**NGramGUI.java Testcases**

**Positive GUI Test Scenarios:**

**Test Case 1:**

**Input Search Text:** abcde

**Number of Suggestions specified by the user:** 5

****

**Screenshot 1**

When a user inputs a search string “**abcde**”, specifies the number of suggestions as some random value ‘**4’** and then clicks on the “**Search**” button, the NGram node’s search results are displayed correctly as shown in the left hand side of the above figure.

**Test Case 2: Input Search Text:** abcdef,test

**Number of Suggestions specified by the user:** 4

****

**Screenshot 2**

When a user enters an input search string “**abcdef,test**”, specifies the number of suggestions as some random value ‘**4’** and then clicks on the “**Search**” button, the search results of the NGram store containing two nodes are displayed as indicated in the Left Hand Side of **Screenshot 2**. This is because, the search results of “**abcdef**”are displayed on node 1 and the search result for ‘**test**’ is displayed on node 2.

**Test Case 3: Input Search Text:** abcd, abcd, abcd

**Number of Suggestions specified by the user:** 6

****

**Screenshot 3**

When a user enters an input search string “**abcd, abcd, abcd**”, specifies the number of suggestions as some random value ‘**6’** and then clicks on the “**Search**” button, the corresponding search results of the NGram node are displayed accordingly by checking its redundancy and displaying the node results for only one input as indicated in the Left Hand Side of **Screenshot 3**.

**Test Case 3: Input Search Text:** abcd, abcd, abcd

**Number of Suggestions specified by the user:** 6

****

**Screenshot 4**

When a user enters an input search string “**abcd, abcd, abcd**”, specifies the number of suggestions as some random value ‘**6’** and then clicks on the “**Search**” button, the corresponding search results of the NGram node are displayed accordingly by checking its redundancy and displaying the node results for only one input as indicated in the Left Hand Side of **Screenshot 4**.

**Negative GUI Test Scenarios:**

**Test Case 1: Input Search Text:** abc, test, - -

**Number of Suggestions specified by the user:** 5

****

NODES NOT DISPLAYED

**Screenshot 1**

When a user enters an input search string “**abc, test, - -**“, specifies the number of suggestions as some random value ‘**5’** and then clicks on the “**Search**” button, an error message “**Please input valid search texts**” is displayed on a dialog box and the corresponding search results of the NGram node are also **NOT displayed** as shown in the above **Screenshot 1**.

**Test Case 2: Input Search Text:** #$123, #%

**Number of Suggestions specified by the user:** 4

****

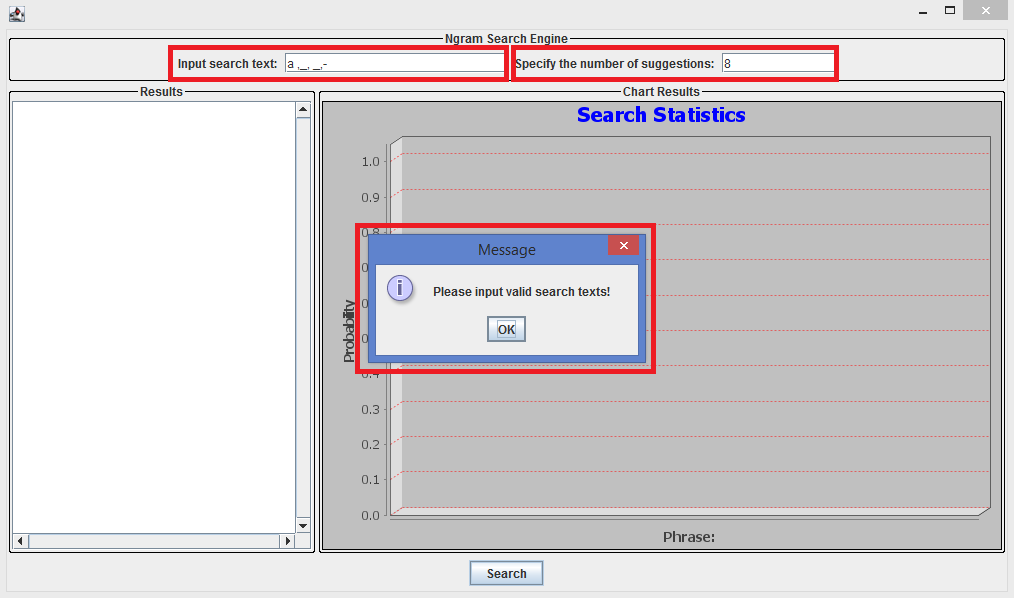
NODES NOT DISPLAYED

**Screenshot 2**

When a user enters an input search string **“#$123, #%**“, specifies the number of suggestions as some random value ‘**5’** and then clicks on the “**Search**” button, an error message “**Please input valid search texts**” is displayed on a dialog box and the corresponding search results of the NGram node are also **NOT displayed** i.e., the store is empty as shown in the above **Screenshot 2**.

**Test Case 3: Input Search Text:** a ,\_, \_,-

**Number of Suggestions specified by the user:** 8

****

NODES NOT DISPLAYED

**Screenshot 3**

When a user enters an input search string “**a ,\_, \_,-**“ specifies the number of suggestions as some random value ‘**8’** and then clicks on the “**Search**” button, an error message “**Please input valid search texts**” is displayed on a dialog box and the corresponding search results of the NGram node are also **NOT displayed** i.e., the store is empty as shown in the above **Screenshot 3**.