Chapter 5 Research Experiment

5.1 Experimental Design

A set of experiments designed to verify the workload model is presented in this study by using a prototype system. The prototype system can connect to a web search service to do a proper search. Following our survey, we chose Yahoo! Web Search APIs because they provide the largest number of APIs and the most performance metrics.

5.1.1 Experimental Specification

Our set of experiments consists of ten basic and synthetic tests for benchmarking. Each test includes a case and a specific function of the prototype system. After finishing the experiments, the workload model is completely verified. We specify the corresponding selection of the prototype system and follow specifications in execution of each test. The specifications of each test consist of a page model selection, a query model selection, a keyword and the corresponding parameters for web search service APIs. The details of each ten tests are described in the following sections.

(1) Test1:Single Page- URL

The first test suite is Single Page-URL. The purpose is to test “Anchor Text” of Single Page. To complete the test, the corresponding selection in the page model and the query model is chosen. The parameters required by the Yahoo! Web Search APIs and keywords should be entered to generate the web search script. “URL” in the Page Model is selected. In addition, this test is a kind of “topic distillation” query, so we also select “Topic Distillation” in the Query Model. The keyword is “SOA”. The two parameters of the Yahoo! Search APIs are set to “query=inurl:SOA” and “Results=100”. Table 5.1 shows the specifications of the test suite.

Table 5.1: The Test Specifications of Single Page-URL

Page Model: Single Page	URL
Query Model: Query Type	Topic distillation
Keyword	SOA
Yahoo! Search API parameter	query = inurl: SOA
Yahoo! Search API parameter	Results =100

(2) Test2:Single Page-Font Size, Font Color, Frame, Meta, Table

The second test suite is Single Page-Font Size, Font Color, Frame, Meta, and Table. The purpose is to test “Font Size”, “Font Color”, “Frame”, “Meta” and “Table” of Single Page. To complete this test, the corresponding selection in the page model and the query model should be chosen. The parameters required by the Yahoo! Web Search APIs and keywords should be entered to generate the web search script. The options for “Font Size”, “Font Color”, “Frame”, “Meta” and “Table” are selected in the Page Model. This test is a kind of “topic distillation” query so we also select “Topic Distillation” in the Query Model. The keyword is “SOA”. The two parameters of the Yahoo! Search APIs are set to “query = SOA” and “Results = 100”. Table 5.2 shows the specifications of the test suite.

Table 5.2: The Test Specifications of Single Page-Font Size, Font Color, Frame, Meta and Table

Page Model: Single Page	Font Size, Font Color, Frame, Meta, Table
Query Model: Query Type	Topic distillation
Keyword	SOA
Yahoo! Search API parameter	query = SOA
Yahoo! Search API parameter	Results = 100

(3) Test3: Single Page-Title

The third test suite is Single Page-Title. The purpose is to test “Title” of Single Page. To complete this test, the corresponding selection in the page model and the query model should be chosen. And the parameters that are required by the Yahoo! Web Search APIs and keywords should be entered to generate the web search script. “Title” option is selected in the Page Model. In addition, this test is a kind of “Topic distillation” query so we also select the “Topic Distillation” in the Query Model. The keyword is “SOA”. The two parameters of the Yahoo! Search APIs are set to “query = i ntitle: SOA” and “Results = 100”. Table 5.3 shows the specifications of the test suite.

Table 5.3: The Test Specifications of Single Page-Title

Page Model: Single Page	Title
Query Model: Query Type	Topic distillation
Keyword	SOA
Yahoo! Search API parameter	query = intitle: SOA
Yahoo! Search API parameter	Results = 100

(4) Test4:Multi Page-Company

The fourth test suite is Multi Page-Company. The purpose is to “Company” of Multi Page. To complete this test, the corresponding selection in the page model and the query model should be chosen. And the parameters that required by the Yahoo! Web Search API and keywords should be

entered to generate the web search script. The “Company” option is selected in the Page Model. In addition, this test is a kind of “Topic distillation” query so we also select the “Topic Distillation” in the Query Model. The keyword is “SOA”. The three parameters of the Yahoo! Search APIs are set to “query = intitle: SOA”, “site = www.ibm.com” and “Results = 100”. Table 5.4 shows the specifications of the test suite.

Table 5.4: The Test Specifications of Multi Page-Company

Page Model: Multi-Page	Company
Query Model: Query Type	Topic distillation
Keyword	SOA
Yahoo! Search API parameter	query=SOA
Yahoo! Search API parameter	site=www.ibm.com
Yahoo! Search API parameter	Results=100

(5) Test5:Query Type-Homepage Finding

The fifth test suite is Query Type-Homepage Finding. The purpose is to test “Homepage Finding” of Query Type to find the homepage of a website. To complete this test, the corresponding selection in the query model should be chosen. And the parameters required by the Yahoo! Web Search APIs and keywords should be entered to generate the web search script. The “Homepage Finding” is selected in the Query Model. The keyword is “ibm”. The two parameters of the Yahoo! Search APIs are set to “query = hostname:www.IBM.com” and “Results = 100” . Table 5.5 shows the specifications of the test suite.

Table 5.5: The Test Specifications of Query Type- Homepage Finding

Query Model: Query Type	Homepage finding
Keyword	ibm
Yahoo! Search API parameter	query = hostname:www.IBM.com
Yahoo! Search API parameter	Results = 100

(6) Test6: Query Type- Named page Finding

The sixth test suite is Query Type- Named page Finding. The purpose is to test “Named page Finding” in the Page Model to find the specific page. To complete this test, the corresponding selection in the query model should be chosen. And the parameters required by the Yahoo! Web Search APIs and keywords should be entered to generate the web search script. The “Named page Finding” option is selected in the Query Model. The keyword is the URL “help.yahoo.com/help/us/ysearch/tips/tips-04.html”. The two parameters of the Yahoo! Search APIs are set to “query = url: help.yahoo.com/help/us/ysearch/tips/tips-04.html” and “Results = 100”. Table 5.6 shows the specifications of the test suite.

Table 5.6: The Test Specifications of Query Type- Named page Finding

Query Model: Query Type	Named page finding
Keyword	help.yahoo.com/help/us/ysearch/tips/tips-04.html
Yahoo! Search API parameter	query=url: help.yahoo.com/help/us/ysearch/tips/tips-04.html
Yahoo! Search API parameter	Results = 100

(7) Test7: Query Type-Topic Distillation

The seventh test suite is Query Type-Topic Distillation. The purpose is to test "Topic Distillation" of Query Type to find the relative pages of a keyword. To complete this test, the corresponding selection in the query model should be chosen. And the parameters required by the Yahoo! Web Search APIs and keywords should be entered to generate the web search script. The "Topic Distillation" option is selected in the query model. The keyword is "SOA". The two parameters of the Yahoo! Search APIs are set to "query = SOA" and "Results = 100". Table 5.7 shows the specifications of the test suite.

Table 5.7: The Test Specifications of Query Type-Topic Distillation

Query Model: Query Type	Topic distillation
Keyword	SOA
Yahoo! Search API parameter	query = SOA
Yahoo! Search API parameter	Results = 100

(8) Test8:Link Structure-Authority-Hub

The eighth test suite is Link Structure-Authority-Hub. The purpose is to test "Authority-Hub" of Link Structure to find the relative pages that link to the authoritative page. To complete this test, the corresponding selection in the query model should be chosen. And the parameters required by the Yahoo! Web Search APIs and keywords should be entered to generate the web search script. the "Authority-Hub" option is selected in the query model. In addition, this test is a kind of "Topic distillation" query so we also select the "Topic Distillation" option in the query model. The keyword is "SOA". The two parameters of the Yahoo! Search APIs are set to "query = link:ibm.com.tw+soa" and "Results = 100". Table 5.8 shows the specifications of the test suite.

Table 5.8: The Test Specifications of Link Structure-Authority-Hub

Query Model: Link Structure	Authority-Hub
Query Model: Query Type	Topic distillation

Keyword	SOA
Yahoo! Search API parameter	query = link:ibm.com.tw+soa
Yahoo! Search API parameter	Results = 100

(9) Test9:Similarity- TLS

The ninth test suite is Similarity- TLS. The purpose is to test “Similarity- TLS”. To complete this test, the corresponding selection in the query model should be chosen. And the parameters required by the Yahoo! Web Search APIs and keywords should be entered to generate the web search script. The “TLS” option is selected in the query model. In addition, this test is a kind of “Topic distillation” query so we also select the “Topic Distillation” option in the query model. The keyword is “Service Oriented Architecture”. The three parameters of the Yahoo! Search APIs are set to “query= Service Oriented Architecture”, “type=phrase” and “Results=100”. Table 5.9 shows the specificationsof the test suite.

Table 5.9: The Test Specifications of Similarity- TLS

Query Model: Similarity	TLS
Query Model: Query Type	Topic distillation
Keyword	Service Oriented Architecture
Yahoo! Search API parameter	query = Service Oriented Architecture
Yahoo! Search API parameter	type = phrase
Yahoo! Search API parameter	Results = 100

(10) Test10:Synonym

The tenth test suite is Synonym. The purpose is to test “Synonym”. To complete this test, the corresponding selection in the query model should be chosen. And the parameters required by the Yahoo! Web Search APIs and keywords should be entered to generate the web search script. The “Topic Distillation” option is selected in the Query Model. The keyword is “SOA”. The three parameters of the Yahoo! Search APIs are set to “query=SOA”, “similar_ok=1” and “Results=100”. Table 5.10 shows the specifications of the test suite..

Table 5.10: The Test Specifications of Synonym

Query Model: Synonym	Synonym
Query Model: Query Type	Topic distillation
Keyword	SOA
Yahoo! Search API parameter	query = SOA
Yahoo! Search API parameter	similar_ok = 1

Yahoo! Search API parameter	Results = 100

5.1.2 Experimental Metrics

There are seven experimental metrics generated for each test as shown in table 5.11. “Response Time” is the time interval between entry of a request and receipt of a response. “Throughput” is the number of operations completed by the system per time unit. “Precision” is the percentage of retrieved data which is relevant. “Recall” is the percentage of relevant data which have been retrieved but here we use “Relative Recall” as explained in the chapter three. “Recall” and “Precision” are two important measures for evaluation of information retrieval. “TotalResultsAvailable” and “TotalResultsReturned” are two numbers that most web searches return. “Search Result” lists results returned by a web search service, including “Title”, “Summary” and “URL”. “Response Time”, “Throughput”, “TotalResultsAvailable”, “TotalResultsReturned” and “Search Result” are automatically generated after each test is executed. We then click on every list from “Search Result” to check relevance and calculate “Precision” and “Relative Recall”.

Table 5.11: Experimental Metric

	Experimental Metric	**Description**
	Response Time	Interval between entry of a request and receipt of a response.
	Throughput	Number of operations completed by the system per time unit.
	Precision	The percentage of retrieved data those are relevant.
	Relative Recall	The percentage of relevant data that have been retrieved.
	TotalResultsAvailable	The number of query matches in the database.
	TotalResultsReturned	The number of query matches returned.
	Search Result	The list of query matches. Each includes Title, Summary and URL. ● Title: the title of the web page. ● Summary: summary text

		associated with the web page. ● URL: the URL for the web page.

5.2 Experimental Execution and Results

The ten baseline test cases have been executed with the research model and the prototype system. We have adopted Yahoo! Web Search APIs. In this section, we detail the results of each experiment with the selection of input parameters and the output test results.

5.2.1 Single Page- URL

In the first test, Single Page-Anchor Text, we click on “URL”,”Topic Distillation” and input keyword “SOA”. We click “Generate Script” as shown in Figure 5.1. There are two main result screens. Figure 5.2 shows the query script that is generated by the prototype sytem. The script illustrates the input requirements have been specified for the web search request. Figure 5.3 shows the search result of Test1. “TotalResultsAvailable”, “TotalResultsReturned” and the lists from “Search Result” are returned by the Yahoo! web search services. “TotalResultsAvailable” is set with “Response Time” and “Throughput” as the main metrics to be automatically generated. Based on the search results, we will check the relevance of each URL and calculate precision and recall. Table 5.12 shows the metrics of the Test1 experiment. In addition, we perform a same query on Google and obtain the search result as shown in Figure 5.3 and metrics as shown in Table 5.13.

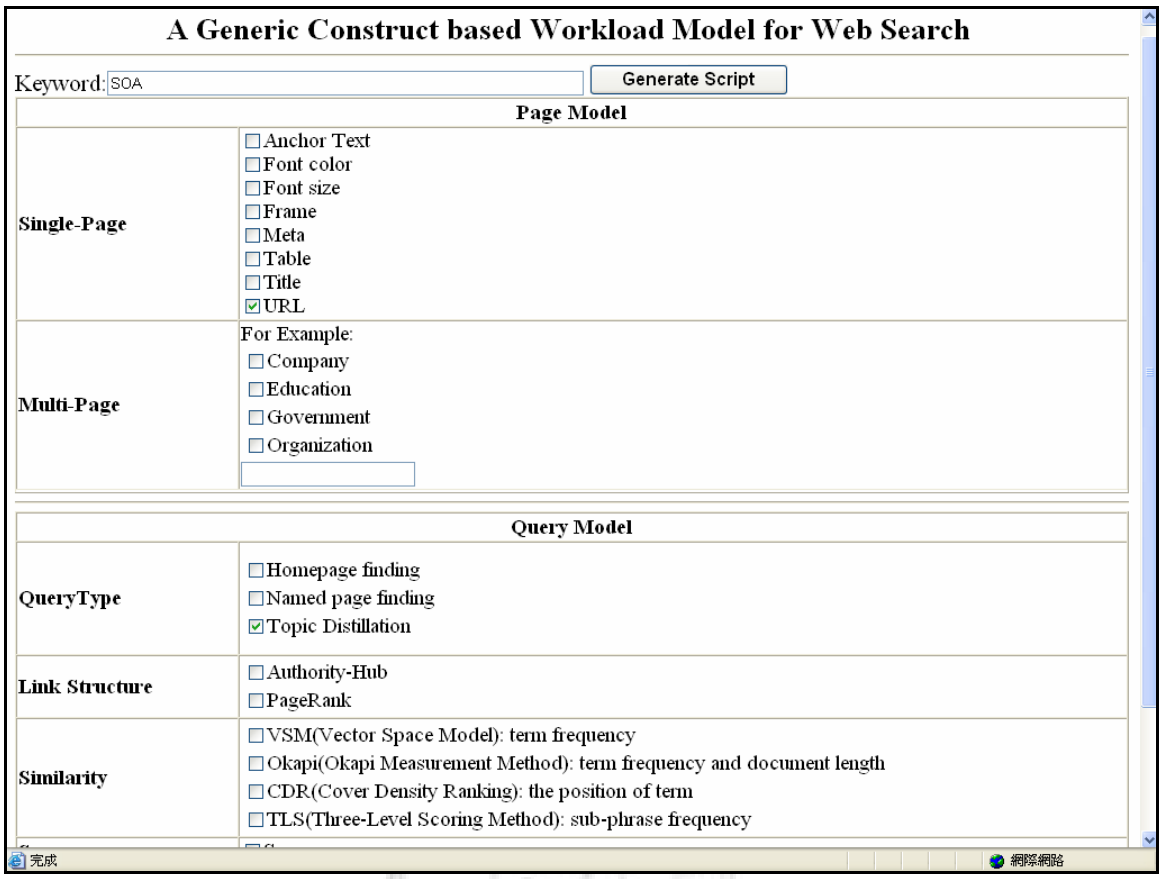

Figure 5.1: Selected Input forTest1

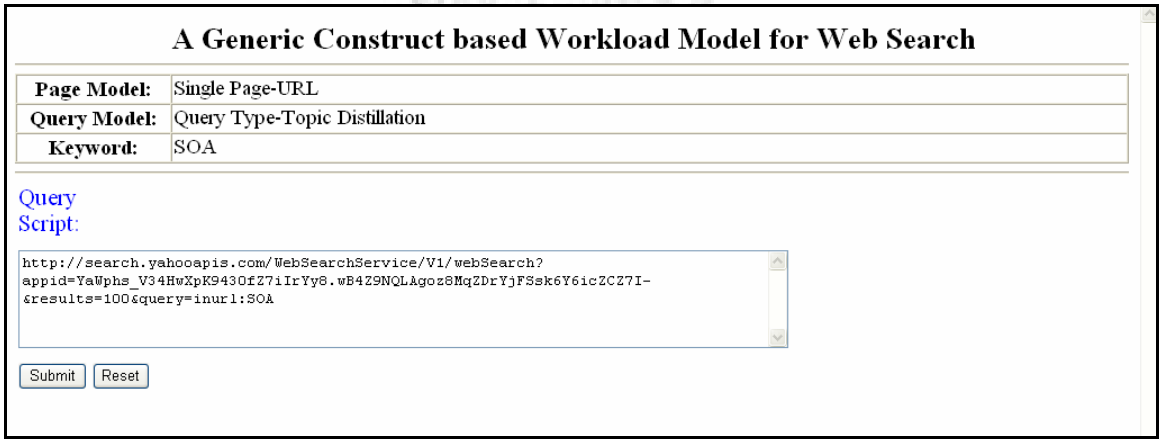

Figure 5.2: Script Output of Test1

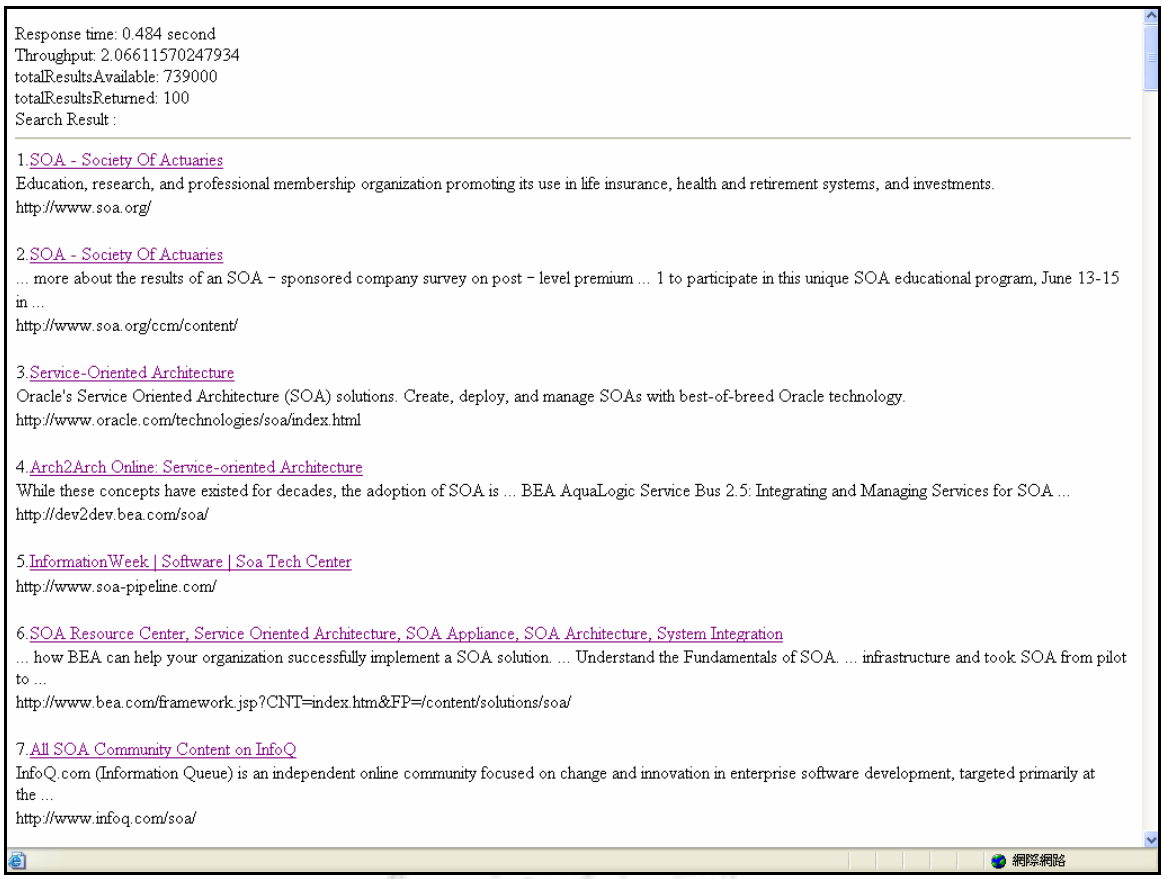

Figure 5.3: The Search Results of Test1

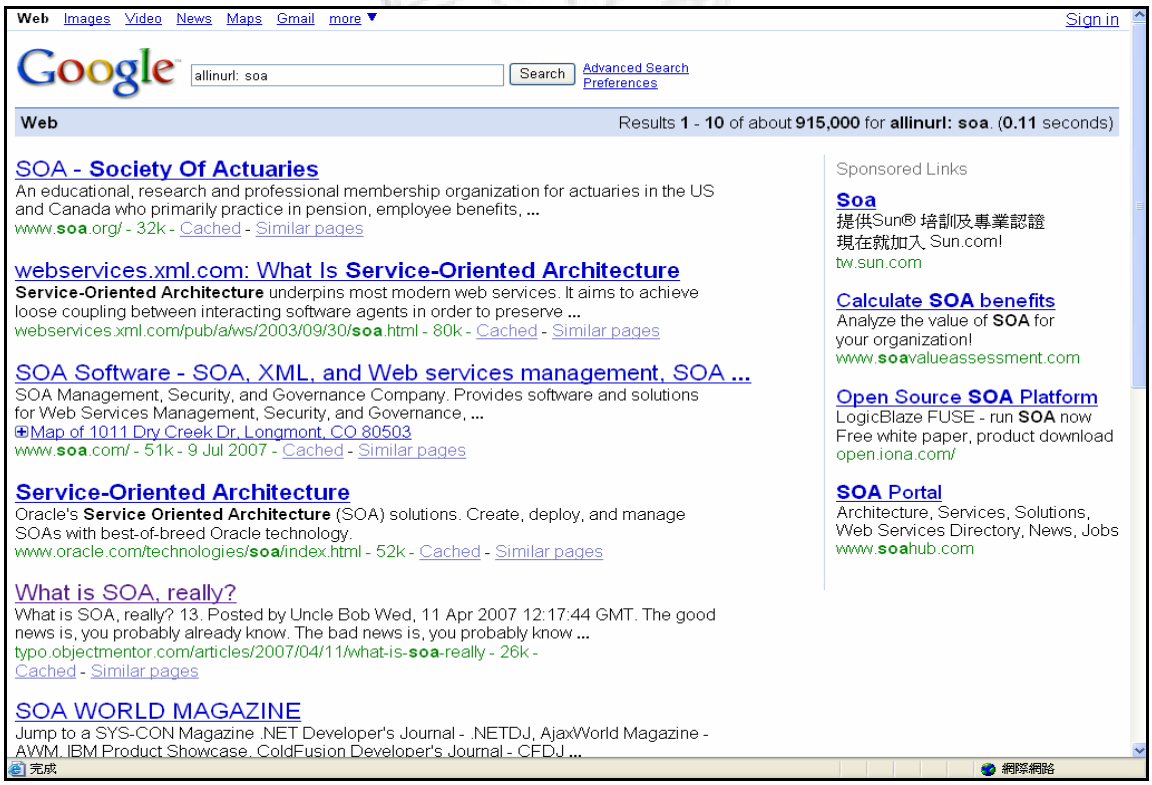

Figure 5.4: The Search Results of Test1 by Google

Table 5.12: Experimental Metrics of Test1

1.	Response Time	0.484 second
2.	Throughput	2.06611570247934
3.	Precision	0.5
4.	Relative Recall	0.48
5.	TotalResultsAvailable	712000
6.	TotalResultsReturned	100
7.	The Search Result	As show in Fig 5.3

Table 5.13: Experimental Metrics of Test1 by Google

1.	Response Time	0.11 second
2.	Throughput	9.09090909090909
3.	Precision	0.6
4.	Relative Recall	0.42
5.	TotalResultsAvailable	915000
6.	TotalResultsReturned	100
7.	The Search Result	As show in Fig 5.4

5.2.2 Single Page- Font Size, Font Color, Frame, Meta and Table

In the second test, the Single Page- Font Size, Font Color, Frame, Meta and Table, we click on “Font Size”, “Font Color”, “Frame”, “Meta”, “Table” and ” Topic Distillation” and input keyword “SOA”. We click on “Generate Script” as shown in Figure 5.5. There are two main result screens. Figure 5.6 shows the query script that is generated by the prototype system. The script illustrates the input requirements have been specified for the web search request. Figure 5.7 shows the search result of Test2. “TotalResultsAvailable”, “TotalResultsReturned” and the lists from “Search Result” are returned by the Yahoo! web search services. “TotalResultsAvailable” is set with “Response Time” and “Throughput” as the main metrics to be automatically generated. Based on the search results, we will check the relevance of each URL and calculate precision and recall. Table 5.14 shows the metrics of the test2. In addition, we perform a same query on Google and obtain the search result as shown in Figure 5.8 and metrics as shown in Table 5.15.

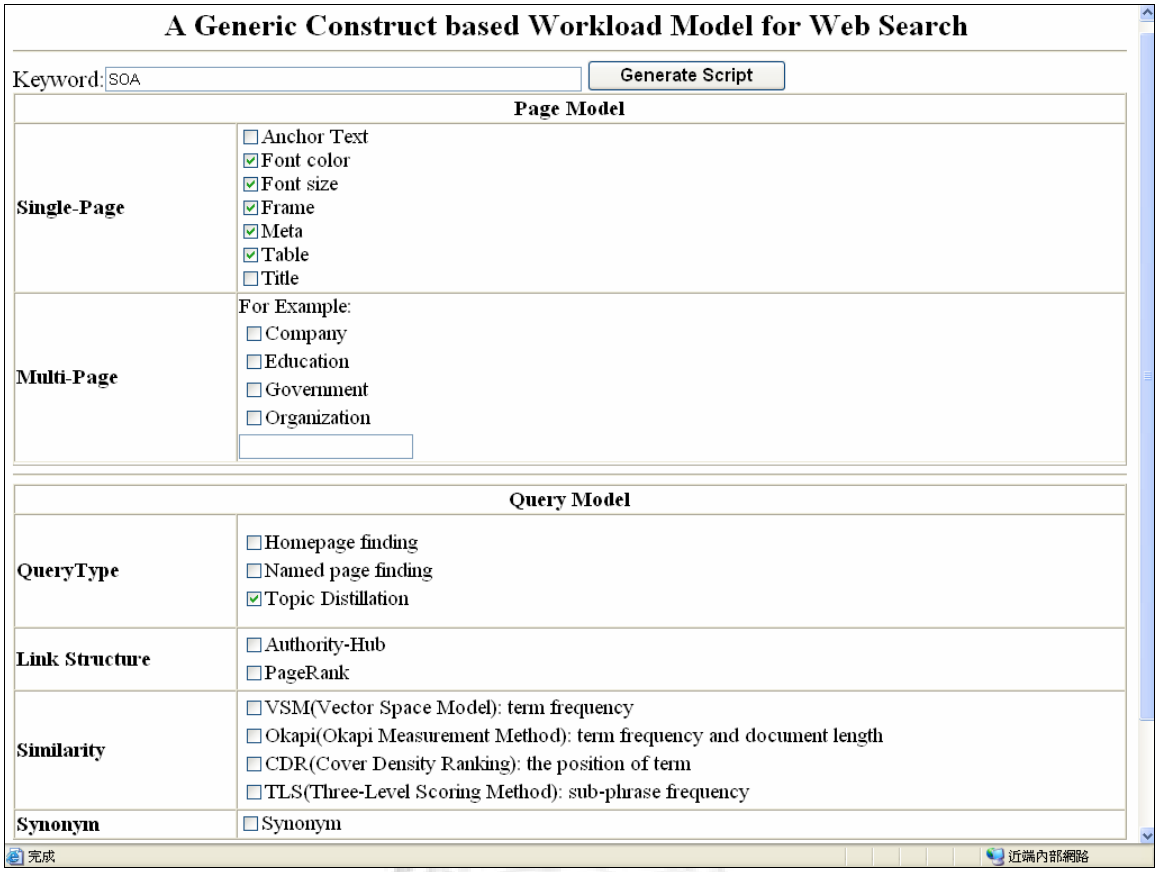

Figure 5.5: Selected Input for Test2

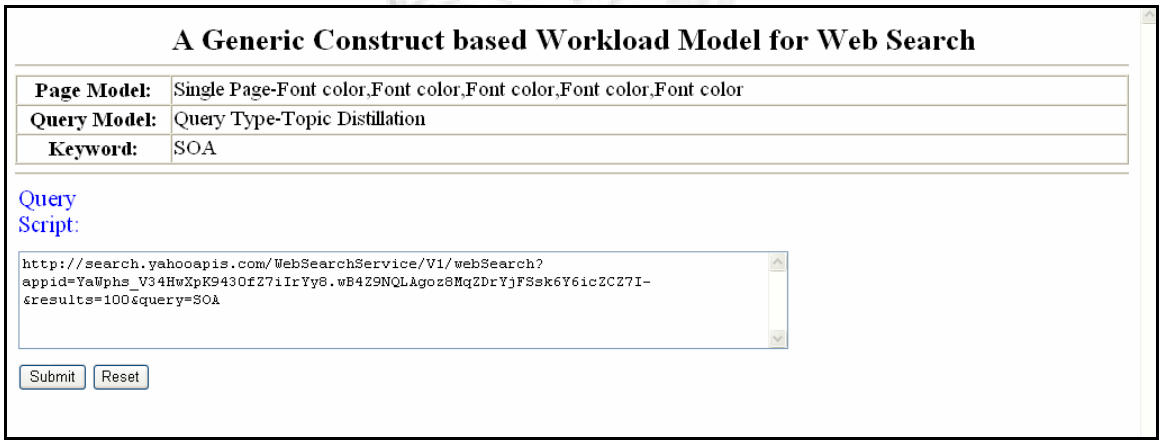

Figure 5.6: Script Output of Test2

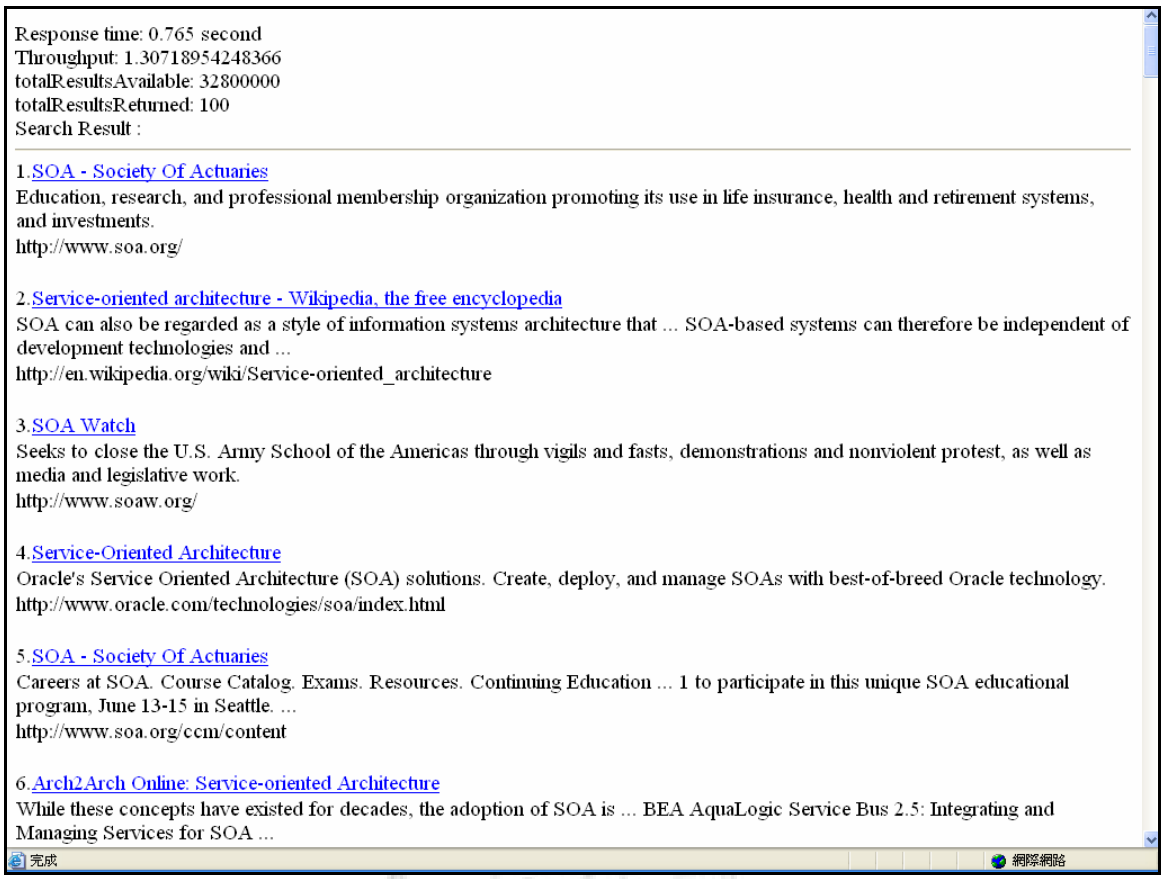

Figure 5.7: The Search Results of Test2

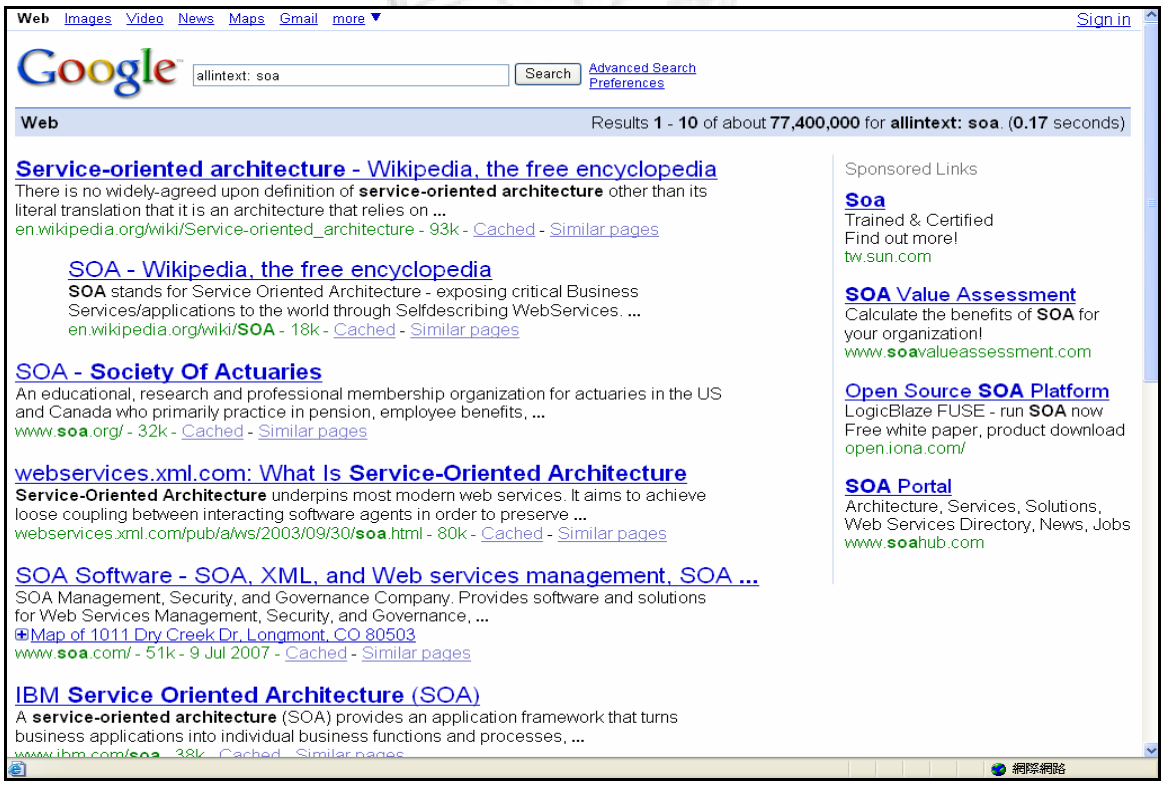

Figure 5.8: The Search Results of Test2 by Google

Table 5.14: Experimental Metrics of test2

1.	Response Time	0.765
2.	Throughput	1.30718954248366
3.	Precision	0.52
4.	Recall	0.605
5.	TotalResultsAvailable	32800000
6.	TotalResultsReturned	100
7.	The Search Result	As show in Fig 5.6

Table 5.15: Experimental Metrics of test2 by Google

8.	Response Time	0.17 second
9.	Throughput	5. 882352941176
10.	Precision	0.53
11.	Recall	0.61
12.	TotalResultsAvailable	77400000
13.	TotalResultsReturned	100
14.	The Search Result	As show in Fig 5.8

5.2.3 Single Page- Title

In the third test, the Single Page- Title, we click on “Title” and” Topic Distillation” and input keyword “SOA”. We click on “Generate Script” as shown in Figure 5.9. There are two main result screens. Figure 5.10 shows the query script that is generated by the prototype system. The script illustrates the input requirements have been specified for the web search request. Figure 5.11 shows the search result of Test1. “TotalResultsAvailable”, “TotalResultsReturned” and the lists from “Search Result” are returned by the Yahoo! web search services. “TotalResultsAvailable” is set with “Response Time” and “Throughput” as the main metrics to be automatically generated. Based on the search results, we will check the relevance of each URL and calculate precision and recall. Table 5.16 shows the metrics of the test3. In addition, we perform a same query on Google and obtain the search result as shown in Figure 5.12 and metrics as shown in Table 5.17.

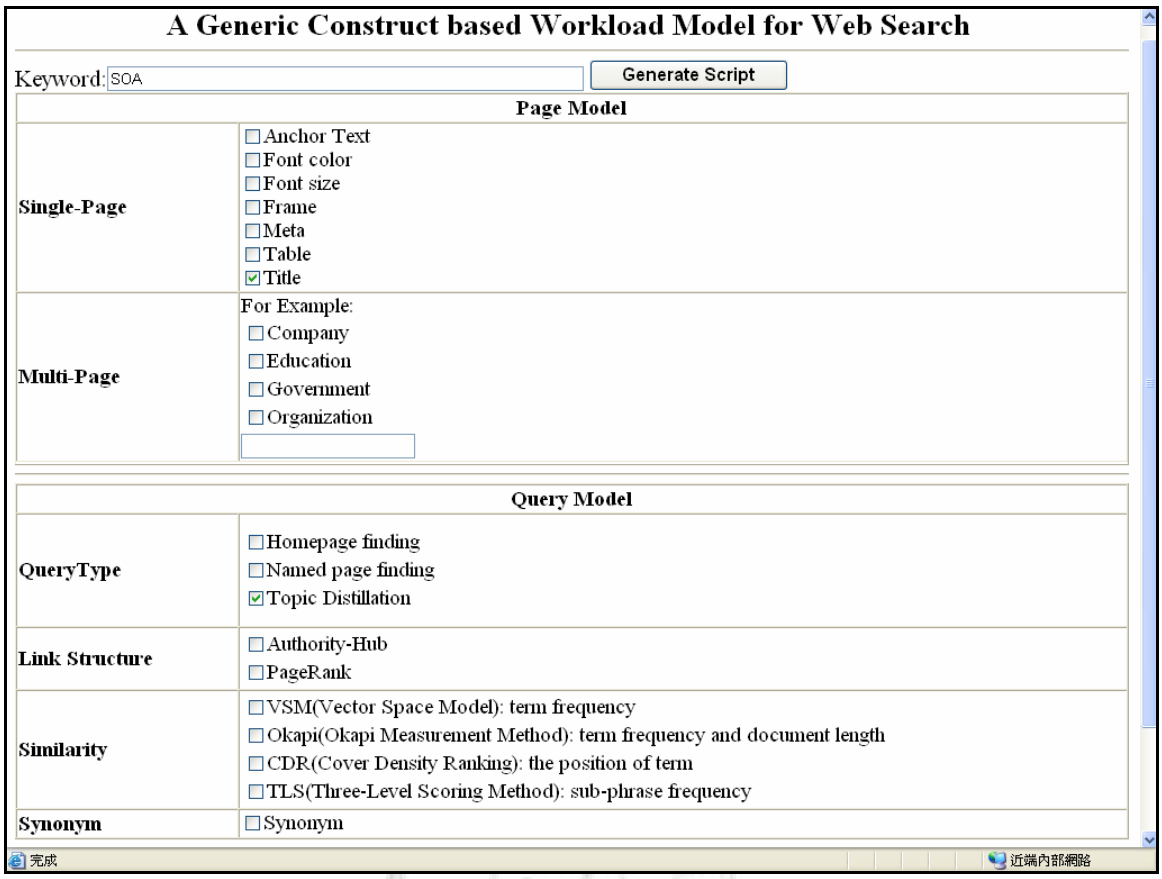

Figure 5.9: Selected Input for Test3

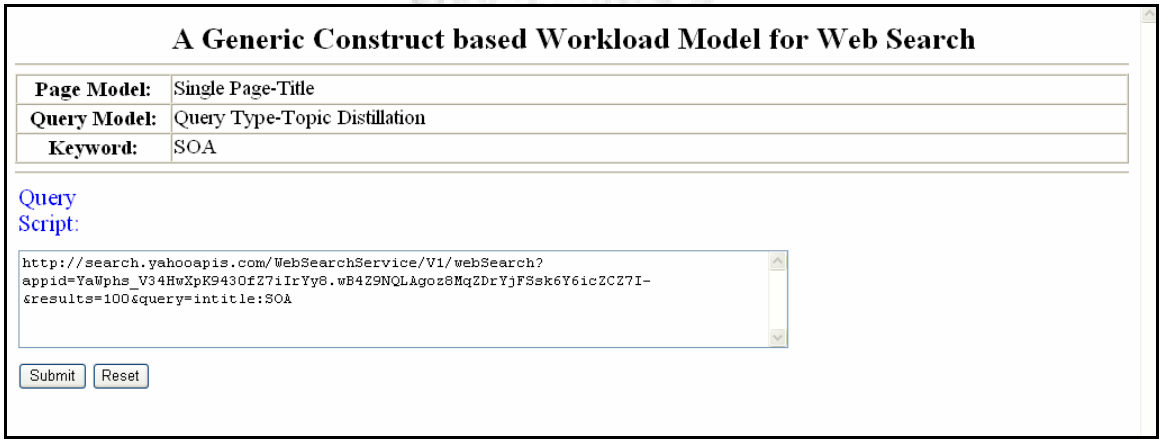

Figure 5.10: Script Output of Test3

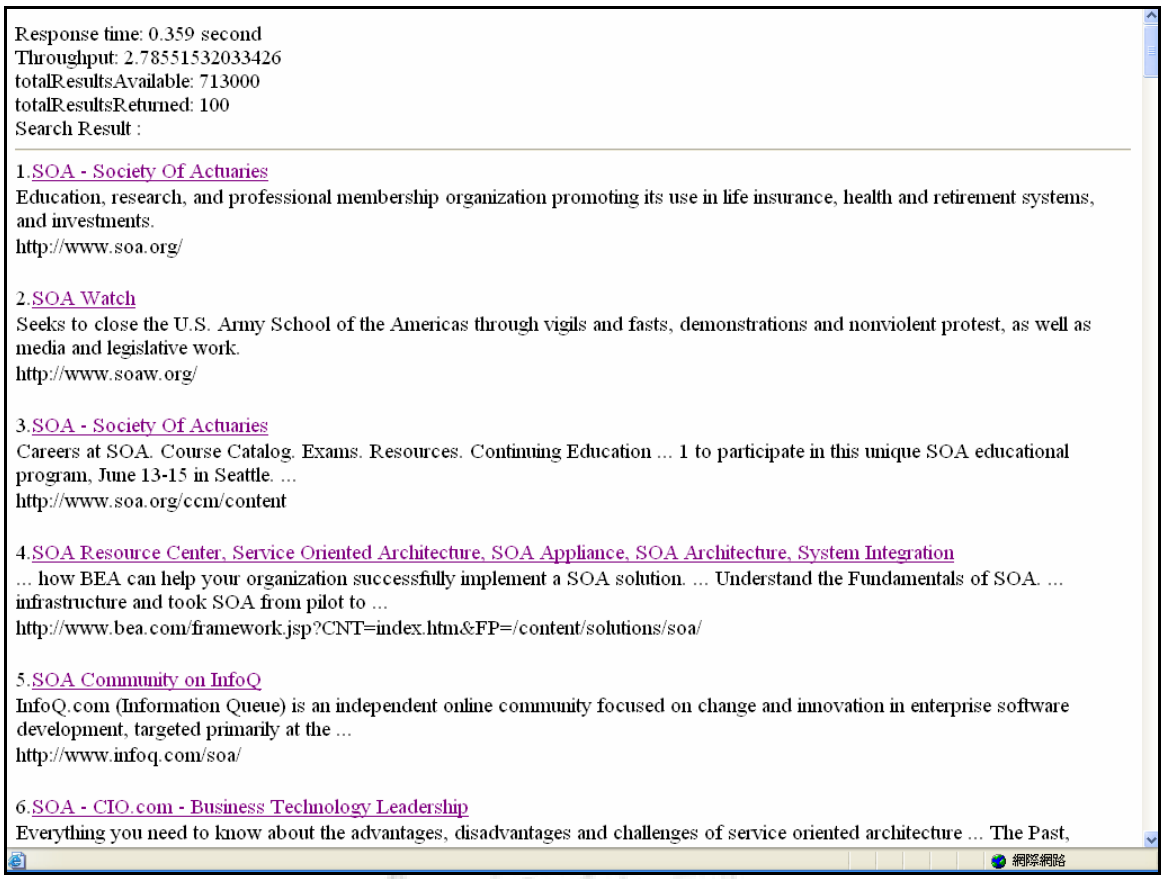

Figure 5.11: The Search Results of Test3

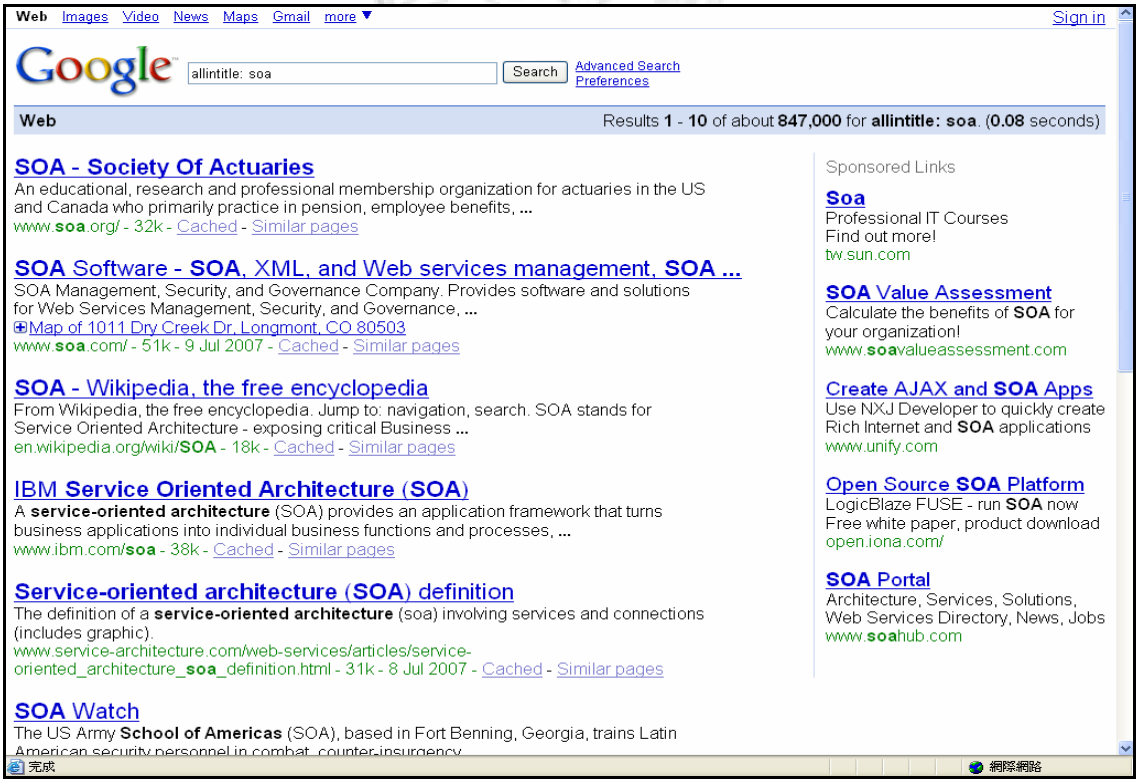

Figure 5.12: The Search Results of Test3 by Google

Table 5.16: Experimental Metrics of test3

1.	Response Time	0.359 second
2.	Throughput	2.78551532033426
3.	Precision	0.6
4.	Recall	0.6
5.	TotalResultsAvailable	713000
6.	TotalResultsReturned	100
7.	The Search Result	As show in Fig 5.11

Table 5.17: Experimental Metrics of test3 by Google

8.	Response Time	0.08 second
9.	Throughput	12.5
10.	Precision	0.61
11.	Recall	0.59
12.	TotalResultsAvailable	847000
13.	TotalResultsReturned	100
14.	The Search Result	As show in Fig 5.12

5.2.4 Multi-Page-Company

In the fourth test, the Multi-Page-Company, we click on “Company”and “Topic Distillation” and input keyword “SOA”. We click on “Generate Script” as shown in Figure 5.13. There are two main result screens. Figure 5.14 shows the query script that is generated by the prototype system. The script illustrates the input requirements have been specified for the web search request. Figure 5.15 shows the search result of Test4. “TotalResultsAvailable”, “TotalResultsReturned” and the lists from “Search Result” are returned by the Yahoo! web search services. “TotalResultsAvailable” is set with “Response Time” and “Throughput” as the main metrics to be automatically generated. Based on the search results, we will check the relevance of each URL and calculate precision and recall. Table 5.18 shows the metrics of the test4.

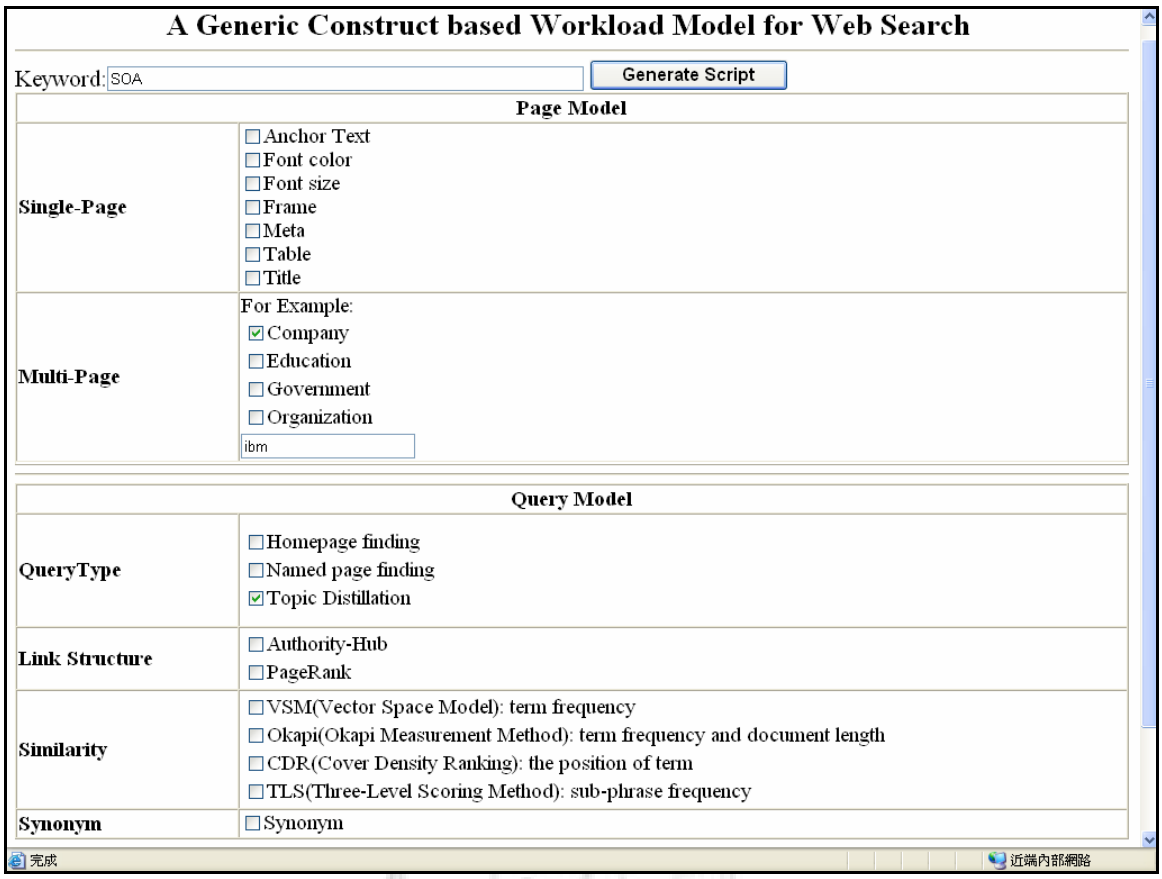

Figure 5.13: Selected Input for Test4

Figure 5.14 Script Output of Test4

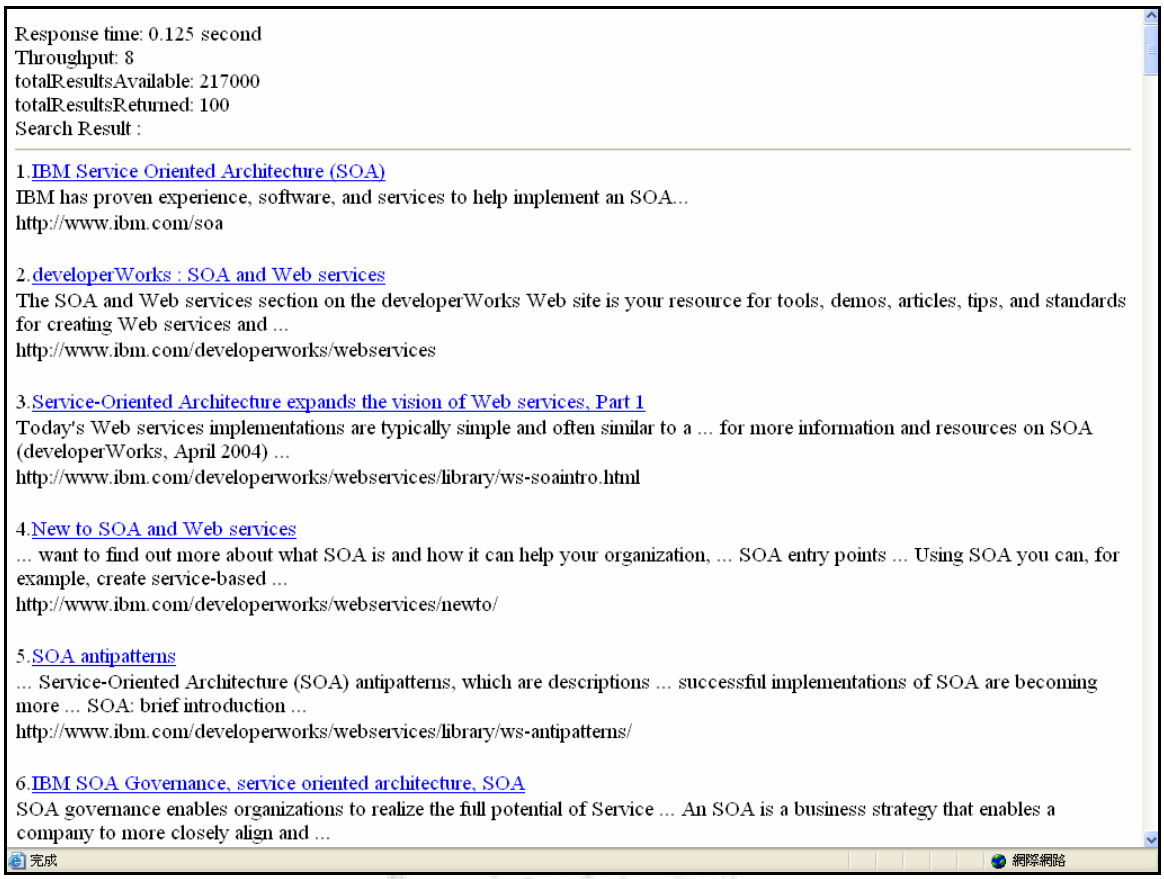

Figure 5.15: The Search Results of Test4

Table 5.18: Experimental Metrics of test4

1.	Response Time	0.125 second
2.	Throughput	8
3.	Precision	0.9
4.	Recall	0.8
5.	TotalResultsAvailable	217000
6.	TotalResultsReturned	100
7.	The Search Result	As show in Fig 5.12

5.2.5 Query Type-Homepage Finding

In the fifth test, the Query Type-Homepage finding, we click on “Homepage Finding” and input keyword “ibm”. We click on “Generate Script” as shown in Figure 5.16. There are two main result screens. Figure 5.17 shows the query script that is generated by the prototype system. The script illustrates the input requirements have been specified for the web search request Figure 5.18 shows the search result of Test5. “TotalResultsAvailable”, “TotalResultsReturned” and the lists from

"Search Result" are returned by the Yahoo! web search services. "TotalResultsAvailable" is set with "Response Time" and "Throughput" as the main metrics to be automatically generated. Based on the search results, we will check the relevance of each URL and calculate precision and recall. Table 5.19 shows the metrics of the test5.

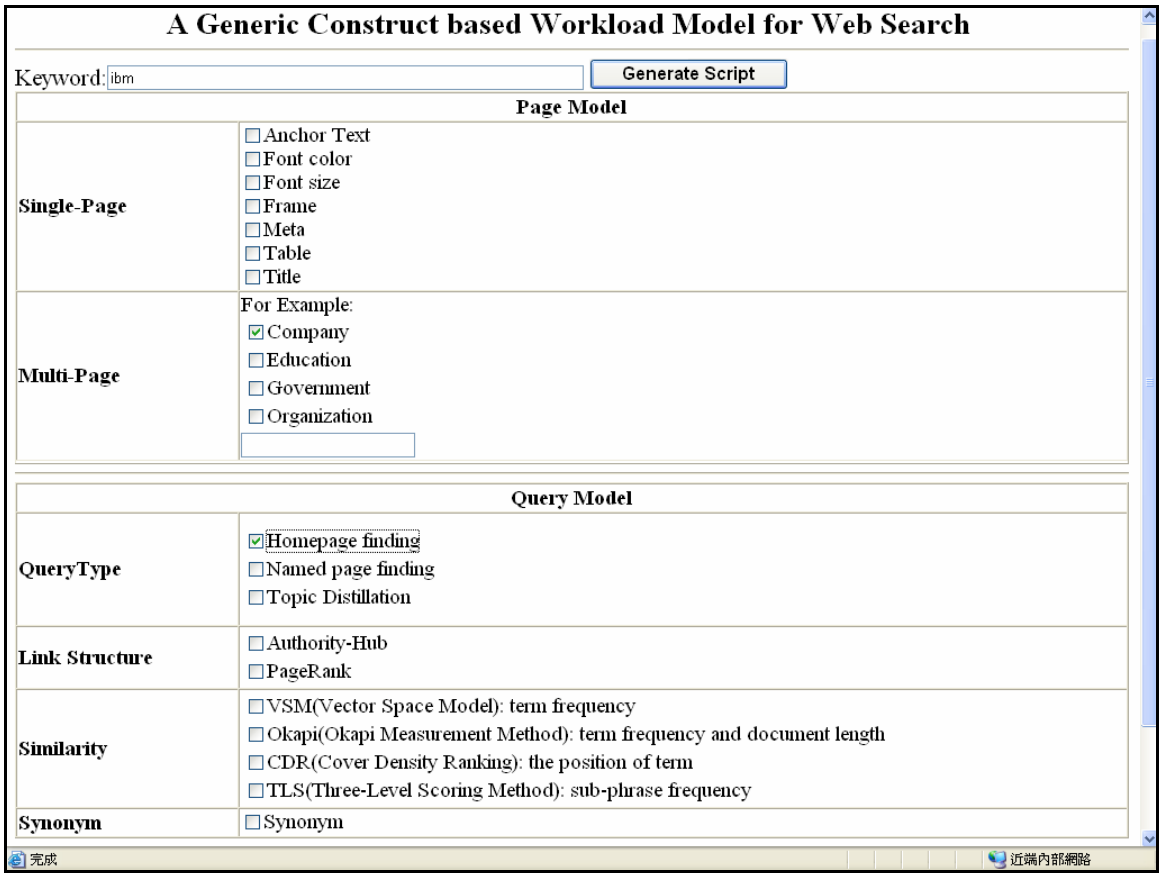

Figure 5.16: Selected Input for Test5

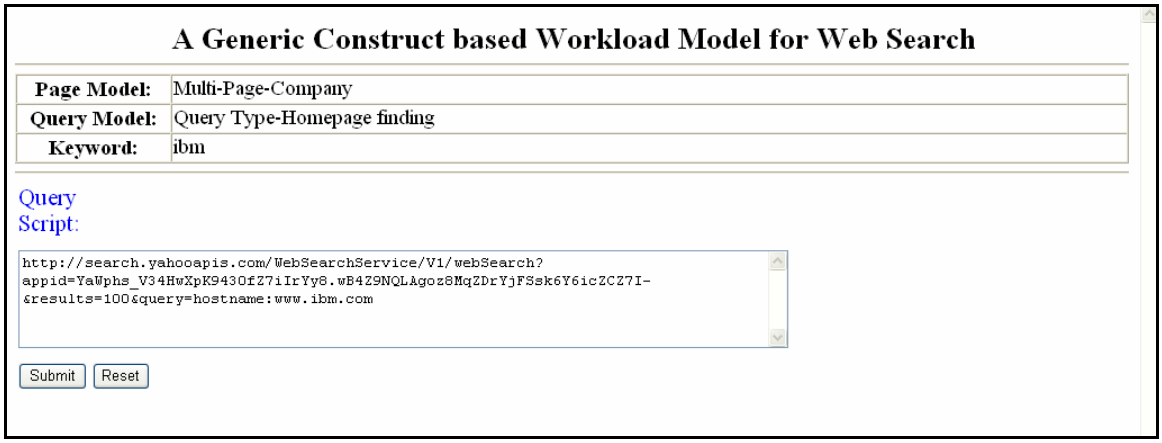

Figure 5.17: Script Output of Test5

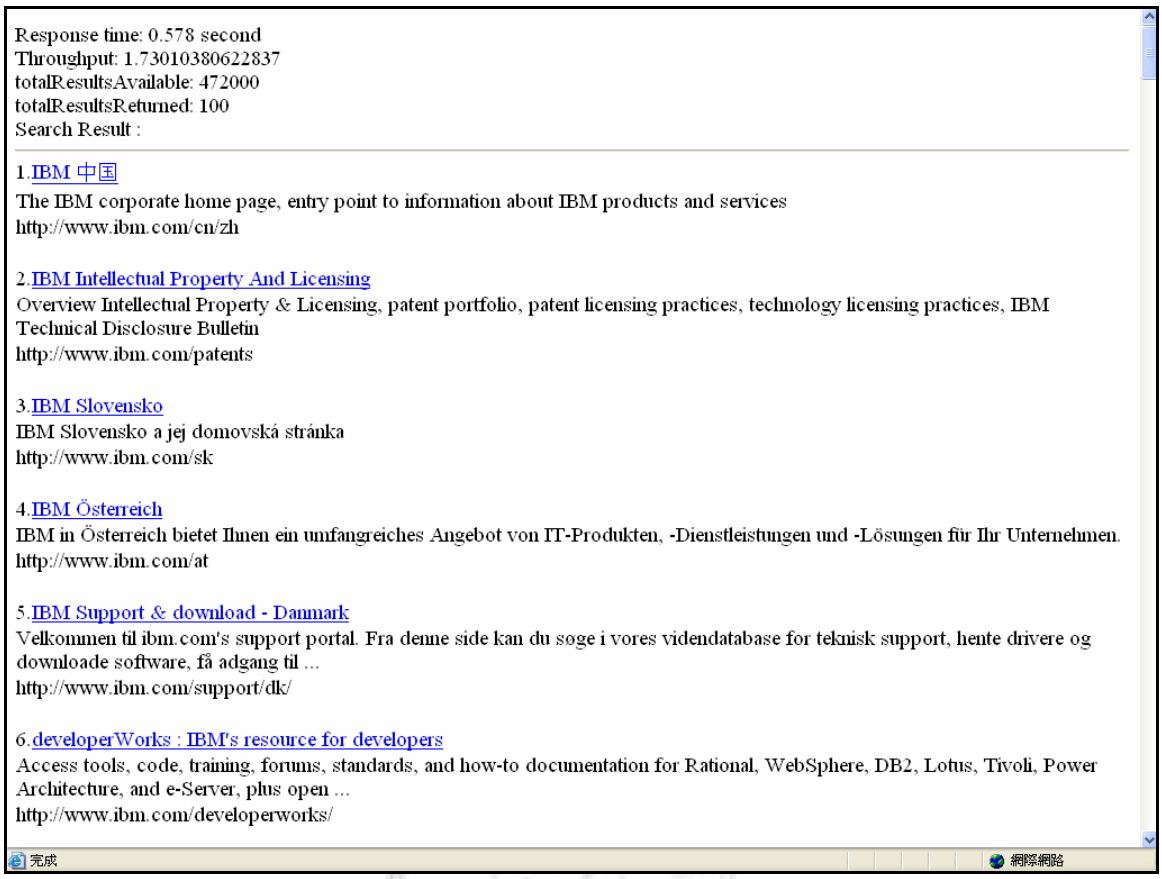

Figure 5.18: The Search Results of Test5

Table 5.19: Experimental Metrics of test5

1.	Response Time	0.578 second
2.	Throughput	1.73010380622837
3.	Precision	1
4.	Recall	1
5.	TotalResultsAvailable	447200
6.	TotalResultsReturned	100
7.	The Search Result	As show in Fig 5.15

5.2.6 Query Type- Named page Finding

In the sixth test, the Query Type- Named page Finding, we click on “Named page Finding” and input keyword “help.yahoo.com/help/us/ysearch/tips/tips-04.html”. We click on “Generate Script” as shown in Figure 5.19. There are two main result screens. Figure 5.20 shows the query script that is generated by the web search service. The script illustrates the input requirements have been specified for the web search request. Figure 5.21 shows the search result of Test6.

"TotalResultsAvailable", "TotalResultsReturned" and the lists from "Search Result" are returned by the Yahoo! web search services. "TotalResultsAvailable" is set with "Response Time" and "Throughput" as the main metrics to be automatically generated. Based on the search results, we will check the relevance of each URL and calculate precision and recall. Table 5.20 shows the metrics of the test6.

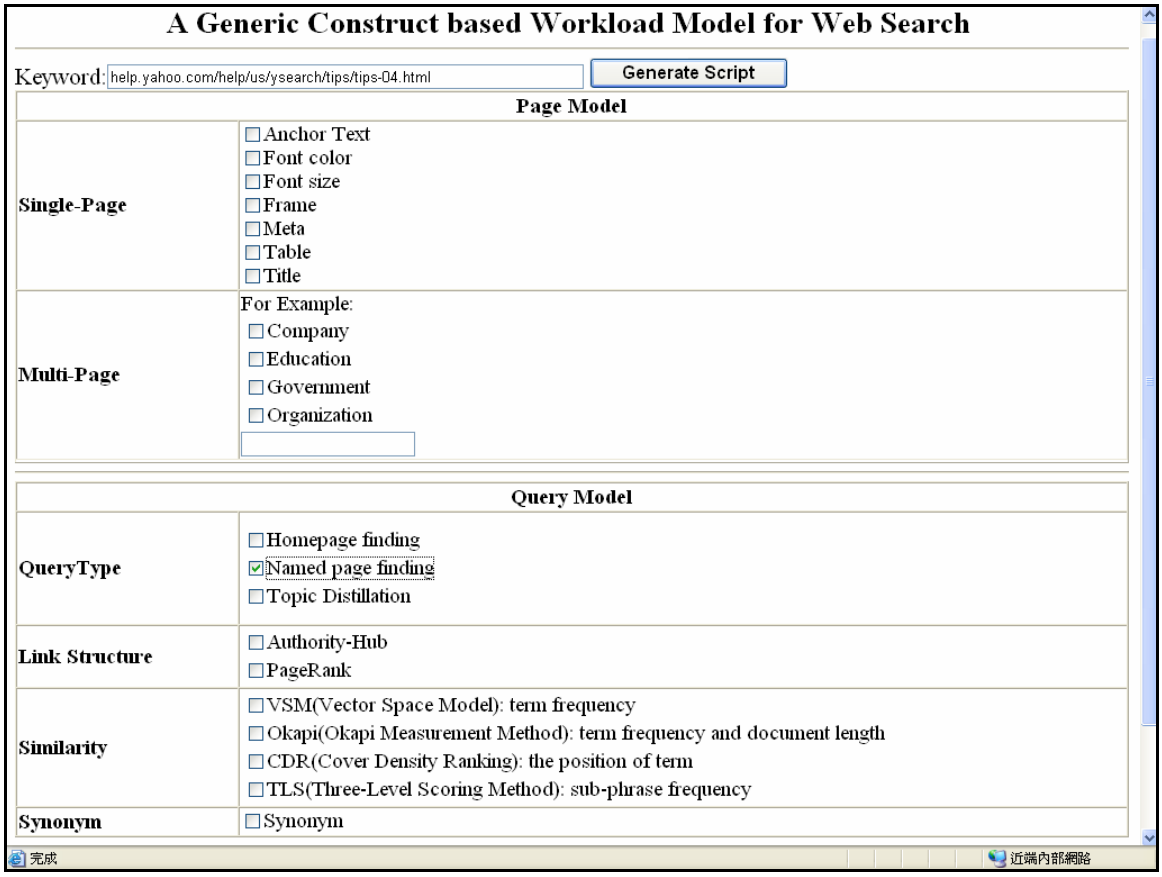

Figure 5.19: Selected Input for Test6

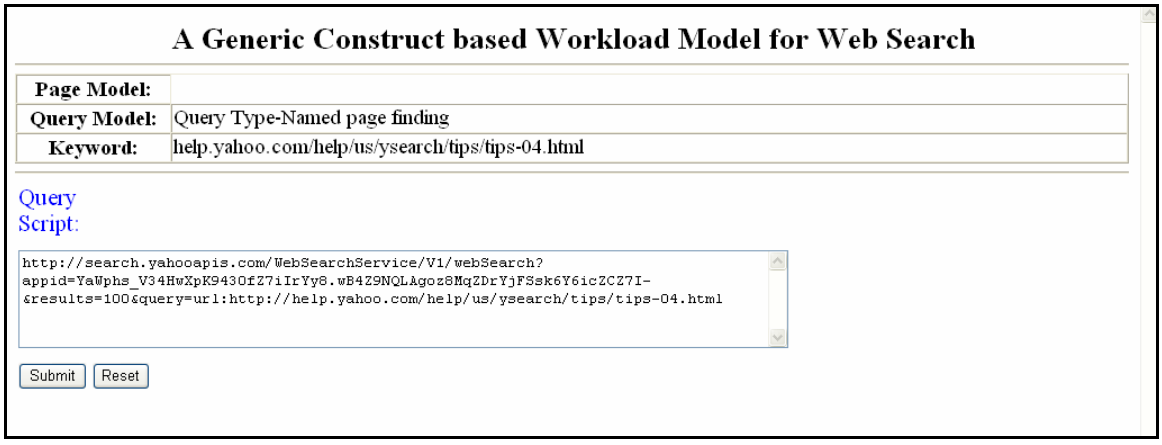

Figure 5.20: Script Output of Test6

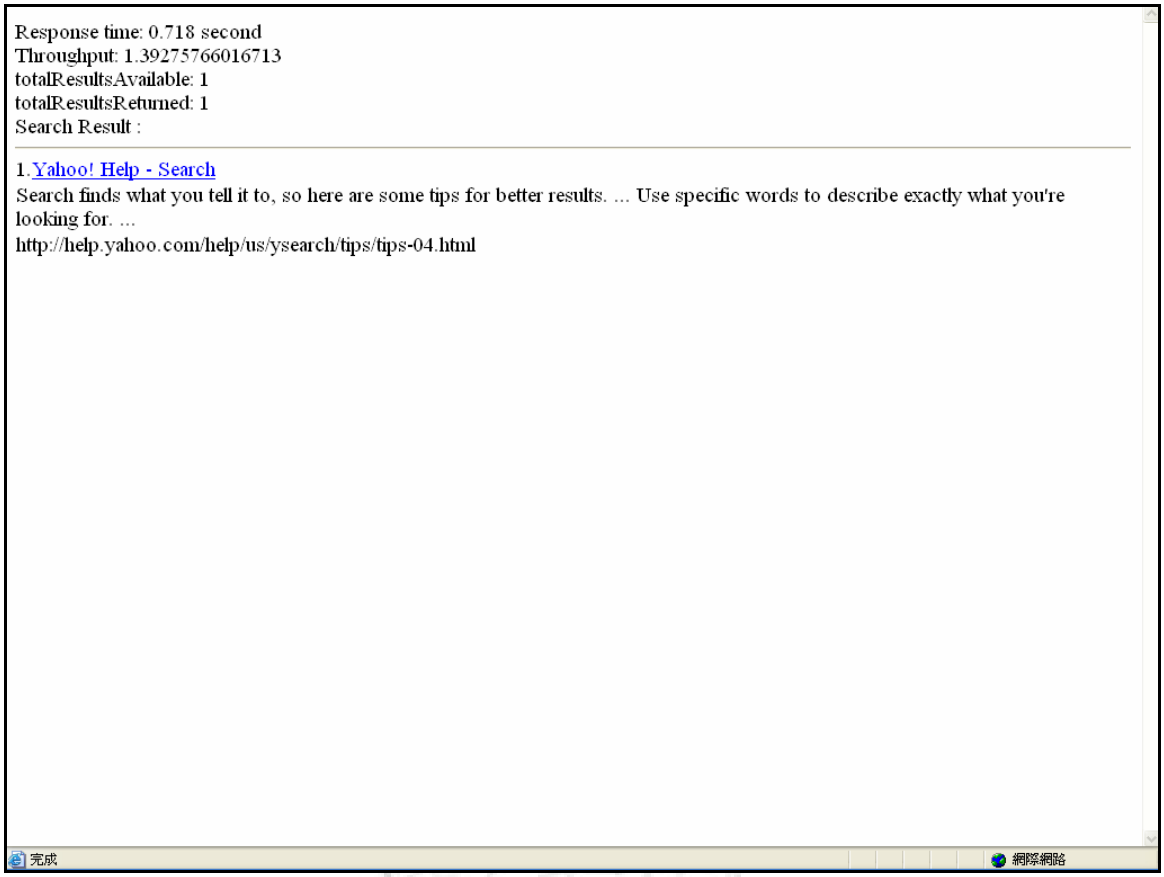

Figure 5.21: The Search Results of Test6

Table 5.20: Experimental Metrics of test6

1.	Response Time	0.718 second
2.	Throughput	1.39275766016713
3.	Precision	1
4.	Recall	1
5.	TotalResultsAvailable	1
6.	TotalResultsReturned	1
7.	The Search Result	As show in Fig 5.18

5.2.7 Query Type-Topic distillation

In the seventh test, Query Type-Topic distillation, we click on “Topic Distillation” and input keyword “ibm”. We click on “Generate Script” as shown in Figure 5.22. There are two main result screens. Figure 5.23 shows the query script that is generated by the prototype system. The script illustrates the input requirements have been specified for the web search request. Figure 5.24

shows the search result of Test7. “TotalResultsAvailable”, “TotalResultsReturned” and the lists from “Search Result” are returned by the Yahoo! web search services. “TotalResultsAvailable” is set with “Response Time” and “Throughput” as the main metrics to be automatically generated. Based on the search results, we will check the relevance of each URL and calculate precision and recall. Table 5.21 shows the metrics of the test7.

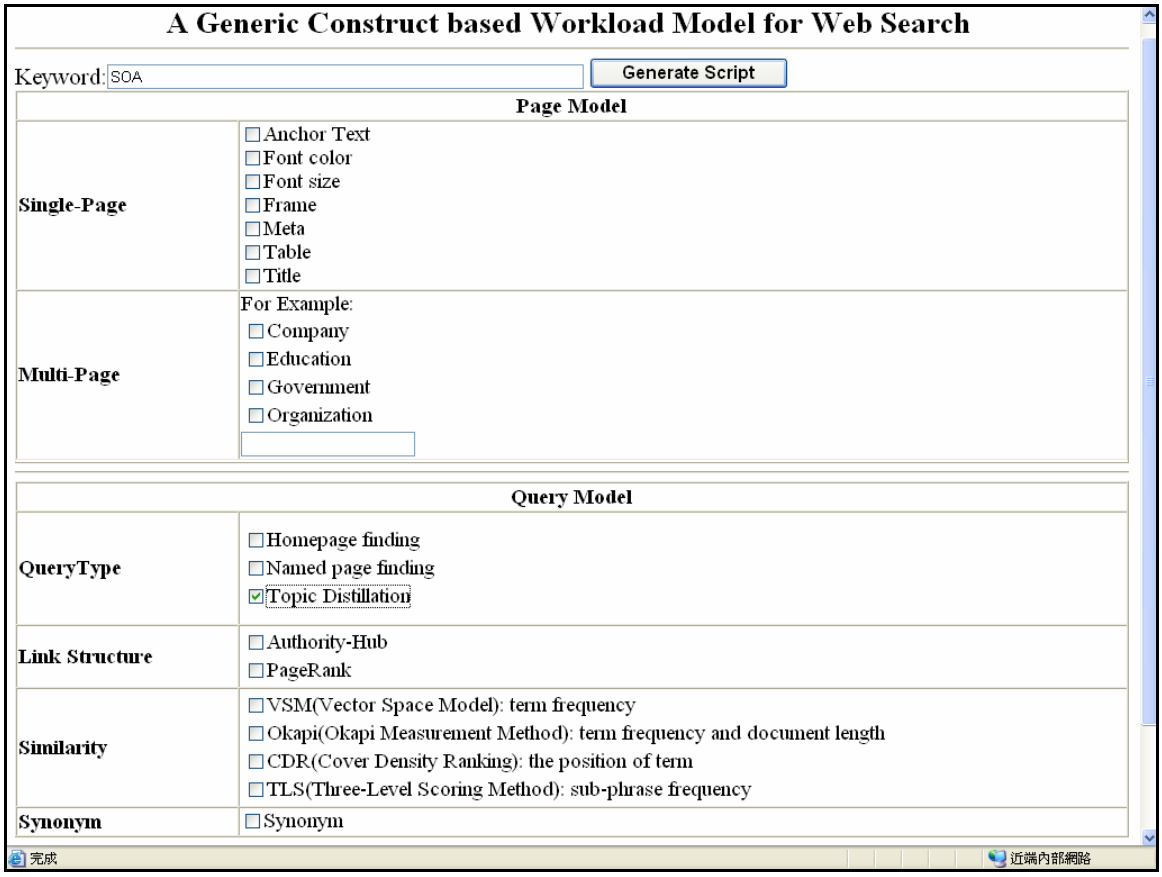

Figure 5.22: Selected Input for Test7

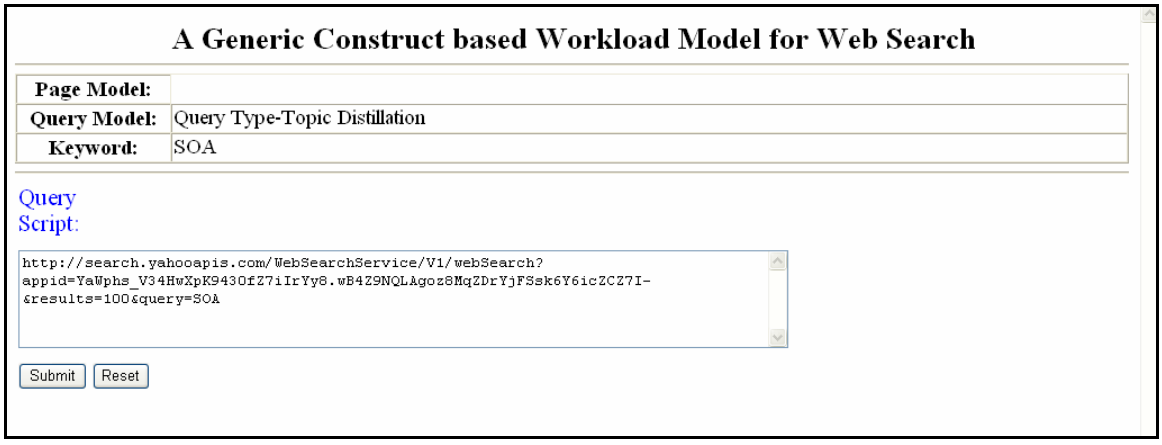

Figure 5.23: Script Output of Test7

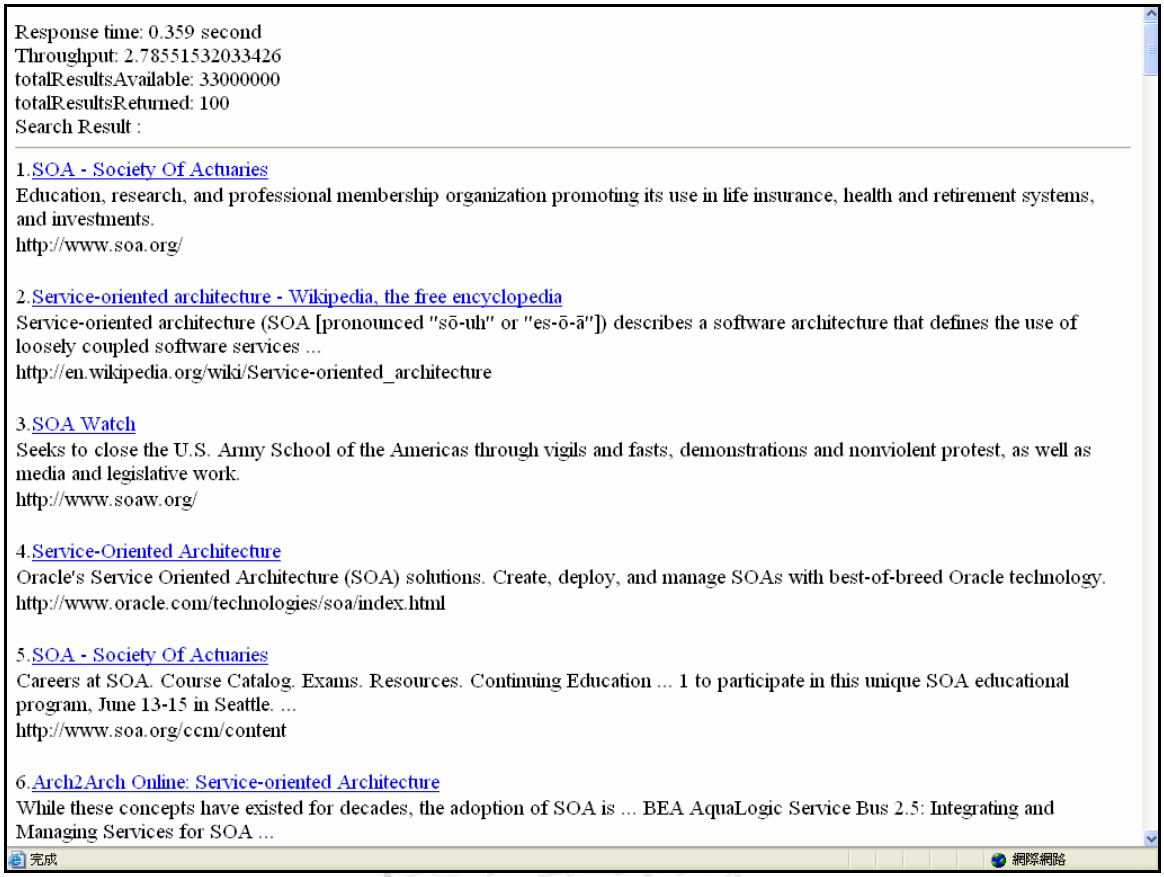

Figure 5.24: The Search Results of Test7

Table 5.21: Experimental Metrics of test7

1.	Response Time	0.359 second
2.	Throughput	2.78551532033426
3.	Precision	0.5
4.	Recall	0.45
5.	TotalResultsAvailable	33000000
6.	TotalResultsReturned	100
7.	The Search Result	As show in Fig 5.21

5.2.8 Link Structure-Authority-Hub

In the eighth test, Link Structure-Authority-Hub, we click on “Authority-Hub” and” Multi-Page-Company” and input keyword “soa” and ”ibm”. We click on “Generate Script” as shown in Figure 5.25. There are two main result screens. Figure 5.26 shows the query script that is generated by the prototype system. The script illustrates the input requirements have been specified

for the web search request. Figure 5.27 shows the search result of Test8. “TotalResultsAvailable”, “TotalResultsReturned” and the lists from “Search Result” are returned by the Yahoo! web search services. “TotalResultsAvailable” is set with “Response Time” and “Throughput” as the main metrics to be automatically generated. Based on the search results, we will check the relevance of each URL and calculate precision and recall. Table 5.22 shows the metrics of the test8.

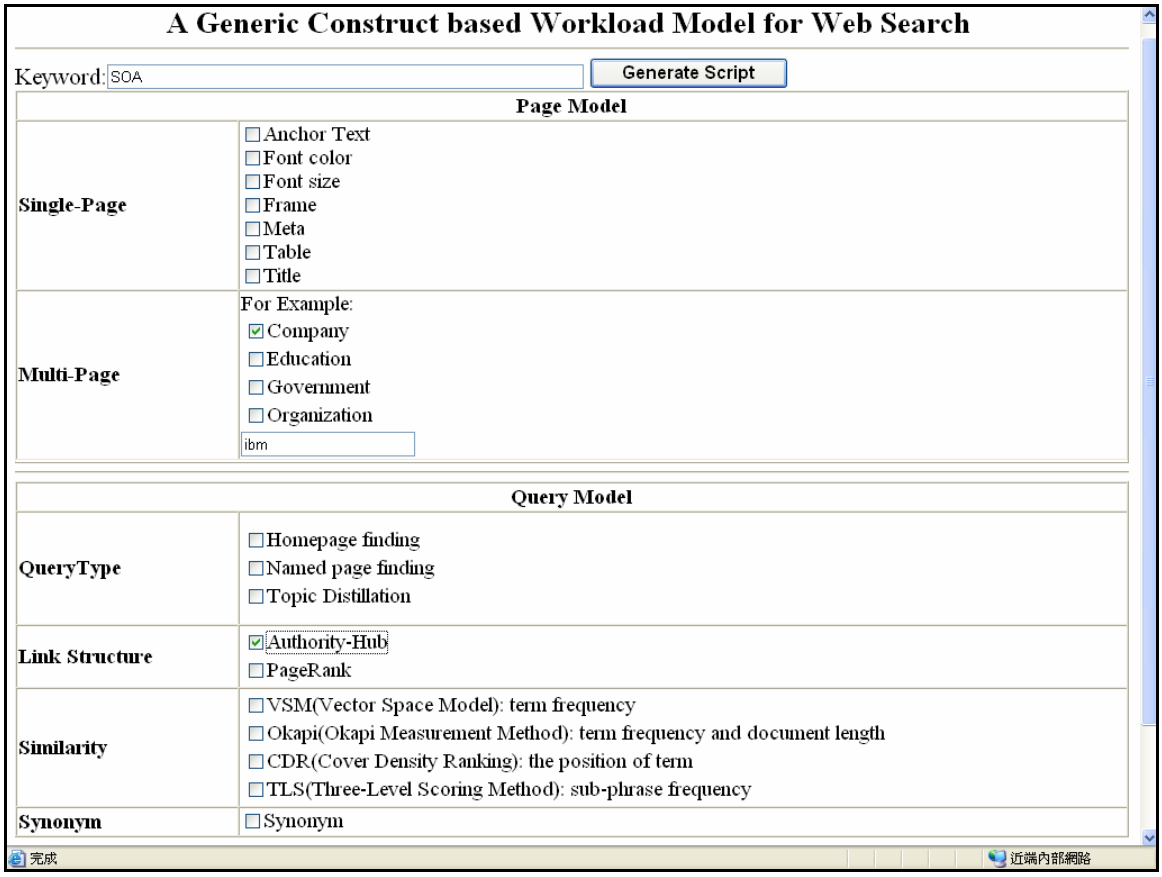

Figure 5.25: Selected Input for Test8

Figure 5.26: Script Output of Test8

Figure 5.27: The Search Results of Test8

Table 5.22: Experimental Metrics of Test8

1.	Response Time	0.406 second
2.	Throughput	2.46305418719212
3.	Precision	0.72
4.	Recall	0.7
5.	TotalResultsAvailable	5080000
6.	TotalResultsReturned	100
7.	The Search Result	As show in Fig 5.24

5.2.9 Similarity-TLS

In the ninth test, Similarity-TLS, we click on “TLS” and” Topic Distillation” and input keyword “Service Oriented Architecture”. We click on “Generate Script” as shown in Figure 5.28. There are two main result screens. Figure 5.29 shows the query script that is generated by the prototype system. The script illustrates the input requirements have been specified for the web search request. Figure 5.30 shows the search result of Test9. “TotalResultsAvailable”, “TotalResultsReturned” and

the lists from “Search Result” are returned by the Yahoo! web search services. “TotalResultsAvailable” is set with “Response Time” and “Throughput” as the main metrics to be automatically generated. Based on the search results, we will check the relevance of each URL and calculate precision and recall. Table 5.23 shows the metrics of the test9.

.

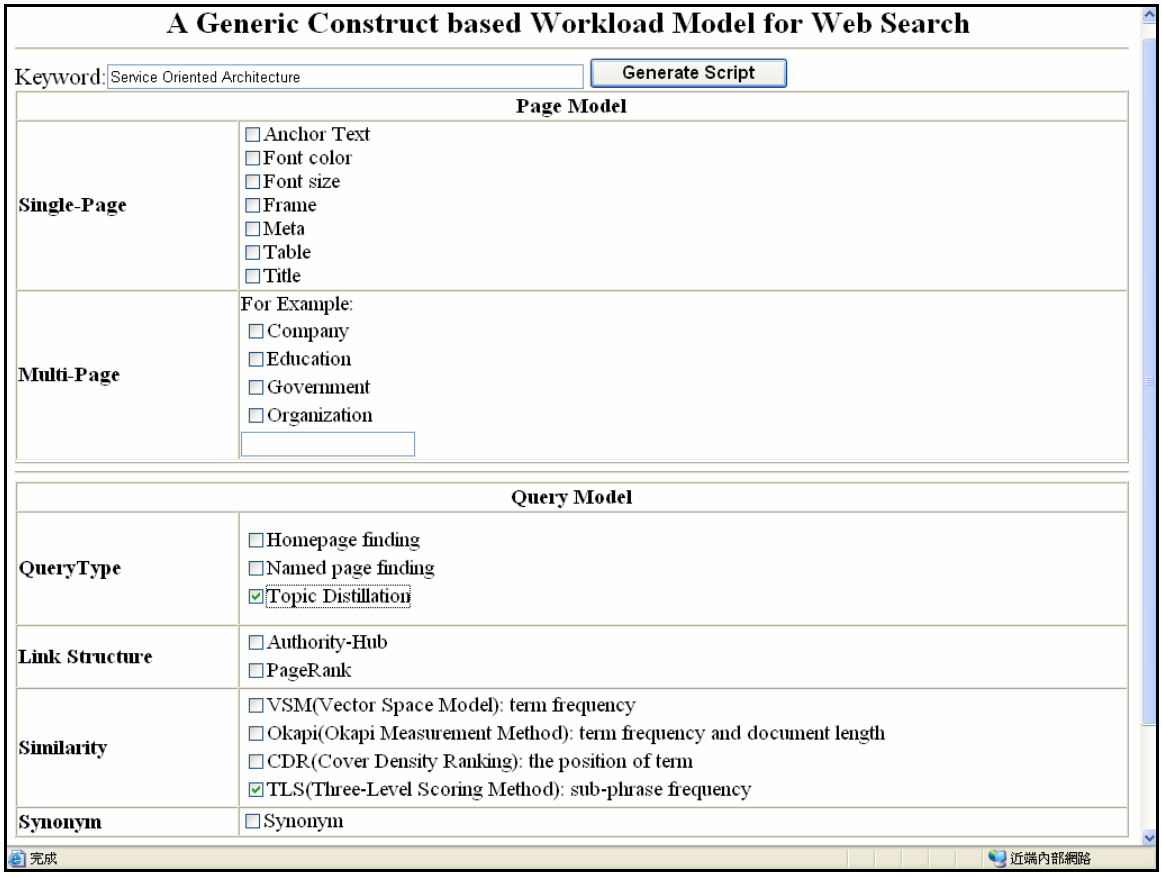

Figure 5.28: Selected Input for Test9

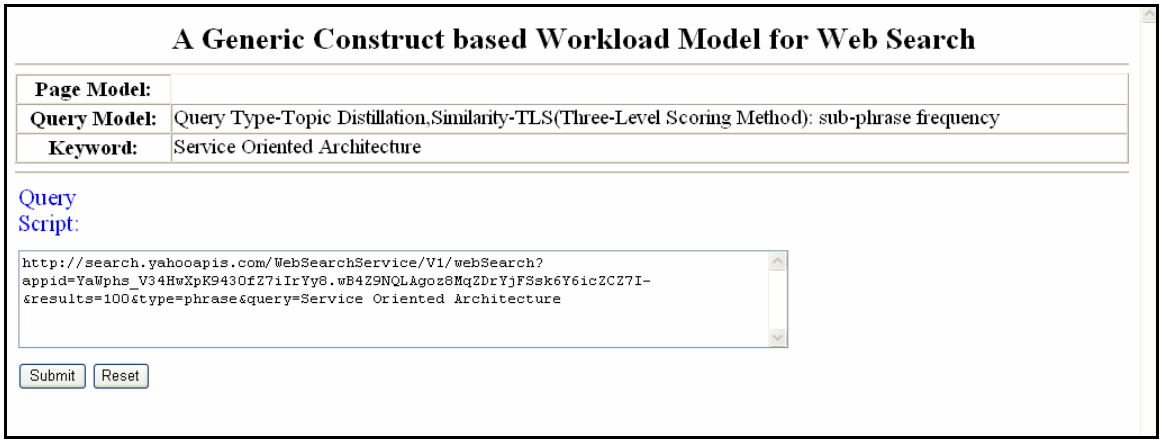

Figure 5.29: Script Output of Test9

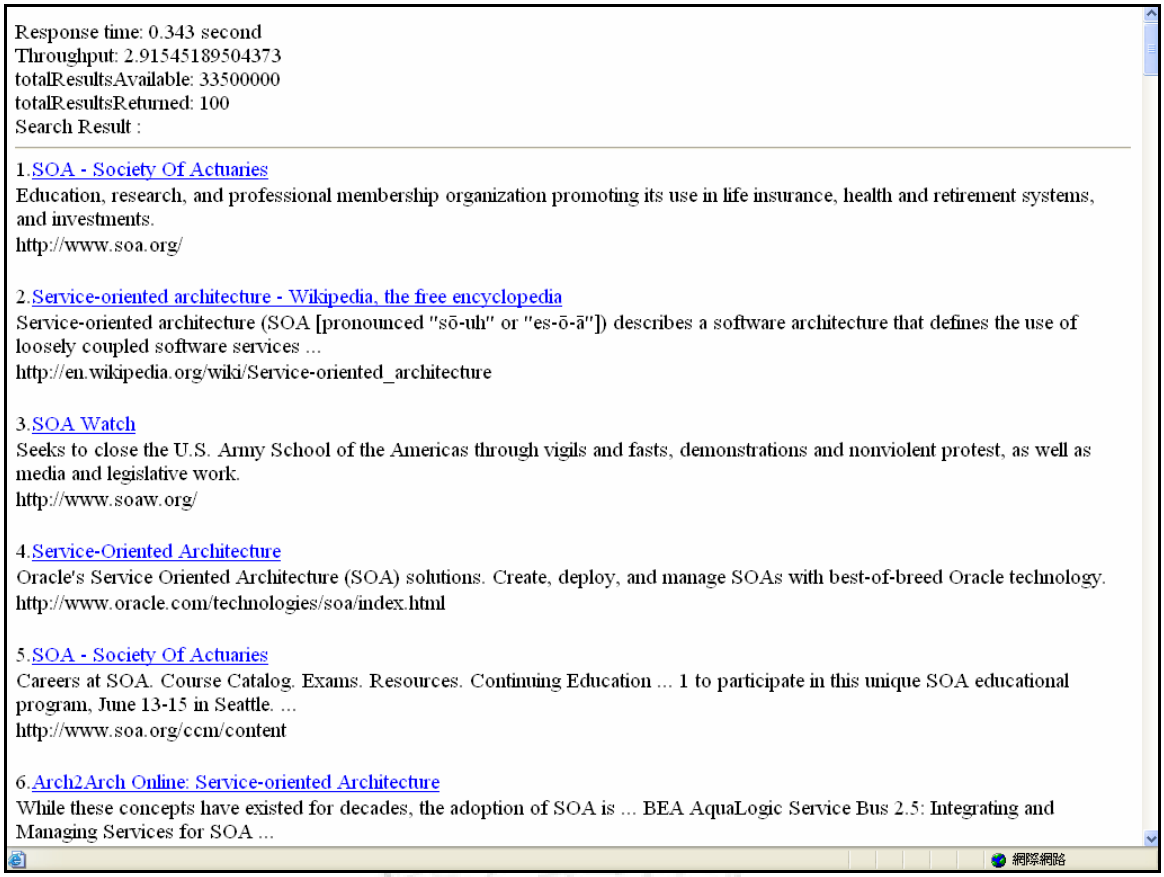

Figure 5.30: The Search Results of Test9

Table 5.23: Experimental Metrics of Test9

1.	Response Time	0.343 second
2.	Throughput	2.91545189504373
3.	Precision	0.735
4.	Recall	0.825
5.	TotalResultsAvailable	33500000
6.	TotalResultsReturned	100
7.	The Search Result	As show in Fig 5.27

5.2.10 Synonym

In the tenth test, Similarity-TLS we click on “Synonym” and” Topic Distillation” and input keyword “SOA”. We click on “Generate Script” as shown in Figure 5.31. There are two main result screens. Figure 5.32 shows the query script that is generated by the prototype system. The script illustrates the input requirements have been specified foro the web search request. Figure

5.33 shows the search result of Test10. “TotalResultsAvailable”, “TotalResultsReturned” and the lists from “Search Result” are returned by the Yahoo! web search services. “TotalResultsAvailable” is set with “Response Time” and “Throughput” as the main metrics to be automatically generated. Based on the search results, we will check the relevance of each URL and calculate precision and recall. Table 5.24 shows the metrics of the test10.

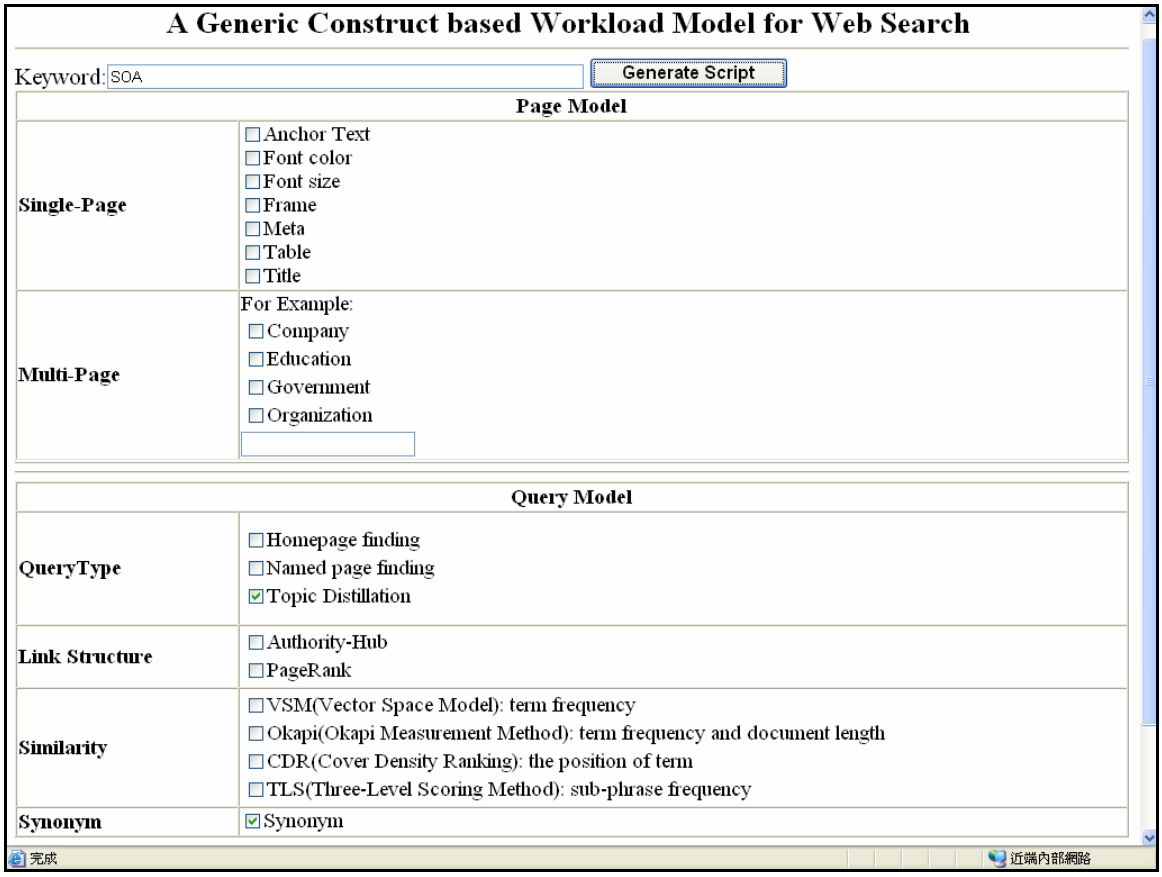

Figure 5.31: Selected Input for Test10

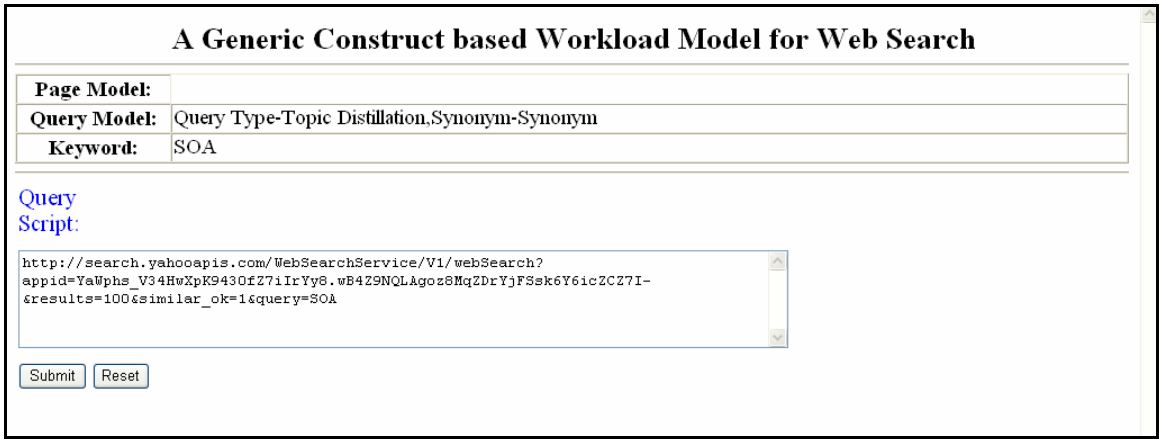

Figure 5.32: Script Output of Test10

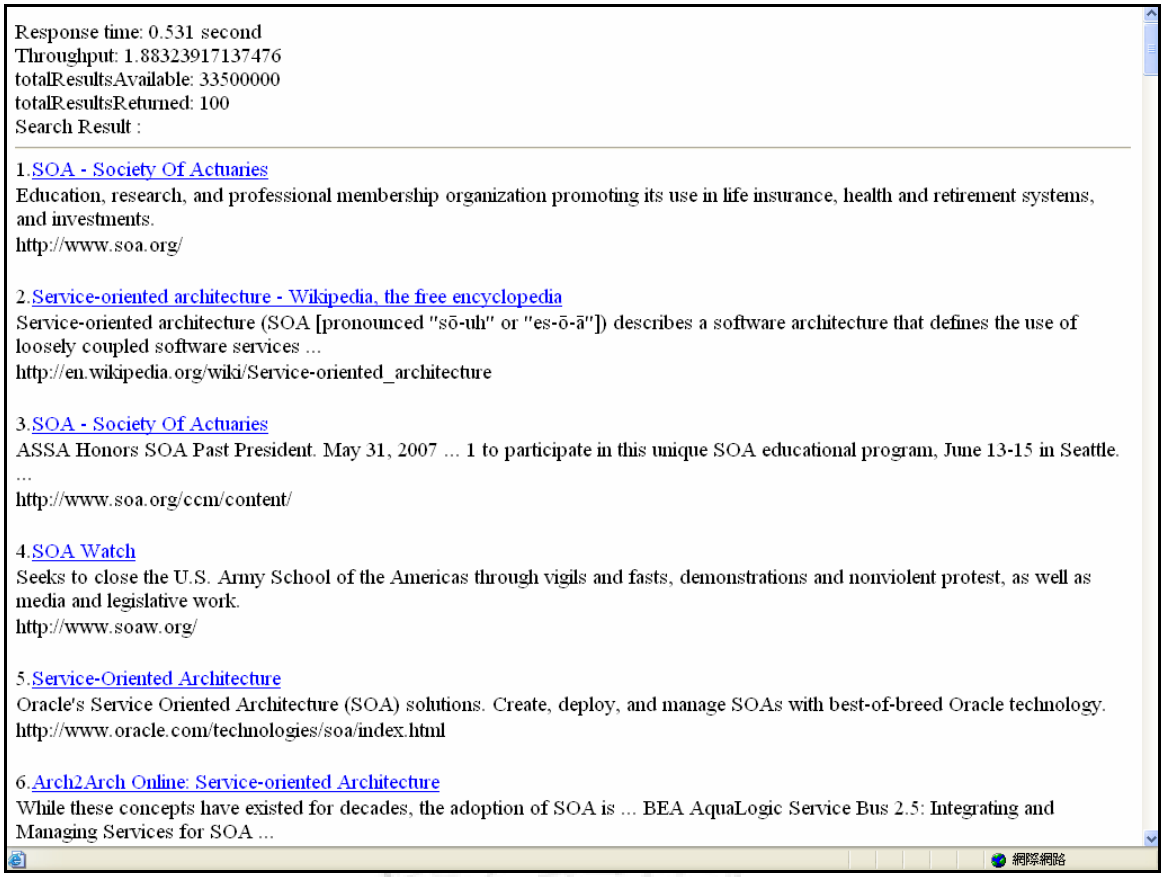

Figure 5.33: The Search Results of Test10

Table 5.24: Experimental Metrics of Test10

1.	Response Time	0.531 second
2.	Throughput	1.88323917137476
3.	Precision	0.375
4.	Recall	0.4
5.	TotalResultsAvailable	33500000
6.	TotalResultsReturned	100
7.	The Search Result	As show in Fig 5.30

5.3 Summary

Some uncontrollable factors can affect the search results on the internet. It would be best to execute all tests simultaneously in sequence because the internet is always changing and growing. The ten test suites are finished by requesting Yahoo! Web Search Services so the tests are executed in a web environment. Web pages are always increasing and the status of network traffic can affect the efficiency of the web search so the relevance assessments must be carried out as soon as the hit lists have been retrieved. Otherwise, the differences in performance metrics between test cases will be meaningless. The performance metrics "Response Time" and "Throughput" are shorter during rush hours. The value of the "TotalResultsAvailable" metric is greater than the search results generated few days previously. In order to maintain consistency of the entire experiment, it is better to execute each test immediately in sequence.

Each test suite provides a different workload to benchmark the web search. And we input the same keyword "SOA" and "Service Oriented Architecture" in each test. Consequently, the ranking of each search result is somehow different. From this, we derive some conclusions on the different rankings. Because a web search is a kind of text search, a page that matches the keyword exactly might not match what users want to find. For example, in the third test" Single Page-Title", the top three results match "SOA" but not "Service Oriented Architecture". In addition, the eighth test "Link Structure-Authority-Hub" performs a text search and searches pages linked to the URL www.ibm.com. This can find pages related to "Service Oriented Architecture" as well as "SOA" and the accuracy of the top five pages from the search result is better than that for the other tests. Obviously, this is the reason why designers of web search engines are always trying to refine ranking algorithms.

An artificial calculation of precision and recall cannot be error free in every case. We need to click on every list of search results to verify whether a page is relevant or not. Although we follow the relevance criteria in the literature, in some cases correct items may be missed or misjudged as artificially working. There are two ways to limit errors. One way is to use the judgments of more than two people and get an average. The other way is to choose the more reliable keyword because it makes judgment of a page's relevance easier.