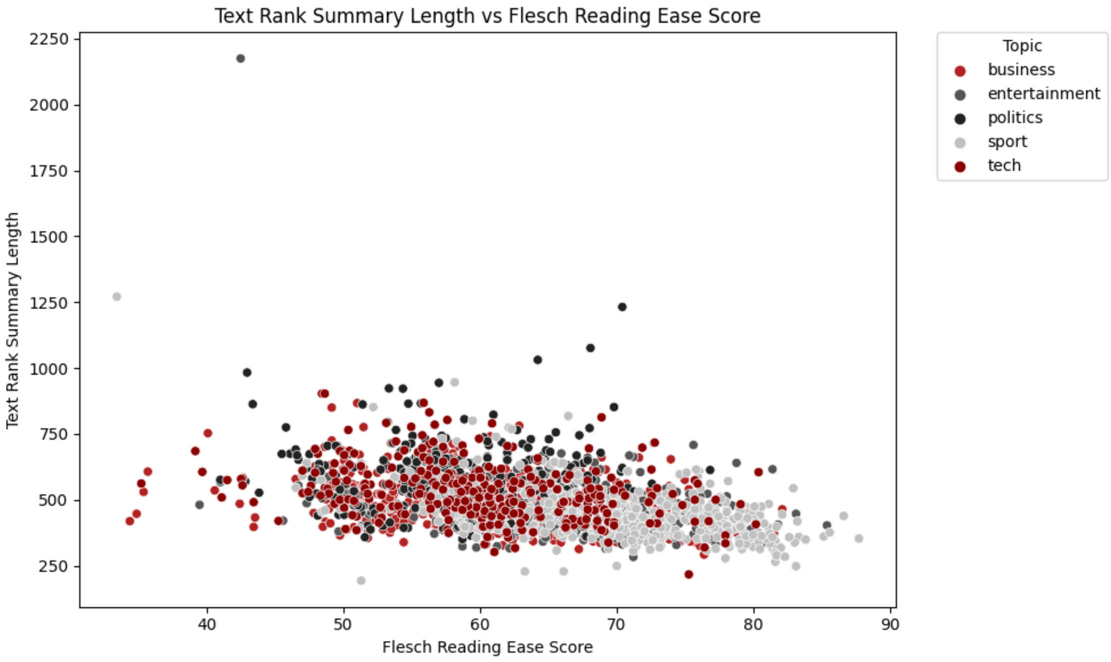
```
plt.show()

# 4. Summarization Quality vs. Complexity
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Flesch Reading Ease Score', y='text_rank_summary_length',␣
 ↪hue='labels', data=data, palette=custom_palette)
plt.title('Text Rank Summary Length vs Flesch Reading Ease Score')
plt.xlabel('Flesch Reading Ease Score')
plt.ylabel('Text Rank Summary Length')
plt.legend(title='Topic', bbox_to_anchor=(1.05, 1), loc='upper left',␣
 ↪borderaxespad=0)
plt.tight_layout()
plt.savefig("/Users/marwahfaraj/Desktop/ms_degree_application_and_doc/
 ↪final_projects/502_final_project/BBC_new_topic_classification/image/
 ↪text_rank_summary_length_vs_flesch_reading_ease_score.jpg",␣
 ↪bbox_inches='tight')
plt.show()

plt.figure(figsize=(10, 6))
sns.scatterplot(x='Dale-Chall Readability Score', y='lsa_summary_length',␣
 ↪hue='labels', data=data, palette=custom_palette)
plt.title('LSA Summary Length vs Dale-Chall Readability Score')
plt.xlabel('Dale-Chall Readability Score')
plt.ylabel('LSA Summary Length')
plt.legend(title='Topic', bbox_to_anchor=(1.05, 1), loc='upper left',␣
 ↪borderaxespad=0)
plt.tight_layout()
plt.savefig("/Users/marwahfaraj/Desktop/ms_degree_application_and_doc/
 ↪final_projects/502_final_project/BBC_new_topic_classification/image/
 ↪LSA_Summary_Length_vs_Dale-Chall_Readability_Score.jpg", bbox_inches='tight')
plt.show()
```

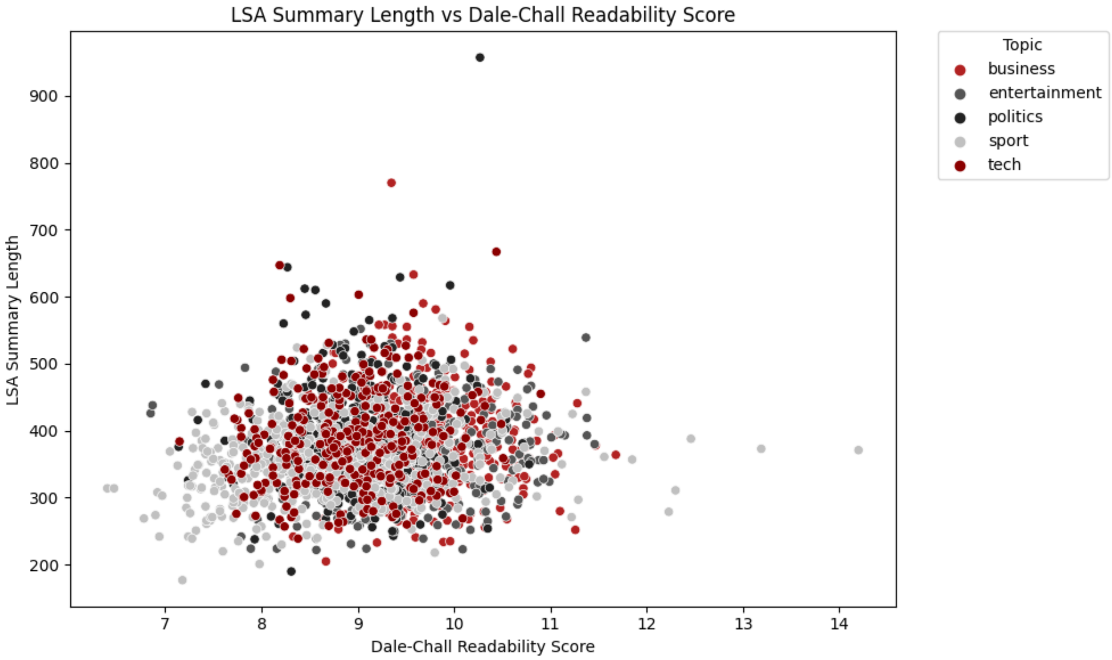

Average Flesch Reading Ease Score by Topic
Flesch Reading Ease Score
0
10
20
30
40
50
60
70
business
entertainment
politics
sport
tech
Topic


Article Length (No. of Sentences) vs Flesch Reading Ease Score
Flesch Reading Ease Score
40
50
60
70
80
90
0
50
100
150
200
250
No. of Sentences
Topic
business
entertainment
politics
sport
tech


Average Text Rank Summary Length by Topic
Text Rank Summary Length
0
100
200
300
400
500
business
entertainment
politics
sport
tech
Topic


Average LSA Summary Length by Topic
LSA Summary Length
0
50
100
150
200
250
300
350
400
business
entertainment
politics
sport
tech
Topic


Text Rank Summary Length vs Flesch Reading Ease Score
Text Rank Summary Length
2250
2000
1750
1500
1250
1000
750
500
250
40
50
60
70
80
90
Flesch Reading Ease Score
Topic
● business
● entertainment
● politics
● sport
● tech


LSA Summary Length vs Dale-Chall Readability Score
LSA Summary Length
900
800
700
600
500
400
300
200
7
8
9
10
11
12
13
14
Dale-Chall Readability Score
Topic
● business
● entertainment
● politics
● sport
● tech

```
[ ]:
```