SEMANTIC SEARCH SYSTEM USING ONTOLOGY FOR COMPUTER LANGUAGES

Bhavesh Gandhi

Abstract

The World Wide Web allows people to globally share data from large repositories. Retrieving desired data/information is a challenging task for the search engines. Some existing search engines apply keyword search technique while others retrieve partially semantic results. The data accessed by the former is quick, but a lot of it is irrelevant and that by the later is semantically consistent to some extent but the response time is high. The proposed project adds a semantic dimension to the search system by a domain specific ontology. The ontology is well maintained in terms of concepts and relations. The system implements data retrieval technique similar to the one applied by the keyword search engines. Thus the proposed system is an advancement to the existing systems and meets both the requirements of being semantic in nature and quick to respond. The system accepts the user's need in the form of query, parses it, rephrases parsed query using ontology and then finally fires it to a keyword search engine. The retrieved links are filtered and reordered based on their relevance. The system also provides suggestions depending on query terms and intellisense. Thus the proposed project intends to develop such a system for major programming languages like C++, java etc.

***Keywords:*–Information Retrieval, Query Processing, Semantic Search.**

I. INTRODUCTION

With the development of the Web, an “Information Big Bang” has taken place on the Internet. Search engines have become the most helpful tool for obtaining useful information from the Internet. However, the search results returned by even the most popular search engines are not satisfactory. The search engines return a lot of pages that have to do nothing with the user’s need. This is because search engines return web pages just because they contain the keywords entered by the user. The user has to look into the results to find the relevant one satisfying the user’s requirement. As this process is often time consuming, the solution to it is to develop a system which gives results relevant to the context of the user’s query.

Semantics is the study of the meaning and relation of words together. When applied to search, it allows a search engine to return results depending on the meaning implied. Semantic Search seeks to improve search accuracy by understanding the user’s intent and searching it in the ontology Semantic search highly improves search accuracy of the query and the search engine delivers the exact content that the user intended to know. By using semantic search engine we will ensure that it results in more relevant and smart results. The point of semantic search is to use meaning to improve the user's search experience. Currently there are semantic search engines which deal in different domains like:

1. Lexxe deals in food, cars, disease
2. DuckDuckGo deals in e-commerce
3. Cognition Search deals in Enterprises

Till date there is no semantic search system which deals in the domain of Computer languages. Our System addresses this particular domain. Our system takes user query as input and returns the list of website links which are more relevant to the user’s intended search. Our system uses ontology for extracting the results related to the query. Ontology is an explicit specification of a conceptualization. Ontology is a description of the concepts and relationships that can exist for an entity or a group of entities. Ontologies are built by identifying various relationships among the concepts and the objects involved. User queries are processed by referring to this ontology.

II. PROBLEM FORMULATION

A. *Problem Definition*

Given a query by user to the system, the system should parse the query and rephrase it according to system Ontology. The rephrased query should be fired to Google API to retrieve links semantically and update system database.

.

B. *Objective*

The results retrieved from the keyword based search system are less relevant semantically. The proposed problem introduces a semantic layer over the existing web so as to refine the results semantically according to the users' requirement.

III. SET THEORY

1. Let 'S' be the Semantic search system.

   S= {…………

2. Identify the inputs as Query Q.

   S = {Q,..

3. Q is query fired by user (C++ domain).

   Q = {q1…qi} where Q ≠Φ, qi is term constituting the query and i<=11.

4. Let 'O' be the output.
5. S = {Q,O …

   O: O is the set of links

6. Identify the processes as P.

   S= {Q, O, P…

   P= {Qp, Qo, U, D, G}

7. Qp is the parsing function.

   Qp= {W, M, Y1, Y}

- W is stop word removal function

  W = {w | w ε qi and w ε WDB.} where WDB is stop word database.

- M is stemming function

  M= {m| m ε qi and m ε MDB.} where MDB is stem word database.

- Y1 is the output of W
- Y is the output of M (i.e. Qp) where (Y=Y1-Y1m, Y1m is set of stem words ε (Y1 ∩ MDB)).

8. Qo is ontology function.

   Qo= {L, R, X}

9. L is Ontology Lookup function and

   L= {T, B, Z}

- T is set of technical term where T ε Yi and T ε TDB (Technical Database).
- B is relation inferred by Ontology where B ε BDB (Relational Database).
- Z is Boolean where if Z=True Result found in DB else link is not found for {T, R} where {T, R} ε DB.

10. R is retrieval function

    R= {r1, r2} r1: retrieval from DB if Z=True, r2: retrieval from G if Z=False.

11. X is relation between function.

    Xi = f (tj, tk)

Wheretj, tk ε TDB and xi ε RDB.

12. U is Update Function responsible for updating the ranking of the selected Ontology.

13. D is Display function.

    D= {E, K, F}

- E = {e1…ei} where ei is the semantic suggestion.
- K= {k1…kj} where kj ε ei and is the Web link.

14. G is Google Application Interface (G API).

    G is activated when Z=False and acts on Y.

15. Identify failure cases as F

    S= {Q, O, P, F …

    Failure occurs when –

- F={Z1}

  Z1: Ontology not found in DB i.e. {T, B}!=DB.

  Z1 ∩O = Φ

16. Identify success case (terminating case) as V

S = {Q, O, P, F, V…

V ε {E, K}

17. Initial conditions as S0

S= {Q, O, P, F, V, S0 …}

S0: Working Internet Connection

IV. System Architecture

SystemArchitecture containsfollowing Components:

1. GUI
2. Query Processor
3. Parser
4. Ontology Manager
5. Analyzer
6. Google API

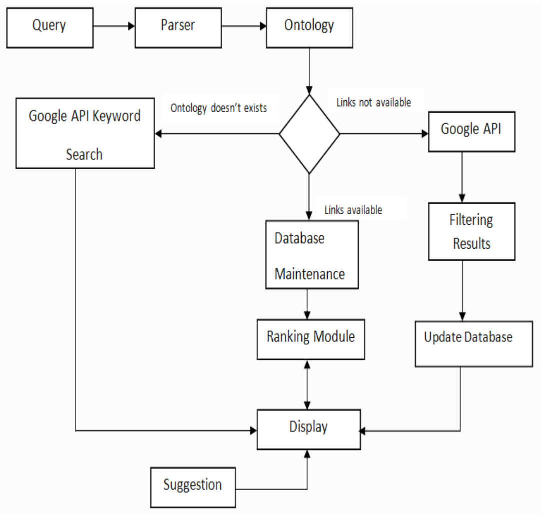

Fig1System Architecture

Description for GUI

Graphical user interface or GUI is used to take query from user and then pass it to query processor. Query is assumed to be in English. Results for the asked query will be displayed on GUI and User will select the required link by interacting (clicking) with GUI.

Description for Query Processor

Query processor will take user query as input, tokenize it and then send it to Parser for further operations.

Description for Parser

Parser takestokenize user query as its input. It has Stop word Remover removes stop words i.e. words which add no weight to query, from the query.Stop word Remover acts on the tokenize query which is fed by Query Processor to it. Every token is matched with the words stored in Stop word Database and is removed from the tokenize query if match is found.

Description for Ontology Manager

Ontology manager acts on refined query which it receives from Parser.Ontology Manager is divided into two sub components, namely,

1. Ontology lookup
2. Retriever

Ontology lookup will identify the technical and relation terms from the refined tokenize query. Once the tokens are identified , then it will look up for the ontology between the technical terms and select the best possible relation which it can infer based on the user query. Control is passed to Retriever if links corresponding to the selected ontology exists in the database.

In case if tokens are not identified successfully or links corresponding to identify ontology is absent then it will give call to Google API by passing the query to API and control is passed to GAPI.If links corresponding to the selected Ontology exist in Database then Retriever will retrieve the links from the selected Ontology and send it to GUI.

Description for Analyzer

Analyzer analyzes the links forwarded by GAPI and store most relevant links related to given query in database for the selected Ontology. Analyzer also maintains the Ranking of the links depending on its popularity.

GAPI

Google API is an external component used to search for the existing and non-existing ontology.

V. Expected Results

The proposed system should retrieve semantically relevant links. The system should also guide the user to phrase queries by displaying suggestions related to input in the form of intellisense.

VI. Conclusion

There areseveral existing Semantic Search engine designed specifically for getting better results. These systems process and retrieve results in a specific format. Given the amount of data in the web, it is not feasible to store entire data in a format useful to retrieve it semantically. The proposed system introduces a semantic layer which processes the web in its existing format but increases the efficiency of the system by retrieving user intended results.

References

[1] G.Madhu, Dr. A.Govardhan, Dr .T. V. Rajinikanth, “Intelligent Semantic Web Search Engines: A Brief Survey” International journal of Web & Semantic Technology (IJWesT) Vol.2, No.1, January 2011.

[2] Hakia Team, “Semantic Search Technology”, White Paper-hakia, January 2010.

[3] A Lexis Nexis Team, "The Evolution of Semantic Search on the Web", International Conference on Intelligent Computation Technology and Automation, 2010.

[4] Renato Ianella, “Semantic Web Architectures”, September 2010, ISBN: 1740648005.

[5] Yi Zhang, Wamberto Vasconcelos, Derek Sleeman, “Onto Search: An Ontology Search Engine1”, Department of Computing Science, University of Aberdeen, Aberdeen, AB24 3UE, Scotland, UK.

[6] Elias Iosif and Alexandros Potamianos, “Unsupervised Semantic Similarity Computation using Web Search Engines”, International Conference on Web Intelligence, 2007 IEEE/WIC/ACM.

[7] Steve Chien, Nicole Immorlica, “Semantic Similarity between Search Engine Queries Using Temporal Correlation”, 2007.

[8] Ning Gao, Zhi-Hong Deng, Jia-Jian Jiang, Hang Yu, “Maxlca: A new query semantic for XML keyword search”,Key Laboratory of Machine Perception (Ministry of Education),School of Electronic Engineering and Computer Science, Peking University, Received February 16, 2011, Revised January 16, 2012.

[9] Sara Cohen Jonathan Mamou Yaron Kanza Yehoshua Sagiv, “XSEarch: A Semantic Search Engine for XML”, School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem 91904, Israel.

[10] Gjergji Kasneci, Fabian M. Suchanek, Maya Ramanath,“How NAGA Uncoils: Searching with Entities and Relations”, 2007.