In [1]:

```
import unicodedata
import re
import pandas as pd
import numpy as np
import string
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import NMF
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import seaborn as sns

# Load the CSV file into a DataFrame
csv_file_path = r'C:\Users\abcd\Desktop\Jupyter data\Raw data\Revised\2016_2020_final.csv'
data = pd.read_csv(csv_file_path, encoding='ISO-8859-1')

# Set up stopwords
stop_words = set(stopwords.words('english'))

# Define custom stopwords
custom_stop_words = {'chilika', 'lagoon', 'japanese','gps', 'asia', 'rural', 'pacific', 'small', 'tarawa', 'sess', 'east', 'so

# Combine custom and built-in stop words
all_stop_words = list(stop_words) + list(custom_stop_words)

# Create TF-IDF matrix
corpus = data['processed_Combined']
vectorizer = TfidfVectorizer(min_df=1, max_features=2000, stop_words=all_stop_words)  # Adjust min_df and max_features as neea
X = vectorizer.fit_transform(corpus)


# Set the maximum number of clusters to evaluate
max_num_clusters = 25

# Set the desired maximum number of iterations
```

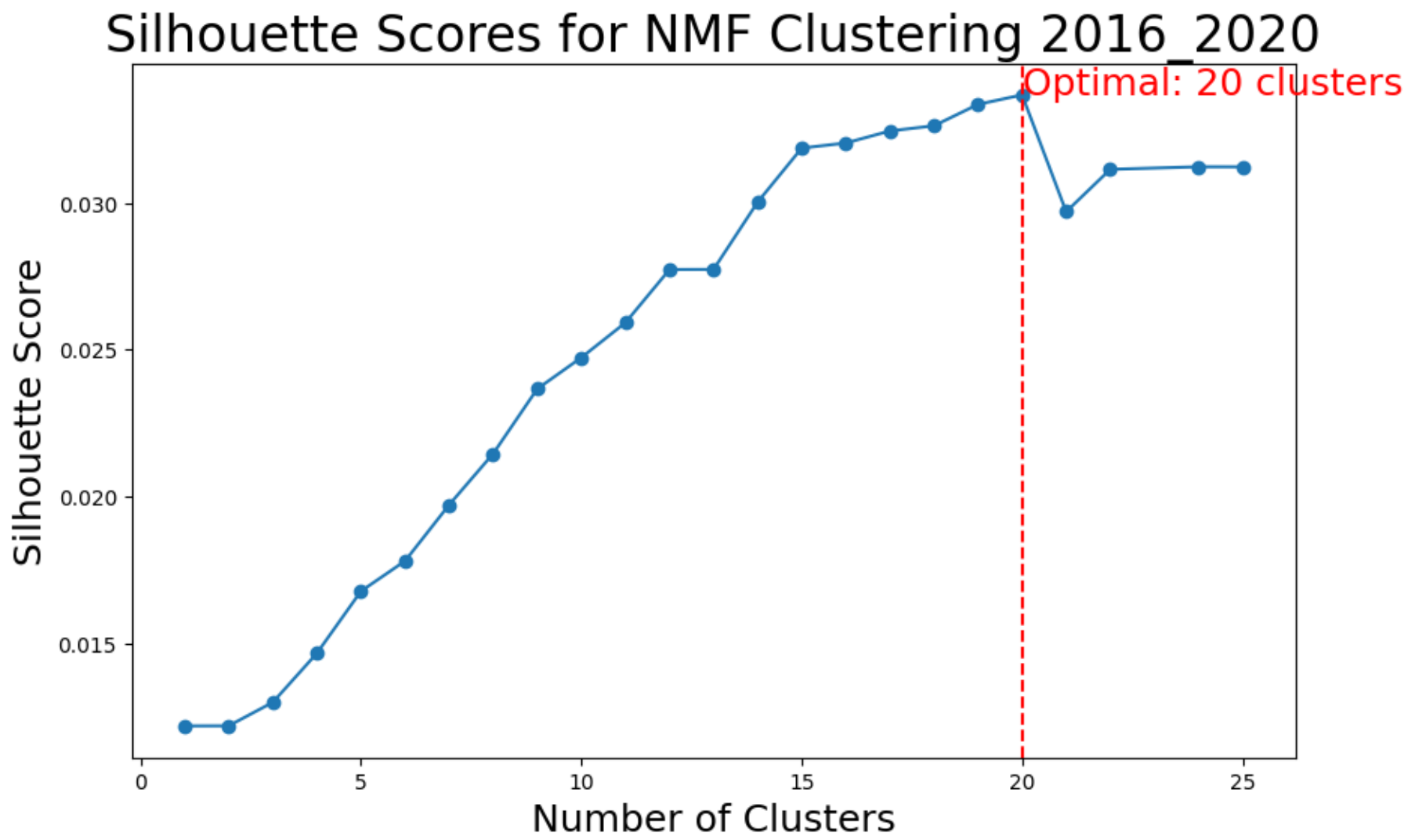
```
max_iter = 1000

# Initialize lists to store the silhouette scores and filtered number of topics
silhouette_scores = []
filtered_num_topics_list = []

# Regularization parameter (l1_ratio) for NMF sparsity regularization
l1_ratio = 0.9 # You can experiment with different values between 0 and 1

# Iterate over the topic range and calculate the silhouette score for each number of topics
for num_topics in range(2, max_num_clusters + 1):
    # Fit NMF model with the current number of topics and apply sparsity regularization
    nmf = NMF(n_components=num_topics, max_iter=max_iter, random_state=42, l1_ratio=l1_ratio)
    nmf.fit(X)

    # Perform NMF with the current number of clusters
    cluster_labels = nmf.transform(X).argmax(axis=1)

    # Calculate the number of documents associated with each topic
    topic_counts = np.bincount(cluster_labels)

    # Filter out topics that have fewer documents than the threshold
    min_documents_threshold = 30
    filtered_topics = [topic_idx for topic_idx, count in enumerate(topic_counts) if count >= min_documents_threshold]

    # Calculate the silhouette score only if the number of filtered clusters is greater than 1
    if len(filtered_topics) > 1:
        silhouette_avg = silhouette_score(X, cluster_labels)
        silhouette_scores.append(silhouette_avg)
        filtered_num_topics_list.append(num_topics)

                # Create a DataFrame to store the silhouette scores
silhouette_df = pd.DataFrame({'Num_Clusters': filtered_num_topics_list, 'Silhouette_Score': silhouette_scores})

# Save the DataFrame to an Excel file
silhouette_df.to_excel(r'C:\Users\abcd\Desktop\Jupyter data\Raw data\Final\silhouette_scores_2016_2020.xlsx', index=False)

        # Plot the silhouette scores
#plt.plot(filtered_num_topics_list, silhouette_scores, marker='o')

# Find the "elbow" point
```

```
if silhouette_scores:
    elbow_index = np.argmax(silhouette_scores) + 2
    elbow_score = silhouette_scores[elbow_index - 2]
else:
    # Set default values if the silhouette_scores list is empty
    elbow_index = 2
    elbow_score = 0

# Add a vertical line at the elbow point
#plt.axvline(x=elbow_index, linestyle='--', color='red')

# Find the optimal number of topics based on the maximum silhouette score
optimal_num_topics = filtered_num_topics_list[np.argmax(silhouette_scores)]

# Annotate the elbow point on the plot
#plt.annotate(f'Optimal: {optimal_num_topics} clusters', xy=(optimal_num_topics, max(silhouette_scores)),
#             xytext=(optimal_num_topics, max(silhouette_scores) + 0), color='red')

#plt.xlabel('Number of Clusters')
#plt.ylabel('Silhouette Score')
#plt.title('Silhouette Scores for NMF Clustering 2016_2020')
#plt.show()
```

In [2]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Read the Excel file
file_path = r'C:\Users\abcd\Desktop\Jupyter data\Raw data\Final\silhouette_scores_2016_2020_1.xlsx'
df = pd.read_excel(file_path)

# Plot the data
plt.figure(figsize=(10, 6))
plt.plot(df['Num_Clusters'], df['Silhouette_Score'], marker='o')
plt.xlabel('Number of Clusters', fontsize=18)
plt.ylabel('Silhouette Score', fontsize=18)
plt.title('Silhouette Scores for NMF Clustering 2016_2020', fontsize=24)

# Annotate the optimal point
optimal_num_clusters = 20  # Change this to the actual optimal number of clusters
plt.annotate(f'Optimal: {optimal_num_clusters} clusters', xy=(optimal_num_clusters, max(df['Silhouette_Score'])),
```

```
            xytext=(optimal_num_clusters, max(df['Silhouette_Score']) + 0), color='red', fontsize=18)

# Add a vertical line at the optimal point
plt.axvline(x=optimal_num_clusters, linestyle='--', color='red')

plt.show()
```



In [3]:

```
# Fit NMF model with the optimal number of topics and apply sparsity regularization
num_topics = optimal_num_topics
nmf = NMF(n_components=num_topics, max_iter=max_iter, random_state=42, l1_ratio=l1_ratio)
nmf.fit(X)

# Get the top terms for each topic
feature_names = vectorizer.get_feature_names_out()
top_terms_dict = {}
for topic_idx, topic in enumerate(nmf.components_):
    top_terms = [feature_names[i] for i in topic.argsort()[:-10:-1]]
    top_terms_dict[topic_idx] = top_terms
    print(f"C {topic_idx + 1}: {top_terms}\n")

# Transform documents to topic distribution
document_topics = nmf.transform(X)

# Get the document-topic matrix from NMF
document_topics = nmf.transform(X)

# Find the dominant topic for each document
dominant_topics = np.argmax(document_topics, axis=1)

# Count the occurrences of each topic label
topic_counts = np.bincount(dominant_topics)

# Print the number of documents in each cluster
for cluster_id, count in enumerate(topic_counts, start=1):
    print(f"C {cluster_id}: {count} articles")
```

C 1: ['research', 'transdisciplinary', 'project', 'knowledge', 'stakeholder', 'process', 'researcher', 'collaboration', 'problem']

C 2: ['climate', 'adaptation', 'change', 'risk', 'uncertainty', 'assessment', 'model', 'sensitivity', 'vulnerability']

C 3: ['value', 'social', 'relational', 'valuation', 'intrinsic', 'individual', 'concept', 'economics', 'people']

C 4: ['scenario', 'future', 'model', 'positive', 'population', 'alternative', 'land', 'ecosystem', 'change']

C 5: ['sdgs', 'development', 'goal', 'sustainable', 'indicator', 'target', 'progress', 'national', 'implementation']

C 6: ['sustainability', 'science', 'system', 'transition', 'digital', 'discipline', 'society', 'research', 'concept']

C 7: ['water', 'governance', 'nexus', 'supply', 'management', 'resource', 'demand', 'system', 'decision']

C 8: ['landscape', 'agroforestry', 'management', 'tree', 'conservation', 'system', 'mediterranean', 'land', 'policy']

C 9: ['service', 'ecosystem', 'forest', 'capital', 'natural', 'assessment', 'biodiversity', 'payment', 'ecological']

C 10: ['blue', 'growth', 'economy', 'fishery', 'degrowth', 'marine', 'coastal', 'smallscale', 'economic']

C 11: ['place', 'meaning', 'sense', 'transformative', 'attachment', 'stewardship', 'transformation', 'narrative', 'transition']

C 12: ['cultural', 'selection', 'multilevel', 'evolution', 'evolutionary', 'group', 'institution', 'grouplevel', 'resource']

C 13: ['indigenous', 'knowledge', 'science', 'local', 'community', 'western', 'river', 'traditional', 'protocol']

C 14: ['food', 'household', 'security', 'consumption', 'production', 'crop', 'healthy', 'health', 'sustainable']

C 15: ['delta', 'vulnerability', 'region', 'amazon', 'coastal', 'population', 'change', 'risk', 'flood']

C 16: ['conflict', 'environmental', 'justice', 'movement', 'social', 'distribution', 'injustice', 'violence', 'metabolism']

C 17: ['resilience', 'community', 'capital', 'social', 'natural', 'adaptive', 'disturbance', 'capacity', 'framework']

C 18: ['urban', 'city', 'myth', 'public', 'citizen', 'data', 'governance', 'vision', 'experiment']

C 19: ['education', 'future', 'program', 'competency', 'educational', 'student', 'curriculum', 'sustainable', 'practice']

C 20: ['trap', 'human', 'socialecological', 'policy', 'system', 'response', 'dynamic', 'ecological', 'model']

```
C 1: 24 articles
C 2: 21 articles
C 3: 26 articles
C 4: 17 articles
C 5: 34 articles
C 6: 55 articles
C 7: 14 articles
C 8: 21 articles
C 9: 19 articles
C 10: 18 articles
C 11: 22 articles
C 12: 11 articles
C 13: 23 articles
C 14: 28 articles
C 15: 15 articles
C 16: 24 articles
C 17: 25 articles
C 18: 17 articles
C 19: 28 articles
C 20: 16 articles
```

In [4]:

```python
from collections import Counter

# Flatten the top terms from each topic
all_top_terms = [term for terms in top_terms_dict.values() for term in terms]

# Count the frequency of each top term
term_counts = Counter(all_top_terms)

# Print the top terms for each topic along with their frequencies
for topic_idx, top_terms in top_terms_dict.items():
    sorted_terms = sorted(top_terms, key=lambda term: term_counts[term], reverse=True)
    term_frequency = [f"{term} ({term_counts[term]})" for term in sorted_terms]
    print(f"Topic {topic_idx + 1}: {', '.join(term_frequency)}\n")
```

Topic 1: research (2), knowledge (2), transdisciplinary (1), project (1), stakeholder (1), process (1), researcher (1), collaboration (1), problem (1)

Topic 2: change (3), model (3), risk (2), assessment (2), vulnerability (2), climate (1), adaptation (1), uncertainty (1), sensitivity (1)

Topic 3: social (3), concept (2), value (1), relational (1), valuation (1), intrinsic (1), individual (1), economics (1), people (1)

Topic 4: model (3), change (3), future (2), population (2), land (2), ecosystem (2), scenario (1), positive (1), alternative (1)

Topic 5: sustainable (3), sdgs (1), development (1), goal (1), indicator (1), target (1), progress (1), national (1), implementation (1)

Topic 6: system (4), science (2), transition (2), research (2), concept (2), sustainability (1), digital (1), discipline (1), society (1)

Topic 7: system (4), governance (2), management (2), resource (2), water (1), nexus (1), supply (1), demand (1), decision (1)

Topic 8: system (4), management (2), land (2), policy (2), landscape (1), agroforestry (1), tree (1), conservation (1), mediterranean (1)

Topic 9: ecosystem (2), capital (2), natural (2), assessment (2), ecological (2), service (1), forest (1), biodiversity (1), payment (1)

Topic 10: coastal (2), blue (1), growth (1), economy (1), fishery (1), degrowth (1), marine (1), smallscale (1), economic (1)

Topic 11: transition (2), place (1), meaning (1), sense (1), transformative (1), attachment (1), stewardship (1), transformation (1), narrative (1)

Topic 12: resource (2), cultural (1), selection (1), multilevel (1), evolution (1), evolutionary (1), group (1), institution (1), grouplevel (1)

Topic 13: knowledge (2), science (2), community (2), indigenous (1), local (1), western (1), river (1), traditional (1), protocol (1)

Topic 14: sustainable (3), food (1), household (1), security (1), consumption (1), production (1), crop (1), healthy (1), health (1)

Topic 15: change (3), vulnerability (2), coastal (2), population (2), risk (2), delta (1), region (1), amazon (1), flood (1)

```
Topic 16: social (3), conflict (1), environmental (1), justice (1), movement (1), distribution (1), injustice (1), violence
(1), metabolism (1)

Topic 17: social (3), community (2), capital (2), natural (2), resilience (1), adaptive (1), disturbance (1), capacity (1), f
ramework (1)

Topic 18: governance (2), urban (1), city (1), myth (1), public (1), citizen (1), data (1), vision (1), experiment (1)

Topic 19: sustainable (3), future (2), education (1), program (1), competency (1), educational (1), student (1), curriculum
(1), practice (1)

Topic 20: system (4), model (3), policy (2), ecological (2), trap (1), human (1), socialecological (1), response (1), dynamic
(1)
```

In [ ]: