**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 ‘**5’** 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 **Screenshot 1** and the probability graph is shown on the NGram GUI.

**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. Also, the probability graph is shown on the NGram GUI.

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

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

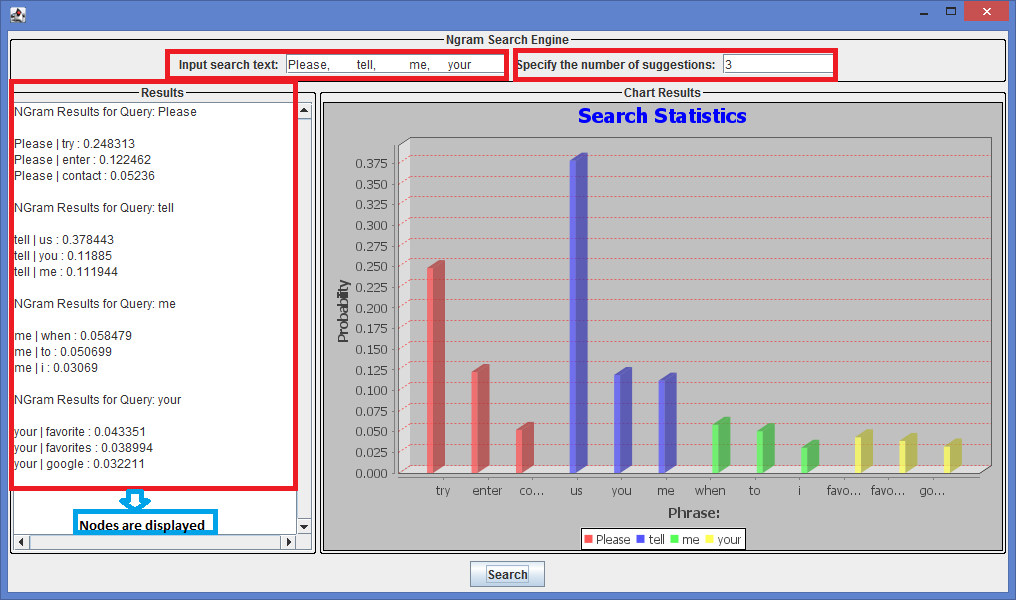
****

**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** and the probability graph is also shown on the NGram GUI.

**Test Case 4: Input Search Text:** Please, tell, me, your

**Number of Suggestions specified by the user:** 3

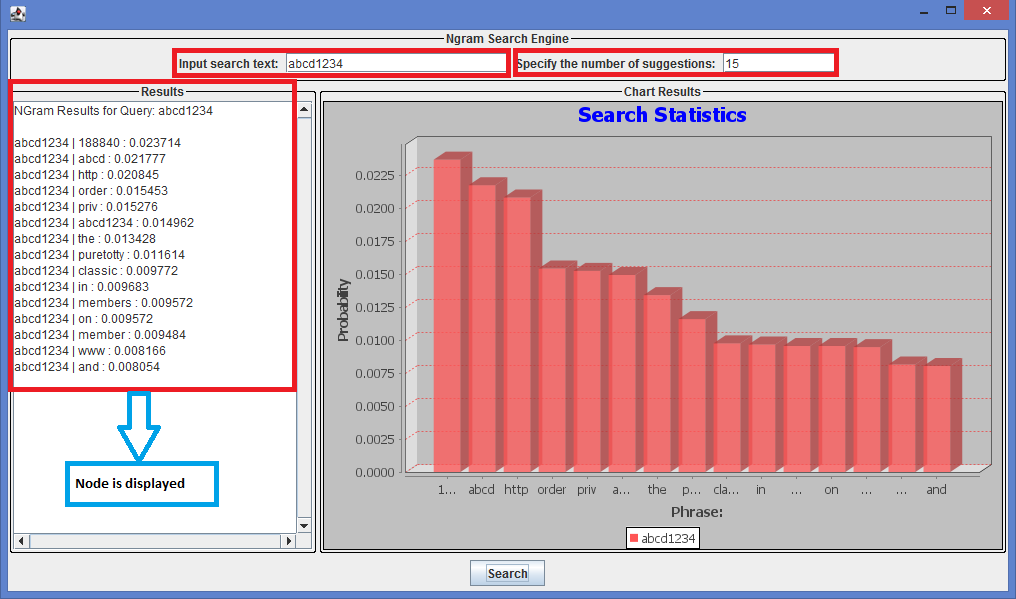
****

**Screenshot 4**

When a user enters an input search string “**Please, tell, me, your**”, specifies the number of suggestions as some random value ‘**3’** and then clicks on the “**Search**” button, the corresponding three search results of the NGram node are displayed correctly inside the store as indicated in the Left Hand Side of **Screenshot 4** and the probability graph is also shown on the NGram GUI.

**Test Case 5: Input Search Text:** abcd1234

**Number of Suggestions specified by the user:** 15

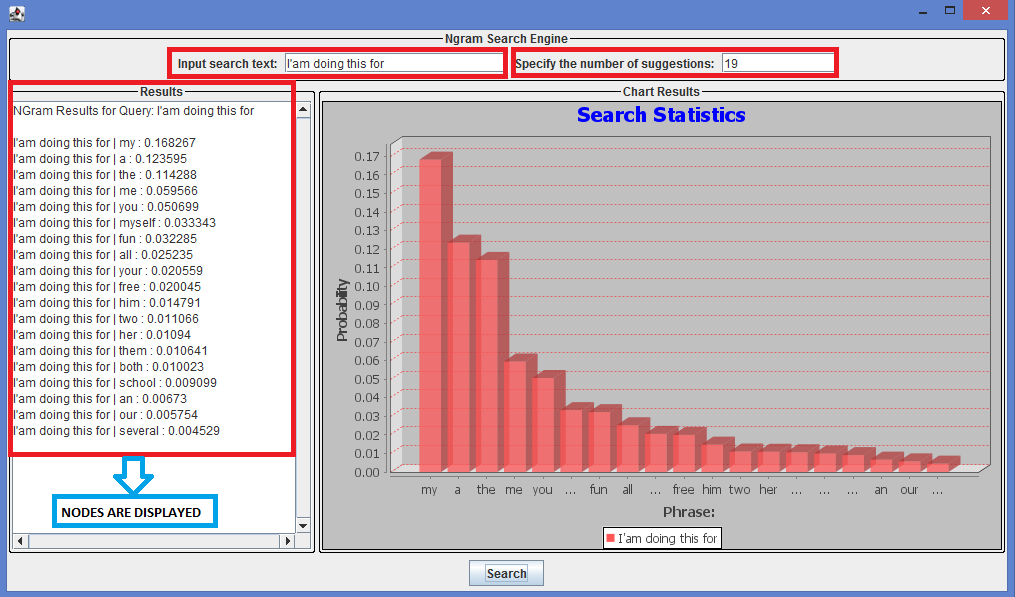
****

**Screenshot 5**

When a user inputs a search string “**abcd1234**”, specifies the number of suggestions as some random value ‘**15’** and then clicks on the “**Search**” button, the NGram node’s 15 search results are displayed correctly as shown in the left hand side of the above **Screenshot 5** and the probability graph is also shown on the NGram GUI.

**Test Case 6: Input Search Text:** I’am doing this for

**Number of Suggestions specified by the user:** 23

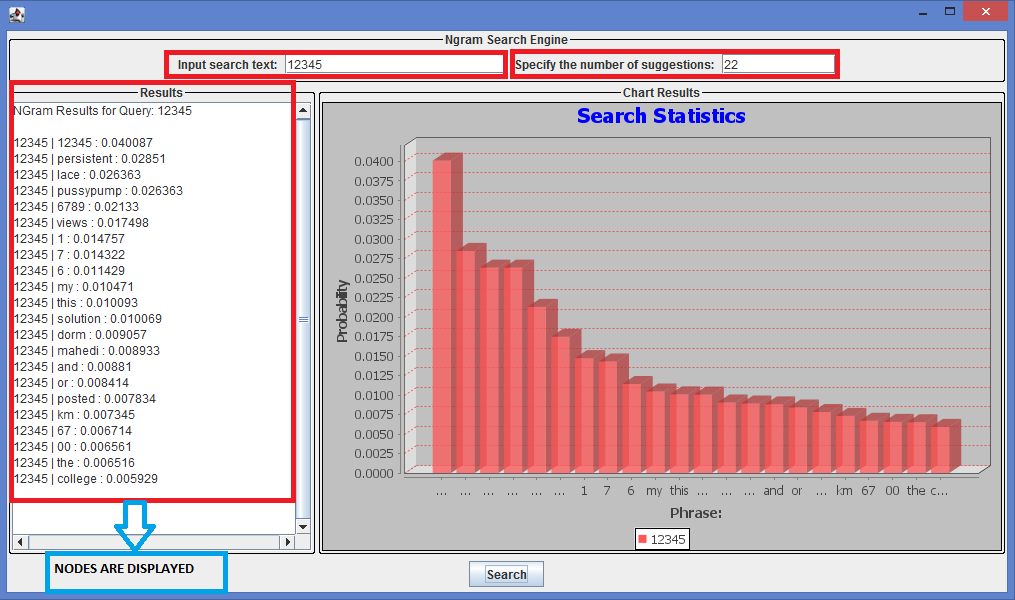


**Screenshot 6**

When a user inputs a search string “**I’am doing this for**”, specifies the number of suggestions as some random value ‘**19’** and then clicks on the “**Search**” button, the NGram node’s 19 search results are displayed correctly as shown in the left hand side of the above **Screenshot 6** and the probability graph is also shown on the NGram GUI.

**Test Case 7: Input Search Text:** 12345

**Number of Suggestions specified by the user:** 22

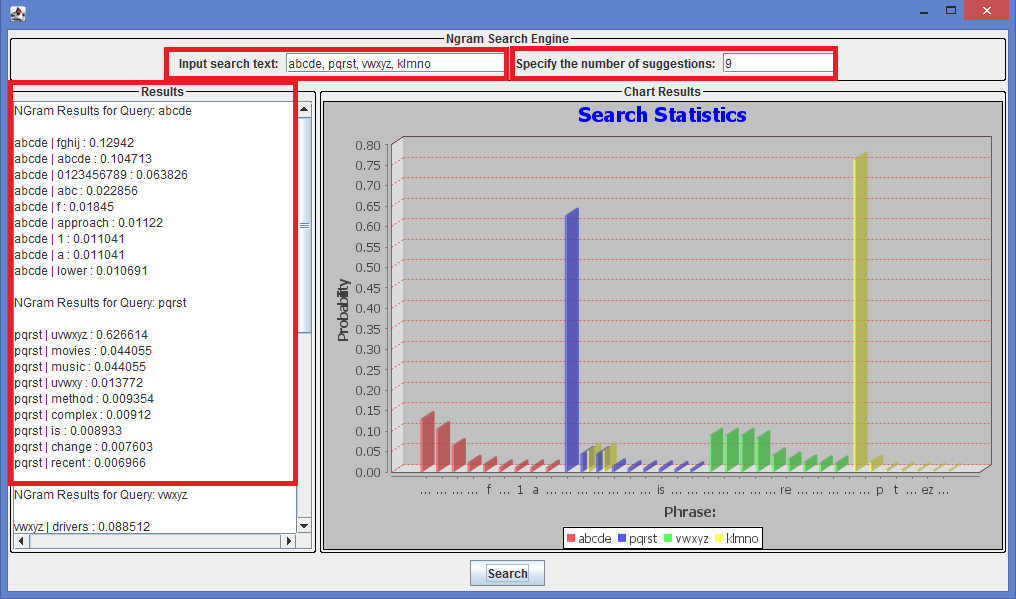


**Screenshot 7**

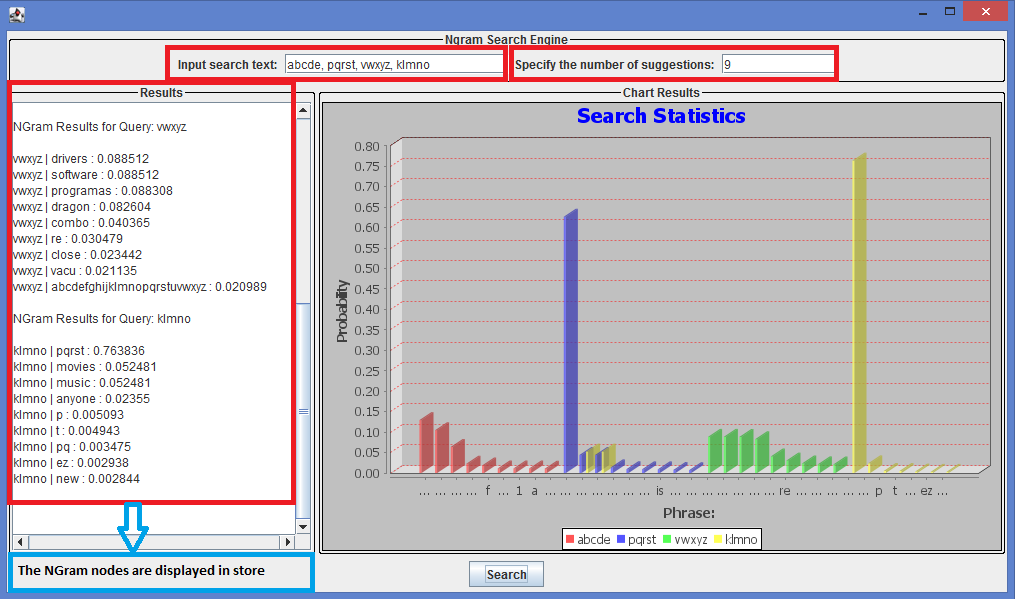
When a user inputs a search string “**12345**”, specifies the number of suggestions as some random value ‘**22’** and then clicks on the “**Search**” button, the NGram node’s 22 search results are displayed correctly as shown in the left hand side of the above **Screenshot 7** and the probability graph is also shown on the NGram GUI.

**Test Case 8: Input Search Text:** abcde, pqrst, vwxyz, klmno

**Number of Suggestions specified by the user:** 9



**Screenshot 8**

****

**Screenshot 9**

When a user inputs a search string “**abcde, pqrst, vwxyz, klmno**”, specifies the number of suggestions as some random value ‘**9’** and then clicks on the “**Search**” button, the NGram search results having 9 nodes are displayed correctly inside the store by use of a scrollbar as shown in the left hand side of the above **Screenshot 8** and **Screenshot 9** and the probability graph is also shown on the NGram GUI.

**Negative GUI Test Scenarios:**

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

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

****

NODES NOT DISPLAYED

**Screenshot 1**

When a user enters an input search string “**abc, test, - -**“, specifies the number of suggestions as some random value ‘**4’** 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

****

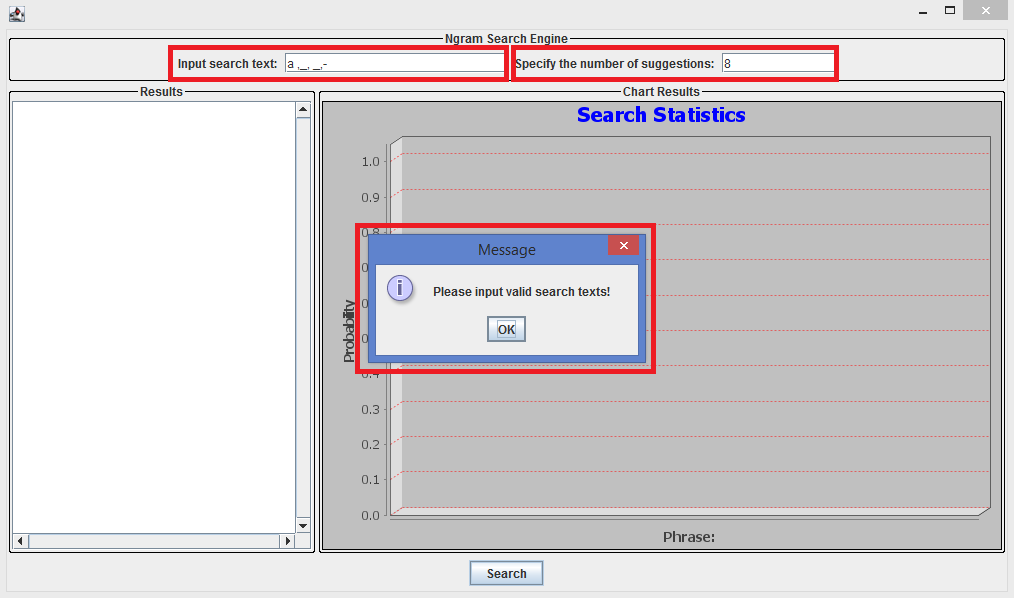
NODES NOT DISPLAYED

**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

****

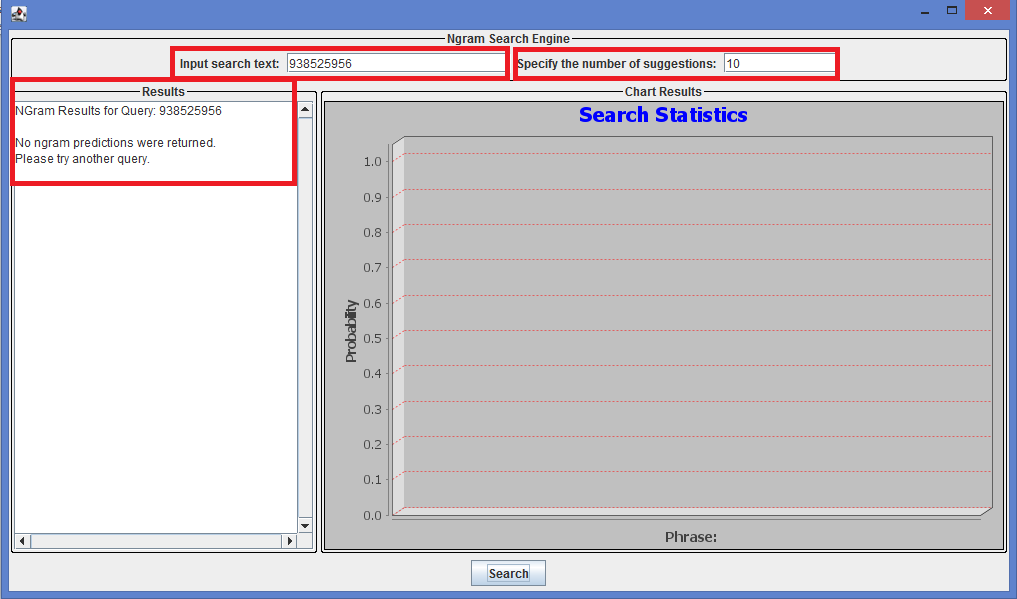
NODES NOT DISPLAYED

**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**.

**Test Case 4: Input Search Text:** 938525956

**Number of Suggestions specified by the user:** 10

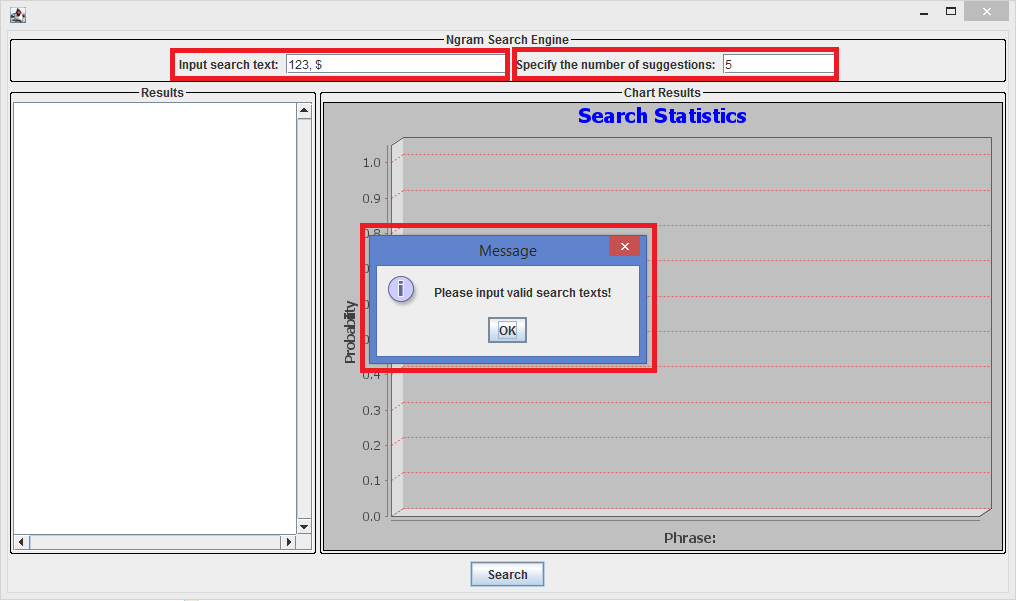
****

**Screenshot 4**

When a user enters a random number input “**938525956**”, specifies the number of suggestions as some random value ‘**10’** and then clicks on the “**Search**” button, there are no search results displayed and an error message “**No ngram predictions were returned. Please try another query**” is displayed on the node i.e., the store is empty as shown in the above **Screenshot 4**.

**Test Case 5: Input Search Text:** 123, $

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

****

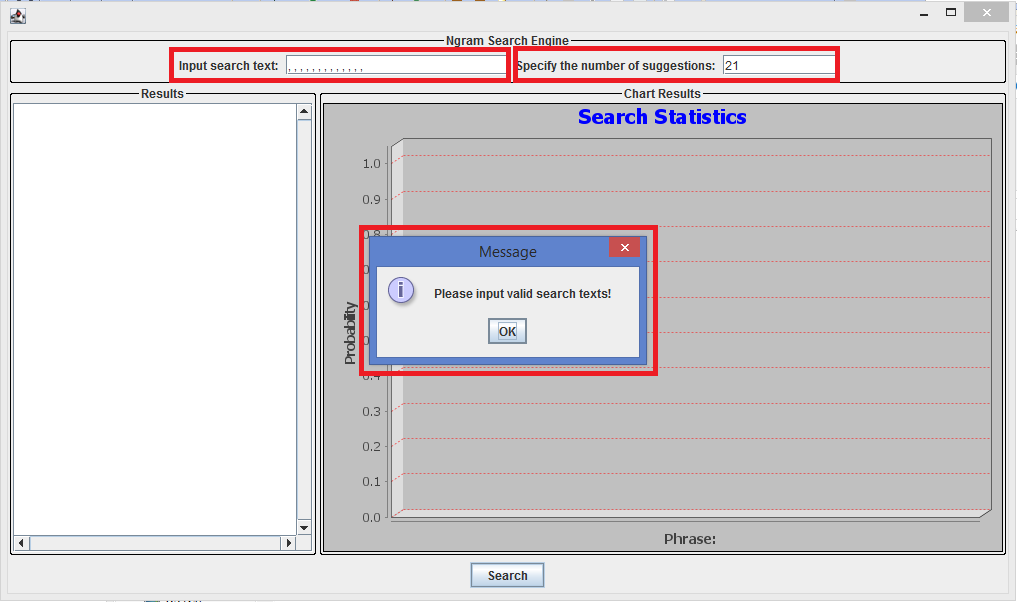
NODES NOT DISPLAYED

**Screenshot 5**

When a user enters a random number input “**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 5**.

**Test Case 6: Input Search Text:** , , , , , , , , , , , , ,

**Number of Suggestions specified by the user:** 21

****

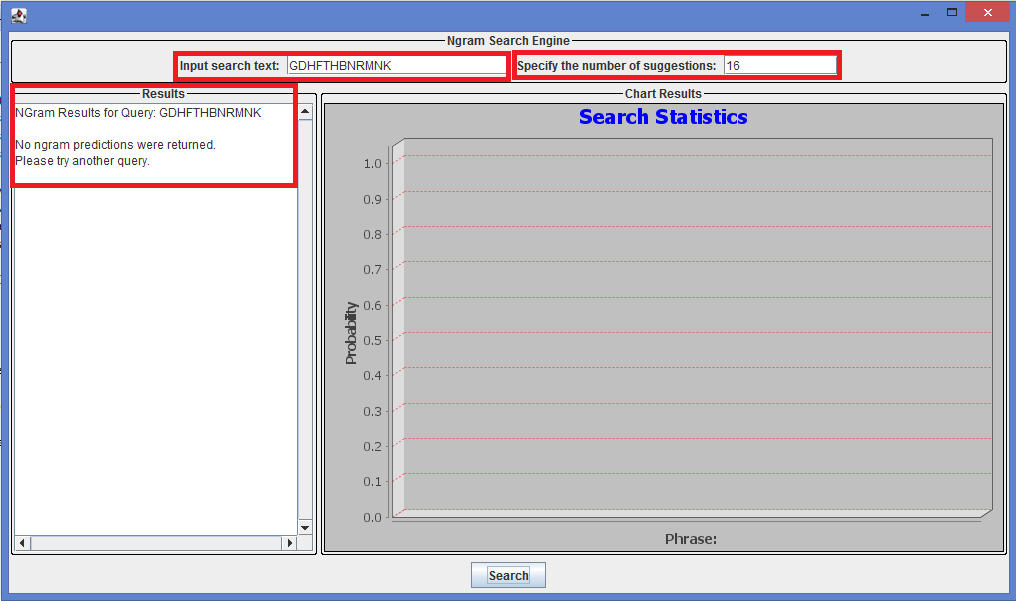
NODES NOT DISPLAYED

**Screenshot 6**

When a user enters a random number input “**, , , , , , , , , , , , ,**”, specifies the number of suggestions as some random value ‘**21’** 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 6**.

**Test Case 7: Input Search Text:** GDHFTHBNRMNK

**Number of Suggestions specified by the user:** 16

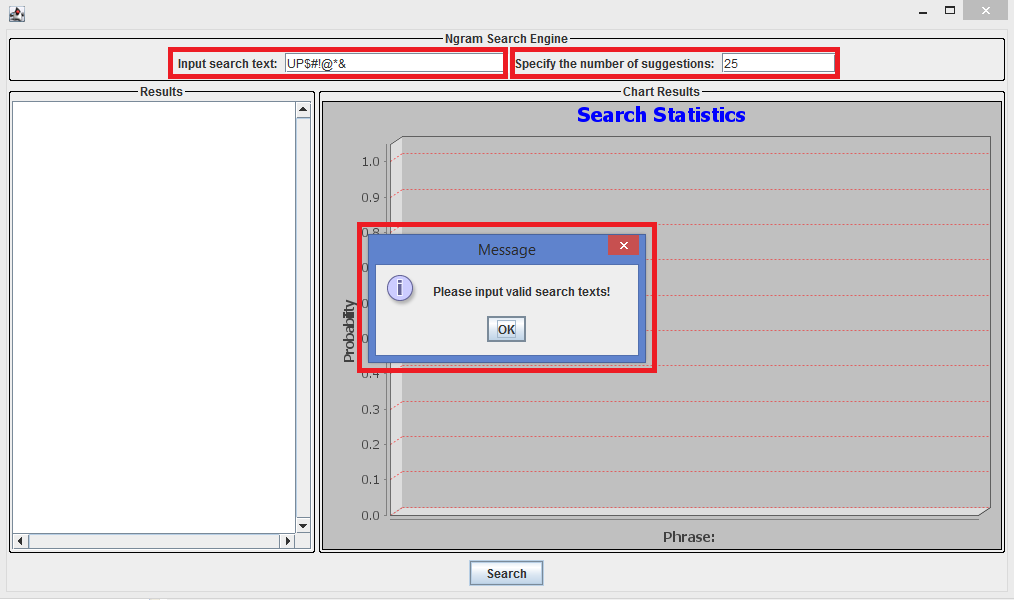
****

**Screenshot 7**

When a user enters a random number input “**GDHFTHBNRMNK**”, specifies the number of suggestions as some random value ‘**16’** 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 7**.

**Test Case 8: Input Search Text:** UP$#!@\*&

**Number of Suggestions specified by the user:** 25



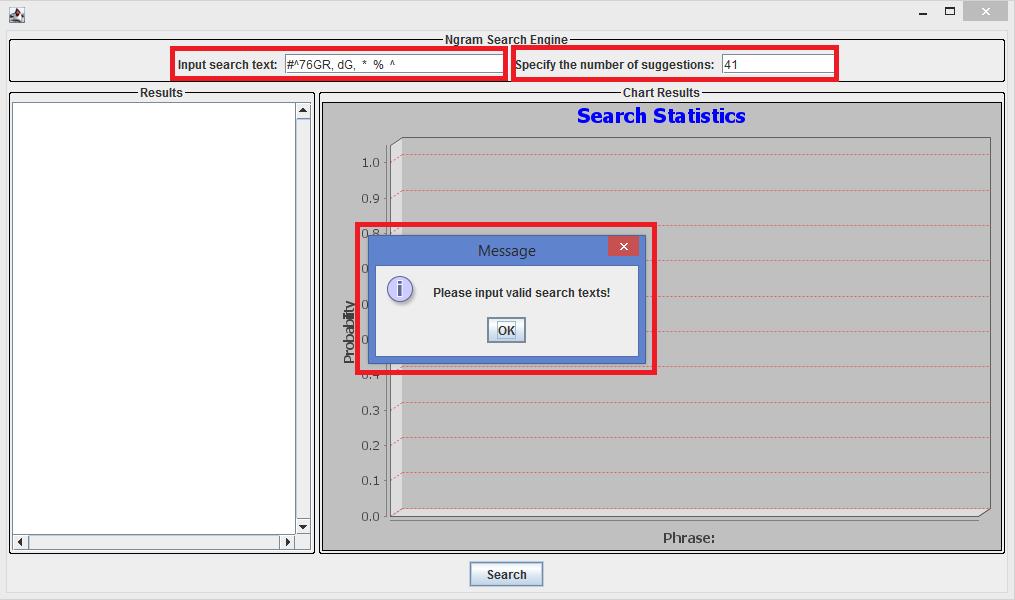
NODES NOT DISPLAYED

**Screenshot 8**

When a user enters a random number input “**UP$#!@\*&**”, specifies the number of suggestions as some random value ‘**25’** and then clicks on the “**Search**” button, an error message “**No ngram predictions were returned. Please try another query.**” is displayed in the place of the node and the corresponding search results of NGram node are also **NOT displayed** i.e., the store is empty as shown in the above **Screenshot 8**.

**Test Case 9: Input Search Text:** #^76GR, dG, \* % ^

**Number of Suggestions specified by the user:** 41

****

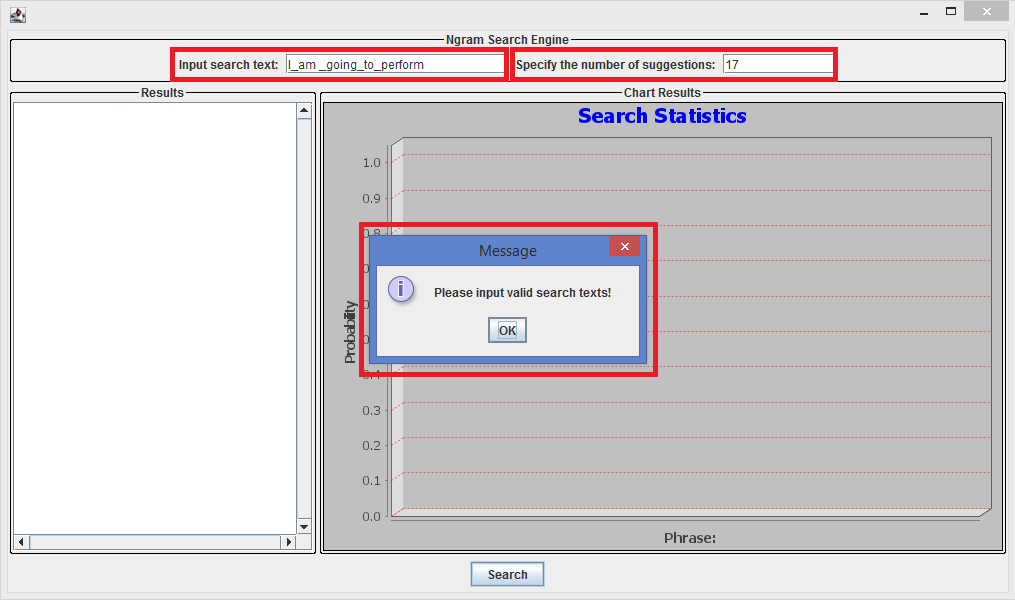
NODES NOT DISPLAYED

**Screenshot 9**

When a user enters an input search string “**#^76GR, dG, \* % ^**“, specifies the number of suggestions as some random value ‘**41’** 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 9**.

**Test Case 10: Input Search Text:** I\_am \_going\_to\_perform

**Number of Suggestions specified by the user:** 17



NODES NOT DISPLAYED

**Screenshot 10**

When a user inputs a search string “**I\_am \_going\_to\_perform**”, specifies the number of suggestions as some random value ‘**17’** 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 10**.