```
from sklearn.datasets import fetch_20newsgroups

# Load a subset of the 20 newsgroups dataset (choosing 4 categories)
categories = ['rec.sport.baseball', 'comp.graphics', 'sci.space', 'talk.politics.misc']
newsgroups = fetch_20newsgroups(subset='all', categories=categories, remove=('headers', 'footers', 'quotes'))

# Display number of documents and categories
print(f"Number of documents: {len(newsgroups.data)}")
print(f"Categories: {newsgroups.target_names}")
```

```
Number of documents: 3729
Categories: ['comp.graphics', 'rec.sport.baseball', 'sci.space', 'talk.politics.misc']
```

```
from sklearn.feature_extraction.text import TfidfVectorizer

# Create a TF-IDF vectorizer
vectorizer = TfidfVectorizer(stop_words='english', max_df=0.5, min_df=2)

# Transform the dataset into TF-IDF features
tfidf_matrix = vectorizer.fit_transform(newsgroups.data)

# Show the shape of the TF-IDF matrix
print(f"TF-IDF matrix shape: {tfidf_matrix.shape}")
```

```
TF-IDF matrix shape: (3729, 18141)
```

```
import numpy as np

# Identify rows in the TF-IDF matrix where the sum of terms is 0 (i.e., zero vectors)
non_zero_indices = np.array(tfidf_matrix.sum(axis=1)).flatten() > 0

# Filter out zero-vector documents
tfidf_matrix_non_zero = tfidf_matrix[non_zero_indices]

# Get the corresponding non-zero documents
filtered_documents = [newsgroups.data[i] for i in range(len(newsgroups.data)) if non_zero_indices[i]]

# Check the new shape of the TF-IDF matrix
print(f"Filtered TF-IDF matrix shape: {tfidf_matrix_non_zero.shape}")
```

```
Filtered TF-IDF matrix shape: (3607, 18141)
```

```
from sklearn.cluster import AgglomerativeClustering

# Perform Agglomerative Clustering on the filtered TF-IDF matrix (choosing 4 clusters)
agg_cluster = AgglomerativeClustering(n_clusters=4, metric='cosine', linkage='average')

# Fit the clustering model
agg_cluster.fit(tfidf_matrix_non_zero.toarray())

# Display the cluster labels for the first 10 filtered documents
cluster_labels = agg_cluster.labels_
print(f"Cluster labels for the first 10 filtered documents: {cluster_labels[:10]}")
```

```
Cluster labels for the first 10 filtered documents: [0 0 0 0 0 0 0 0 0 0]
```

```
from sklearn.metrics.pairwise import cosine_distances
import scipy.cluster.hierarchy as sch

# Compute the cosine distance matrix for the filtered TF-IDF matrix
filtered_distance_matrix = cosine_distances(tfidf_matrix_non_zero)

# Compute the linkage matrix using the average linkage method
linkage_matrix = sch.linkage(filtered_distance_matrix, method='average')

# Check the number of filtered documents (should match the number of labels)
print(f"Number of filtered documents: {len(filtered_documents)}")
```

```
<ipython-input-16-8cb0e8a0766d>:8: ClusterWarning: The symmetric non-negative hollow observation matrix looks suspiciously like an uncon
  linkage_matrix = sch.linkage(filtered_distance_matrix, method='average')
```

```
Number of filtered documents: 3607
```

```
# Count the number of documents in each cluster
unique, counts = np.unique(cluster_labels, return_counts=True)
cluster_sizes = dict(zip(unique, counts))
print(f"Cluster Sizes: {cluster_sizes}")
```

```
Cluster Sizes: {0: 3604, 1: 1, 2: 1, 3: 1}
```

```
# Print documents from the first cluster
for i, doc in enumerate(filtered_documents[:10], start=1):
    if cluster_labels[i] == 0:  # Replace 0 with the desired cluster number
        print(f"\nDocument {i}: {doc[:100]}...\n")  # Print first 200 characters
```

```
Document 1:
Are you kidding?  I'm stuck with the Toronto SkyDome, where their idea
of a 7th inning stretch is t...


Document 2:




Sigh.  You're absolutely right.  We have no political power whatsoever.
Therefore, we should ...


Document 3:

 The above statement ignores reality. The BD WERE provoked.



 Damn, Phil. You must have seen a ...


Document 4:
I'm sorry about your friend.  Really.  But this anecdote does nothing to
justify the "war on drugs"...


Document 5: Hank Greenberg was probably the greatest ever.  He was also subject to a
lot of heckling from bigots...


Document 6:
Scott,
        I'm not so sure if this is helpful, but I usually use XV v2.21.  I use Sun IPCs and IPXs, a...


Document 7:
to take this to its, er, "logical" conclusion, it is impossible to
ascertain whether or not i am a ...


Document 8: Suppose the Soviets had managed to get their moon rocket working
and had made it first.  They could ...


Document 9:
Hank Greenberg, Sid Gordon, Ron Blomberg....


Document 10:
Peter Nelson posted a very eloquent response to this point in
talk.politics.misc, so I need not co...
```