An Approach For Multidomain Query Optimization And Answering

Heena Agrawal[1], A. M. Karandikar[2],

Student, M.Tech, Computer Science and Engineering, Ramdeobaba College of Engineering and Management, Nagpur, India[1]

Assistant Professor, Computer Science and Engineering, Ramdeobaba College of Engineering and Management, Nagpur,

India [2]

***Abstract*—** In queries consists of multiple domains it is observed that general purpose search engines are unable to answer multidomain queries and out of which one of the domain is selected by specific search services but it is not possible to get an integrated framework.Qqueries which are answered by combining knowledge from two or more domains are Multidomain queries. And the objective is to present a multi-domain queries with an abstract formalism and to develop and optimizing a query system of several search services to choose best query plan to return relavant answers in ranked order. In this paper an approach is for handling multidomain queries on the web. It integrates different kind of services like web service, web search and combination of them. The google search and the web service of dapper is used to refine the results of query as more relevant results. It introduces the well defined model for expressing query results in ranking order. Here the query tends to find the results with cross domain joins to address the solution for optimized answering of multidomain queries. They gives the survey for multidomain query answering with answers as an optimized solution to obtain better search results using web services. The output is gained by chunking the whole query in several domain and then combining the whole result. The query which is chunked go for finding on each domain and then the result which is more relevant to all the domains in multidomain query are selected as an top N results. For getting the optimized and relevant results the sharpNLP tools are used as chunking and POS tagging . After that the best result can be selected. Query optimization and query answering both are done on multidomian query entered in web based system for search and the output is web service and google where web service, dapper go for specific entity selection like url, so that the search will be fast and relevant. The idea in this paper is to collect the best results from the best using web search and service and the relevance is maintained by ranking the results.

Keywords Multidomain query, web search, web service, query optimization, semantic score, reranking, relevance, chunking, tagging.

I. Introduction

Multi-domain queries are the queries which can be answered by combining knowledge from two or more domains. The recent evolution of the Web is characterized by an increasing number of search engines and query interfaces, ranging from generic ones (Google) to domain-specific ones (geo-localization services or on-line catalogs). An increasing amount of search services available on web work in isolation; their intrinsic limit is the inability to support complex queries ranging over multiple domains.

If we consider a query involving multiple domains, such as "find all database conferences held within six months in locations whose seasonal average temperature is 28 ^{0}C and for which a cheap travel solution exists" , requires combining search engines specialized over different domains, for instance: (i) finding interesting conferences in the desired timeframe on online services made available by the given scientific community; (ii) finding if the conference location is served by low-cost flights; (iii) finding if there are luxury and cheap hotels in proximity of the conference location .

Web services are the method of sharing data and functionality among loosely-coupled systems. The research terms that comes in developing and optimizing a query system for multiple-domain web queries stressing on specific features in them is given here. The search using Google is easy but sorting the expected data out of the search results is very difficult. Web search is the general search like google search and also web service is the specific service based search whose goal is to create a framework in which applications distributed across the internet can interoperate through a set of standard protocols like web links or url's, etc. These url's were captured using web service dapper where Dapper is distributed systems tracing infrastructure, and describes how our design goals of low overhead, application-level clearity, and ubiquitous deployment on a large level; system were find. Dapper shares conceptual equalities with other tracing systems. Here it deals with search services where the answers are in ranking order. The algorithms based on PageRank have revealed different results on different data. Page rank is the link analysis that assigns a numerical weighting to each element of a hyperlinked set of documents.

In this way , we mean to develope a multidomain query answering where queries are expressed as execution strategies over web services by doing the scheduling of service invocations. Results are returned in ranking order with reference to the semantic score

or frequency count of number of occurrences of the query keywords into the web based data for that search. Optimized multidomain query answering is done using sharpNLP tools which are chunking and POS tagging. Here query to search is tagged and chunked and then relevant results are abstracted with page rank and frequency count score for tagged keywords.

II. LITERATURE REVIEW

A. System Architecture

The framework given in [1] is for query answering. The architecture is defined in fig.1. The user had a query on the global ontology, equipped with a set of mappings with the services schemata and some integrity constraints. According to the mappings the query is rewritten and the constraints as a query over the services which is transformed into several possible executable query plans considering possible limitations in accessing the services from which the content is extracted.

Framework consists of 3 layers:

Query formulation layer :- This layer allows users to convey their requests to the system by using an interface of global ontology which hides the specificity of the services. The important task of this layer is to rewrite the user query through mappings, whose results are expressed like multi-domain queries over physical services data with access limitations.

Query execution layer :- This layer generates a query plan optimized considering the parameter related to the services and the cost model. This optimization is performed on situations, such as: (i) the types of operations involved in the query plan; (ii) available profiling information on specific services; (iii) ranking of the results.

Data layer :- The data layer addresses the view in the framework of the physical services; they may be either Web Services or wrapped, data-intensive Web sites. With estimates of the figures which are relevant to the optimization problem Services are constantly profiled so as to feed the optimizer of the layer above.

B. Query Optimization

The optimization approach contains in enhancing a highly combinatorial solution space that characterizes all possible translations from the user query into fully initialized query plan. It is divided into three phases, that giving details of query plans [2].

- The first phase is the selection of a given query rewriting such that every service is called with one of the available access patterns. This phase transforms the conjunctive query over web services into several annotated access queries over access patterns of the corresponding services.
- The second phase is the selection of a query plan for the given query rewriting. This phase fixes the order of execution of the query over the services as well as the position and kind of joins between services used in the plan.
- The third phase is the assignments of the exact number of fetches to be performed over the chunked services. This phase allows to fully determine the execution strategy for a query and therefore to compute its semantic score according to number of occurances of query keywords.

C.Joiner Architecture

The focus is to develop techniques for integrating the results extracted from several existing search engines. Designing a system which offers a common interface to several, known search services [4] is our main idea.

Architecture is given in fig.2-

- The input queries are taken in by user interface from the user in the form of terms collections and shows the results as output; allows users to indicate their.
- The query decomposer reduces the original query into many subqueries and identifies the services that can better address each subquery, using search services information coming from the search service directory.
- The joiner selects the join method, and when appropriate interacts with the service caller to request a new block of results from a search service.
- The matcher performs the join of the elements in a given tile, producing the result entries using correlation provided by the directory service.
- The composer forms the entry marks which are sent to the user interface for their publishing.
- The ranking guesser is an optional module that calculates the average ranking of each block of records. These records are given by a search engine from the service.
- The joiner activates the service caller to request from a given search service a new block.
- Web services are registered by registration service into our framework.

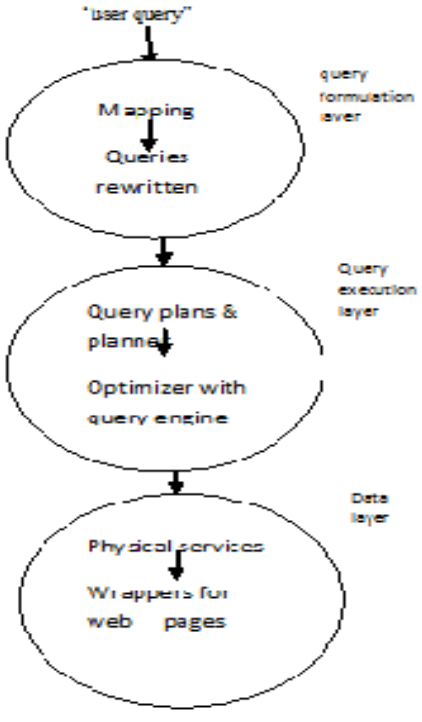
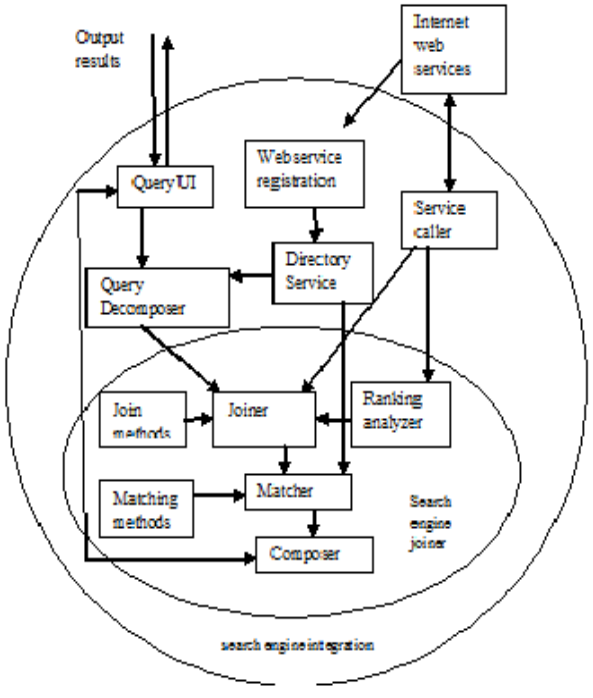


Fig.1, Reference stages diagram



Fig,2, Joiner Scenario

D .Overall Architecture And Execution Flows

In the multi-domain query answering we tabulated *two main flows: the* ***registration flow*** *- that deals with* the declaration and explaination of domains, and the search service registration and their association to domains- and the ***query execution flow*** - that deals with the actual processing of the queries. Fig.3, shows the overall architecture of the system, consisting of the two important execution flows.

The terms which are shown by the activity flows are represented by a conceptual model that describes: (1) domains and their properties (classification taxonomies and associated concepts); (2)search services (request/response interfaces with annotations for in/out parameters and description of response , with functional and nonfunctional properties); (3) high level multi-domain user queries (simplified natural language queries, formed by subqueries); (4)low-level queries (adorned conjunctive datalog queries); (5)query plans (descriptions of strategies for query execution, by acts of coarse-granularity which contains with limitations to access and definining strategies which are ranking-aware for building results); and (6) query execution schedules (well-defined schedules of fine-granularity operations, including service invocations, which have the execution control flow) .

In registration flow, following problems are addressed: (a)semantic representation, storage, management, and access to domains and their descriptions; (b) semantic description, storage, management, and access to search services; (c) clustering of services based on similarity; (d) mapping of services to domains; and (e) definition of admissible join conditions between services.

In the query execution flow we address the following problems:(f) definition of proper interfaces for submission of multi-domain user queries; (g) splitting of the query into subqueries; (h) mapping of subqueries to domains; (i) mapping of subqueries on given domains to associated search services, which defines queries of low level; (j) generation of query plans and their evaluation. In front of many cost metrics to choose the most promising one for execution; (k) generation and processing of query execution plans; and (l) transformation and rendering of the results for user consumption.

E. Search Service

Searches may be of two kind, web search (google) or web service search. Search services are to enable the annotation of the request/response services interfaces. Here we are interested in operations belonging to a Web service which perform data retrieval, and actually in operations that return itemized and ranked information. The service analyzer reflects the clustering of the available services, on seeing their

similarity; the definition of join connections between services; and the mapping of services to domains.

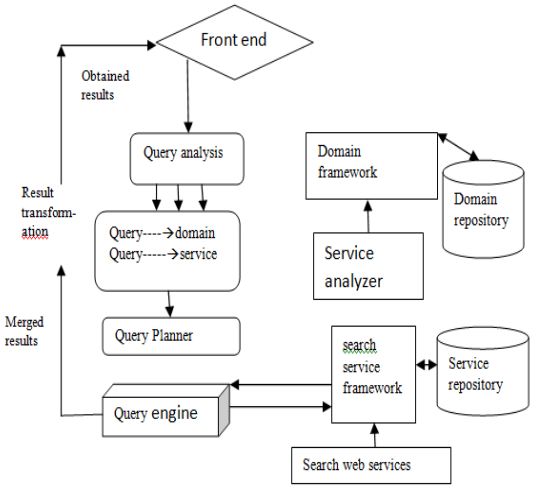


Fig.3, Overall execution steps

F. *Query to Domain And Service Mapping*

To analyze the query, method consists of applying a first splitting of the sentence, and then to check whether the formed subqueries map consistently to separate domains, by invoking the Query-Domain mapper [5]. If incoherent mapping is we conjecture that the splitting is not well enough, and therefore we: (i) ask for feedback from the user; or (ii) try a different splitting based on cohesion of words w.r.t. domains. The final result of the splitting in (high-level) subqueries is therefore just a first step towards the mapping of subqueries to domains.

This component addresses the problems of mapping subqueries to domains and of mapping subqueries to associated search services, which is to define low-level queries. The process in which a mapping of a query to a domain can be successful only if: (i) each subquery comprises only requests to one domain; and (ii) the words used in the subquery are definite, thus allowing a hard identification of their semantics (and therefore a correct
mapping to the domains) [5].

Several methods can be conveyed to optimize the recognition of query-subquery structures which comply with the separation into distinct domains of concern so as to achieve the objective (i); these include [5]:

- iterative invocation of the NLP tool based on defined lexical interpretation obtained from review from user, or review from other components;
- sustainment of explanation of search services or domains for assessing the correctness of the query splitting;
- syntax/logic analysis result is done On sentences.

G. *Semantic score and Relavnce and Reranking*

The count of occurances of any keyword of the query into the related web searches is Semantic score.

Relevance is the semantic similarity between keywords and a specified web document. Our algorithm is different from other semantic similarity methods. We estimate the similarity between the keywords and each word in the document (ofcourse removing the stop words) to get the final relevance[8].

As we know, the search results are returned by search engines according to their importance and relevance. Generally the web pages which are most are returned at the upper positions, and attract much more attention from users.

H. *Present Google based approach and proposed approach*

In present google search based approach the search results are found but they seems accurate but it cannot go for specific search like for urls or any aentity of search even not with one specific search service. But it can be done as required by doing combinational task of search and service i.e. google search and web service. This results in output consist of specific search results. It is the further research approach that how to build a service which organizes needed architectures, mapping and optimizations to answer the multidomain query in particular execution flow having search results with more relevance factor. This work results in optimized answering of multidomain queries.

- Google search is a normal search which gives complete information not a specific one
- The overall detailed links are given here which let to go for complete page for input text.
  This is shown in following output screen of fig.4.

- web-based service is that which allows you to extract and use information from any website on the Internet. It allows you to create a "feed" (which we call a Dapp) for any site without programming works exactly as your browser does and reads content from a website's page. The difference is that in place of displaying the entire page, service streams a feed of just the content you select.
- A Dapp is the definition of the content you want from a certain page type.
  This service work is shown in following fig. which goes for one decided entity of web result.

III. PROPOSED WORK

On going through the literature review it is extracted that how the query flows from entry to exit with a result. But the issue is for finding the best results for any web based systems query answering. In this the optimization is connected to plan the query with multidomain which relates to domain and the keywords which supports the explaination for domain knowledge. In above review, the process gives the all steps for query execution with optimization. The paper proposes the method for multidomain query anwering with optimization and listing them in best relevance order. For this web based execution is done here. Using web search and web service with combination of them and applying chunking and POS tagging the ranking procedure proposes the relevant answering of multidomain query.

It is considered that for web search the google search is used and for web service dapper is used. Here Google Search (or Google Web Search) is a web search engine whose main purpose is to hunt for text in publicly accessible documents offered by web servers. Google's rise to success was in large part due to a patented algorithm called PageRank that helps rank web pages that match a given search string. Google is the most popular search engine, which is eager to influence their website's Google rankings. The general idea is to affect Google's relevance algorithm by incorporating the keywords being targeted in various places "on page", in particular the title element and the body copy. In this way the framework of google search works this approach and the google is used as first step towards approach. After this the next is web service which is dapper. Dapper is a free, web-based service that allows you to extract and use information from any website on the Internet. It allows you to create a "feed" (which we call a Dapp) for any site without programming. These feeds can be used then in a variety of ways: like in RSS feed, a widget, in Facebook applications, and websites. Dapper works exactly as your browser does, it reads content from a website's page. The difference is that in place of displaying the complete page, Dapper streams a feed of the content only, you select. A Dapp is the defined as the content you want from a certain page type. For example, create a Dapp that reads just the movie titles and thumbnails from a video website search results page, and streams them as an RSS feed. After that when Dapper tends to that website, it uses the Dapp to know which content to read and send in the feed. In this way the dapper is used to extract selected type if information from the search and attach that results from google results and then combine the results of google and dapper so that the from google even the better results can be extracted and dapper gives the result say for url links only and we list those links as dapper results and the google results are also listed say we go for top N results.

But how the multidomain query enters into the optimized query answering is important. Here the query is entered and it goes to use sharpNLP tools which are chunking and POS tagging. Chunking is an analysis of a sentence which identifies the constituents (noun groups, verbs, verb groups, etc.), but they won't specify their internal structure, not their task in the main sentence. Chunking speaks for the level of specificity, to reach for higher meanings or search for more specific bits/portions of missing information. Chunking splits the sentence into words, i.e., non-overlapping regions of text. Part-of-speech tagging (POS tagging or POST), which is also called as grammatical tagging or word category disambiguation, it is the method of marking up a word in a text (corpus) as corresponding to a particular part of speech, depending on its definition, as well as its context means relationship with adjacent and words in a phrase, sentence or paragraph.

For listing the results as top N results the ranking algorithm is used. Here it gives ranking to pages by how often the search terms observed in the page, or how strongly connected to the search terms were within each resulting page. By this the occurance of number of times the particular keyword of any query analyzed in query is termed as frequency count or semantic score. Here the results of google and dapper are combined after calculating their frequncy count for multidomain query and the results with highest count are considered as top N results.

IV. IMPLEMENTATION

Many times we come across multidomain query where the main requirement is relevance and quality search. In this proposed approach it is considered that how the results from google can be modified on the grounds of relevance using web service known as dapper using sharpNLP tools like POS tagging and chunking. The objective for this approach is to implement multi domain-specfic search, Integrate different kinds of services, offers search service to return answers in ranked order with more relevant data from query plan. Web service is used to get multidomain data means unrestricted domains will be there.

Steps involved in approach are—

- Google search results and web service (dapper) results
- Combination of web search and web service which lists the relevant outputs with semantic score

- Analysis of semantic score of outputs as per their semantic relevance with on normal search service using sharpNLP tools.

- Using results of google , dapper and search service as database apply reranking on database with respect to their semantic score.

For first step, current evolution of the Web is characterized by an increasing number of search engines and query interfaces, ranging from generic ones (Google) to domain-specific one. Now wrapping technology is evolving so as to enable the development of specialized services extracting content from data intensive Web sites and exposing them as Web Services. For this google search (web search) and dapper (web service) is used which gives search results in two different formats.

Google search is a normal search which gives complete information not a specific one. The overall detailed links are given here which let to go for complete page for input text. Dapper is a free, web-based service that allows you to extract and use information from any website on the Internet. It allows you to create a "feed" (which we call a Dapp) for any site without programming. Dapper works exactly as your browser does, it reads content from a website's page. The variation is that as an alternative of showing the whole page, Dapper gives a feed of the content you select. A Dapp is the definition of the content you want from a certain page type.

Further step is divided into the terms as combined result of google and dapper search and list of relevant semantics into the link with top N results. Here the combined search is performed means search results will be from both google and dapper. Results obtained from google and dapper search are good source of optimized search results. So when we add both these information sources together then there is a new combination of search service. Here the web links and URL''s from google and dapper search resp. are displayed in a single list by selecting top N results.

In next part relevance and frequency score is observed. On finding the relevance search on these links the semantic score is found. Relevance is the semantic similarity between keywords and a specified web document. We calculate the similarities between the keywords and each word in the document (of course removing the stop words) to get the final relevance. When a set of semantic features is presented, the overall relevance results from the sum of the individual relevance values associated with each of the semantic features of different domains. On the basis of number of occurences of different words in relevance the semantic score for each word is counted.

SharpNLP is used to find relevance. From SharpNLP tools chunking and pos tagging are applied. Chunking is an analysis of a sentence which identifies the constituents (noun groups, verbs, verb groups, etc.), but does not specify their internal structure, nor their role in the main sentence. Chunking speaks for the level of specificity, to reach for higher meanings or search for more specific bits/portions of missing information. Chunking splits the sentence into words, i.e., non-overlapping regions of text. Part-of-speech tagging (POS tagging or POST), also called as grammatical tagging or word category disambiguation, is the method of entitling up a word in a text (corpus) as corresponding to a particular part of speech, based on its definition, and its context—i.e. Relationship with adjacent and words in a phrase, sentence or paragraph.

In the next step of reranking, the list of semantic scores are settled in descending order .On the basis of semantic score of refined outputs the ranking of listed candidate results are done in descending value. Then from them the top N are extracted and they are displayed as a final optimized Query output i.e. Final search results.The user expects results in ranking order; so, by composing answers using multiple services, we must produce a global ranking that is a good composition of the various partial rankings, and use the global ranking in producing the output in reranked order.

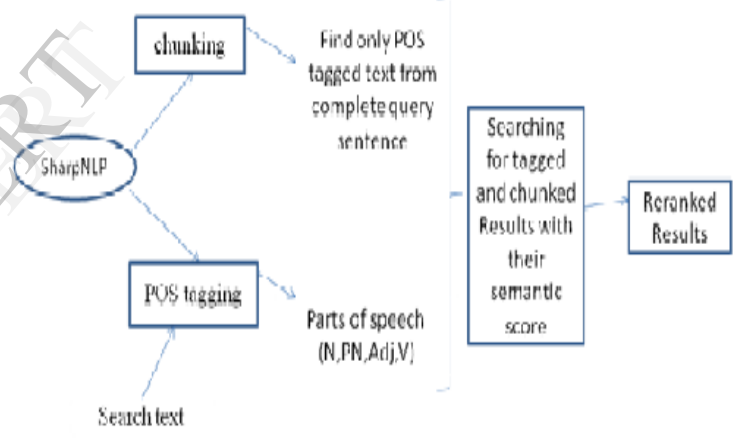


Figure.4, Flow for SharpNLP Tools

V. EXPERIMENTAL RESULTS

As the phases got implemented in ecah step, the final result is obtained for developing the multidomian query optimized answering which result in relevant output. Yhis results need some steps to get performed resulting into proposed approach's satisfaction. Steps are:

- showing multi-domain queries consisting of an abstract formalism
- Developing and optimizing a query system of several search services to choose best query plan to return relevant answer in ranked order

- to implement multi domain-specific search
- integrates different kinds of google search and dapper service
- offers search service to return answers in ranked order with more relevant data from query plan using SharpNLP tools elevant and best frequency scored which results.

All this experiments are done and established in this phases. After this the result obtained is with all objectives gained and the final result is evaluated manually using user experiences.

Result of phase-1 are "Google search results and web service (dapper) results" is-

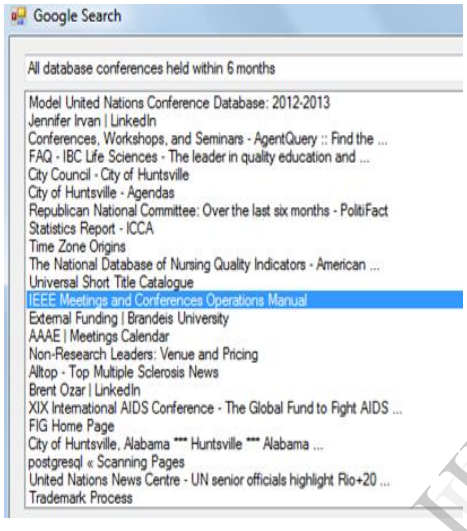


Figure.5, Google result

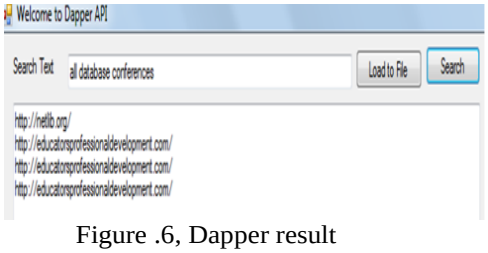


Figure .6, Dapper result

Result of phase-2 are "Combination of web search and web service which lists the relevant outputs with semantic score" is—

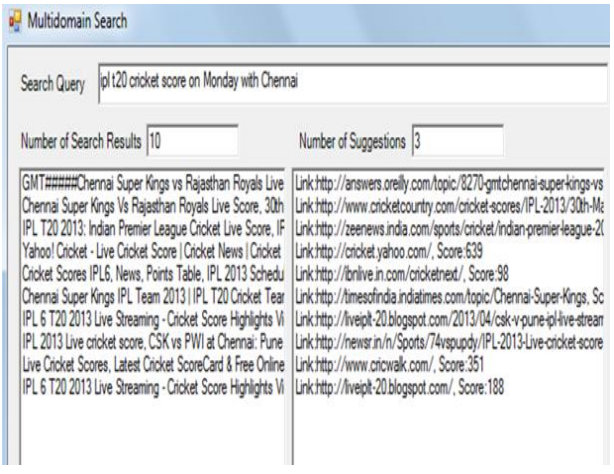


Figure.7, Google + Dapper Result

Result for phase-3 "Analysis of semantic score of outputs as per their semantic relevance with on normal search service usin sharpNLP tools" and phase-4 " Using results of google , dapper and search service as database apply reranking on database with respect to their semantic score" means final result is—

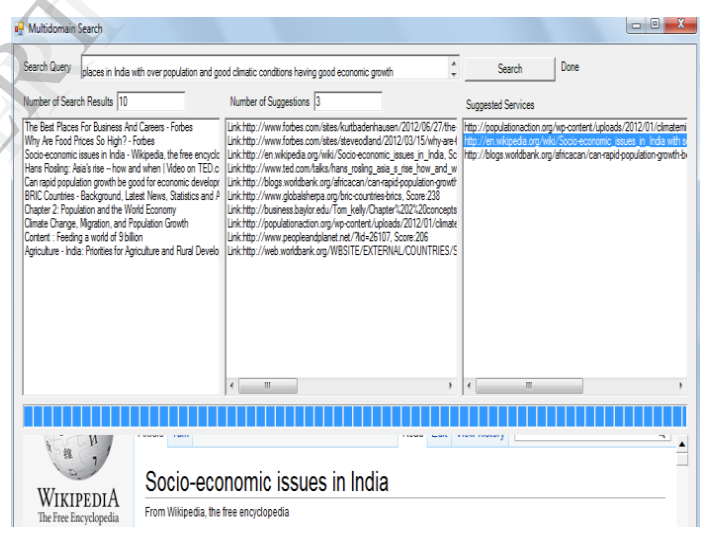


Figure.8, Final relevant result

VI. CONCLUSION

This paper presented set of terms need to be addressed when addressing multidomain queries. It gives the terms, architectures and methods to be consisted. It encourages the u web search and services uses to optimize the results resulting into relevant output. SharpNLP tools have participated here in better search results finding. Frequency count of multidomin queries keywords are considered as an important issue.

In our future work, we envision offering to users a more relevant and optimized multidomain

query answering and more relevance can be addressed. The evaluation can be enhanced to go for improved techniques for finding better multidomain results. In this way, the multidomain search will get existence to contribute in advanced search techniques.

REFERENCES

[1] D. Braga, D. Calvanesey, A. Campi, S. Ceri_, F. Daniel, D. Martinenghi, P. Merialdoz, R. Torlonez, "NGS: a Framework for Multi-Domain Query Answerin," .

[2] Daniele Braga, Stefano Ceri, Florian Daniel, Davide Martinenghi, "Optimization of MultiDomain Queries on the Web," PVLDB '08, August 23-28, 2008, Auckland, New Zealand

[3] U. Srivastava, K. Munagala, J. Widom, and R. Motwani, "Query optimization over web services," in VLDB-06, 2006, pp. 355–366.

[4] D. Braga, A. Campi, S. Ceri, and A. Raffio, "Joining the results of heterogeneous search engines," Information Systems, Proceeding of the 32nd VLDB Conference, Seoul, Korea, 2006.

[5] Davide Barbieri, Alessandro Bozzon, Daniele Braga, Marco Brambilla, Alessandro Campi, Stefano Ceri, Emanuele Della Valle, Piero Fraternali,Michael Grossniklaus, Davide Martinenghi, Stefania Ronchi, Marco Tagliasacchi , " Data-driven optimization of search service composition for answering multi-domain queries," VLDB '09, August 24-28, 2009, Lyon, France.

[6] Giorgio Ghisalberti, Marco Masseroli, Salvatore Vadacca, " Multi-Domain Data Search and Retrieval:A Service-Oriented Life Science Scenario "Dipartimento di Elettronica e Informazione, Politecnico di MilanoPiazza Leonardo da Vinci, 32, 20133 Milano, Italy.

[7] Ruofan Wang, Shan Jiang, Yan Zhang, "Re-ranking Search Results Using Semantic Similarity," .

[8] S.G.Choudhary, S.R.Kalmegh, Dr. S. N. Deshmukh, "Semantic Search Algorithms based on Page Rank and Ontology: A Review," 3rd International Conference on Intelligent Computational Systems (ICICS'2013) January 26-27, 2013 Hong Kong (China)

[9] http://open.dapper.net/help.php

[10] S.G.Choudhary, S.R.Kalmegh, Dr. S. N. Deshmukh, "Semantic Search Algorithms based on Page Rank and Ontology: A Review", 3rd International Conference on Intelligent Computational Systems (ICICS'2013) January 26-27, 2013 Hong Kong (China).

[11] Benjamin H. Sigelman, Luiz Andr´e Barroso, Mike Burrows, Pat Stephenson, Manoj Plakal, Donald Beaver, Saul Jaspan, Chandan Shanbhag, "Dapper, a Large-Scale Distributed Systems Tracing Infrastructure", Google Technical Report dapper-2010-1, April 2010

[12] Sandipan Dandapat, "Part-of-Speech Tagging and Chunking with Maximum Entropy Model",Department of Computer Science and Engineering Indian Institute of Technology Kharagpur India 721302