Test Plan and Results

Team Semantics
Tristan Weger, Ryan Rubadue, Yahya Emara

Overall Test Plan

Our approach to testing was based on three areas of testing. First, testing each component of the project on its own. Second, testing each component using both sentence tokens and segments. Third, is testing the entire code as well as normal and abnormal cases and inputs into the tests. The combination of these three areas of testing should allow us to see any abnormalities or errors in our code and promptly fix them to help improve reliability of the system. Relative performance of the system is not a concern, so our test cases are designed to focus on producing accurate results.

Test Case Descriptions

Test each component in process of analyzing each document
Test overall result of document analysis

1.1 **Tokenizing Test 1**
1.2 This test will ensure the sentence tokenization process is functional
1.3 5 different transcripts of presidents each 50 years apart is used to check results
1.4 Inputs: Full transcripts of unstructured text data
1.5 Outputs: Tokenized sentences
1.6 Normal
1.7 Blackbox
1.8 Functional
1.9 Unit Test
1.10 Results: All sentences were correctly tokenized and no sentences were missed, test passed
2.1 **Filtering Test 1**
2.2 This test will ensure the sentence are relevant and provide information
2.3 5 different transcripts of presidents each 50 years apart is used to check results
2.4 Inputs: Tokenized sentences of each transcript
2.5 Outputs: Filtered relevant sentences and removed stop words from each sentence
2.6 Normal
2.7 Blackbox
2.8 Functional

2.9 Unit Test
2.10 Results: Relevant sentences were extracted from the dataset without stop words, test passed
3.1 **Segmentation Test 1**
3.2 This test will ensure that related sentences are being grouped together correctly
3.3 5 different transcripts of presidents each 50 years apart is used to check results
3.4 Inputs: Filtered sentence tokens
3.5 Outputs: Segments of logically related sentences
3.6 Normal
3.7 Blackbox
3.8 Functional
3.9 Unit Test
3.10 Results: Segments produced are of logically related sentences, test passed

4.1 **Cosine Similarity Test 1**
4.2 This test will ensure that cosine similarity function between both sentences and segments is functional
4.3 5 different transcripts of presidents each 50 years apart is used to check results for both sentences and segments
4.4 Inputs: filtered tokenized sentences/segments
4.5 Outputs: Cosine similarity matrix that is size nxn where is n is number of sentences/segments
4.6 Normal
4.7 Blackbox
4.8 Functional
4.9 Unit Test
4.10 Cosine similarity matrix produced correctly, test passed

5.1 **Embedding Test 1**
5.2 This test verifies the program can correctly embed individual sentences within the data
5.3 5 different transcripts of presidents each 50 years apart is used to check results for both sentences and segments.
5.4 Inputs: List containing sentences present within the transcript (either filtered or unfiltered)
5.5 Outputs: A list of numerical embedding values that correspond the the sentences analyzed.
5.6 Normal
5.7 Blackbox
5.8 Functional
5.9 Unit Test
5.10 If a list of embeddings that correctly corresponds to the input list of sentences is created, the test is successful
6.1 **Embedding Test 2**
6.2 This test is similar to Embedding Test 1, but splits the data by paragraphs instead of sentences

6.3 5 different transcripts of presidents each 50 years apart is used to check results for both sentences and segments
6.4 Inputs: List of paragraphs with the transcripts
6.5 Outputs: A list of numerical embeddings that correspond to each paragraph
6.6 Normal
6.7 Blackbox
6.8 Functional
6.9 Unit Test
6.10 Successful if a correctly formatted list of numerical embeddings is produces the size of the paragraphs read

7.1 **Heatmap Test 1: Pre-PCA**
7.2 This test will ensure the pre principal component analysis heatmap is correct.
7.3 A shortened 5x5 heatmap will be used to easily check and ensure accuracy.
7.4 Inputs: pre principal component analysis cosine similarity matrix.
7.5 Outputs: color map determined by the range of the cosine similarity matrix.
7.6 Normal
7.7 Blackbox
7.8 Functional
7.9 Unit Test
7.10 Results: Heatmap color scale aligned with data, test passed.

8.1 **Heatmap Test 2: Post-PCA**
8.2 This test will ensure the post principal component analysis heatmap is correct.
8.3 A shortened 5x5 heatmap will be used to easily check and ensure accuracy.
8.4 Inputs: post principal component analysis cosine similarity matrix.
8.5 Outputs: color map determined by the range of the cosine similarity matrix.
8.6 Normal
8.7 Blackbox
8.8 Functional
8.9 Unit Test
8.10 Results: Heatmap color scale aligned with data, test passed.

9.1 **Heatmap Test 3: Pre-PCA Normalized**
9.2 This test will ensure the normalized pre principal component analysis heatmap is correct.
9.3 A shortened 5x5 heatmap will be used to easily check and ensure accuracy.
9.4 Inputs: pre normalized principal component analysis cosine similarity matrix.
9.5 Outputs: color map determined by the range of the cosine similarity matrix.
9.6 Normal
9.7 Blackbox
9.8 Functional
9.9 Unit Test
9.10 Results: Heatmap color scale aligned with data, test passed.

10.1 **Heatmap Test 4: Post-PCA Normalized**
10.2 This test will ensure the normalized principal component analysis heatmap is correct.
10.3 A shortened 5x5 heatmap will be used to easily check and ensure accuracy.
10.4 Inputs: normalized post principal component analysis cosine similarity matrix.
10.5 Outputs: color map determined by the range of the cosine similarity matrix.
10.6 Normal
10.7 Blackbox
10.8 Functional
10.9 Unit Test
10.10 Results: Heatmap color scale aligned with data, test passed.

11.1 **PCA95 Test 1**
11.2 This test will ensure the principal component analysis retains sentence meaning.
11.3 Samples of heat maps will be randomly overlaid to check for PCA accuracy.
11.4 Inputs: pre-PCA and Post-PCA heatmap samples.
11.5 Outputs: Comparison pass or fail depending on similarities of data.
11.6 Normal
11.7 Blackbox
11.8 Functional
11.9 Unit Test
11.10 Results: Heatmap samples showed similar hotspots resulting in test pass.

12.1 **PCA95 Test 2**
12.2 This test will ensure principle component analysis is reducing dimensionality.
12.3 Multiple data sets will be input and post PCA matrix shapes are recorded.
12.4 Inputs: different presidential speech datasets.
12.5 Outputs: post PCA matrix dimensions.
12.6 Normal
12.7 Blackbox
12.8 Functional
12.9 Unit Test
12.10 Results: Dimensionality reduced by about 1/3 , test is passed.

13.1 **Clustering Test 1**
13.2 This test determines whether our clustering method (K-Means) produces relatively correct clusters
13.3 At least 3 sets of data will be used to check results
13.4 Inputs: 2 dimensional embedding data
13.5 Outputs: List of cluster numbers corresponding to each data point
13.6 Normal
13.7 Functional
13.8 Blackbox
13.9 Unit

13.10 Results: List of cluster labels. The test is passed if sentences under the same level seem to be related to each other.

14.1 **Clustering Test 2**
14.2 Determines whether number of chosen clusters is suitable for data
14.3 At least 3 sets of data will be used to check results
14.4 Inputs: 2 dimensional embedding data
14.5 Outputs: Line graph corresponding to 'Sum of Squares' calculation for different number of clusters
14.6 Normal
14.7 Performance
14.8 Blackbox
14.9 Unit
14.10 Results: Line graph corresponding to sum of squares value for different cluster sizes. The test passes if the cluster size corresponds to a positive sum of squares value.

15.1 **Semantic Analysis of Entire Corpus 1**
15.2 Verifies semantic analysis runs correctly across entire dataset during a single execution
15.3 The entire dataset will be used to verify results
15.4 Inputs: Text Corpus
15.5 Outputs: Semantic Analysis of all documents within corpus
15.6 Abnormal
15.7 Blackbox
15.8 Functional
15.9 Integration
15.10 Results: Successful test corresponds to producing semantic analysis of all documents and execution passing under stress load of processing all documents.

Test Case Matrix

	Normal/ Abnormal	**Blackbox/ Whitebox**	**Functional/ Performance**	**Unit/ Integration**
1.1	Normal	Blackbox	Functional	Unit
2.2	Normal	Blackbox	Functional	Unit
3.1	Normal	Blackbox	Functional	Unit
4.1	Normal	Blackbox	Functional	Unit
5.1	Normal	Blackbox	Functional	Unit

6.1	Normal	Blackbox	Functional	Unit
7.1	Normal	Blackbox	Functional	Unit
8.1	Normal	Blackbox	Functional	Unit
9.1	Normal	Blackbox	Functional	Unit
10.1	Normal	Blackbox	Functional	Unit
11.1	Normal	Blackbox	Functional	Unit
12.1	Normal	Blackbox	Functional	Unit
13.1	Normal	Blackbox	Functional	Unit
14.1	Normal	Blackbox	Performance	Unit
15.1	Abnormal	Blackbox	Functional	Integration