Matching Based Algorithm for Semantic Search and its Implementation

M. Ilyas[1], S. M. Adnan[2], W. Ahmad[3], T. Fatima[4], U. Habiba[5], J. Rashid[6]

[1,4,5]Computer Science and Information Technology Department, University of Sargodha, Sargodha, 40100, Pakistan
[2,3,6]Computer Science Department, University of Engineering and Technology Taxila, Taxila,47080, Pakistan
[2]syed.adnan@uettaxila.edu.pk

Abstract-With rapidly increasing information and data available online, end users are not satisfied with search results of even renowned search engines. Most of the time, search engines results in irrelevant pages to the users. Semantic search aims to improve search results by retrieving data based on keywords and its synonyms as well. It focuses on intent and context of query. Semantic search is also used to get variety in results. This research work discusses and compare the architecture of various semantic search engines based on how much semantics they include in their results. We proposed a new semantic search algorithm based on ontology matching within the semantic web. Prototype for proposed algorithm and its results are discussed.

Keywords-Semantic Search, Ontology Matching, Semantic Results, Semantic Matching, Semantic Search Engine.

I. INTRODUCTION

Today, internet is basic need of the people. People use internet to get information, entertainment, and communication. To retrieve data against query we need search engine. On the World Wide Web [i] search engines are programs that can be accessed from anywhere. They retrieve the sites against the keyword that contain the relevant information which user needs. So, as we know there are many search engines most common are Google, yahoo, Bing etc. They have implemented search in different ways. For example, keyword, location, semantics based etc. Here our main target is semantic search engine [ii].

World Wide Web (WWW) let the people to use information from the large databases. Amount of information is increasing day by day. We need to search information through special tools called "Search engines" [iii]. There are two main types of search results: navigational and research [iv]. In navigational searches semantic search cannot be applied. Research search can be applied in Semantic search engine. In semantic search user provides the search engine with word and the users get variety result including synonyms as well [v].

In this research paper Section-I describe Introduction of semantic searching. Section-II describes the background of the domain where Section-III describes processes and architecture of semantic search engine. Section-IV describes detailed algorithm. Section V describes its results also comparison of famous search engines yahoo, Google and Bing results. Section-VI describes conclusion and future work.

II. BACKGROUND

Starting from what is search engine. A Search engine is really a collection of programs. We have lot of search engines which we use daily most popular Google, Bing, Yahoo etc. They help people to find information on the other sites. A search engine is also a kind of special site on web that is use to access other sites. It locates other sites based on keyword in the query of user. Different search engines use different mechanisms for their working. But commonly all perform below four processes:

1) Crawling the Web.
2) Indexing the pages.
3) Ranking the pages.
4) Displaying the results.

Search engine is the most powerful tool to get any kind of information in World Wide Web [vi]. Here focus is on semantic search engine. Now upcoming search engines will not be based only on word index [vii]. Semantic search is a searching method which searches query according to context and meaning of words user is using to find result. It not only focuses on the keywords used. The major difference between the traditional and semantic search engine is that the traditional search engine retrieves the result based on the keywords while the semantic search engine retrieves results on the bases of the context and intent of the user query. Search Engine have many challenges so far including: scalability, content availability, evolution, visualization, ontology availability, Multilinguality [viii].

Purpose of this research work is to improve search results using semantics. To get variety in the result instead of getting site related to keyword. There are many semantic search engines working. They provide result on the basis of synonym. Semantic search engines are present e.g. Duck Duck Go, Hakia, Kngin etc. Search Engines are getting smarter and smarter. [ix] It is simply clear that no search engine is perfect in semantic search even Google the most common used engine also do not implements the semantic search exhaustively. Google uses semantic technology somehow but is not yet completely semantic search engine [x]. Main idea in the all of approaches is that search engine must understand the meaning of the query. So, it can answer on the basis of semantics (meaning) [xi]. So, the basic difference between these technologies from the technical point of view is relevancy algorithm. Relevancy algorithm [xii] is basically about ranking which is related to which site or document is best for the typed query which one is on the top priority. The purpose of our research is to propose the algorithm that is optimized one and better than the others.

III. PROCESSES AND ARCHITECTURE

Processes generally used in search engines are following:

A. Processes: Search engine components support following two process/Functions.

B. The Indexing Process: It is process if building the data structures that enable searching. It further comprises of the following three tasks [xiii].

1) Text Acquisition: It identifies the relevant searched document and stores it for indexing and stores the Meta data of the document as well.

2) Text Transformation: It converts the document into index terms or features. Index term is basically a part of document which is used in further searching that's why it is stored in the index and feature is the part of text document which represent the content of document

3) Index Creation: It takes indexed terms which are created by text transformations and create the data structures for fast searching. Gathering and recording statistical information regarding words, features, and documents and each one is weighted using statistical information regarding topic. And then inversion takes place in which conversion of the document-term information into term-document. That will occur for indexing and at last indexes are distributed across multiple computers and sites.

C. The Query Process: The data structures which are built in indexing process are used to produce a ranked list of documents for a user's query. It is also further comprising of the following three tasks [x].

1) User Interaction: It supports the creation (in which Interface is provided and parser for query language. Mostly web queries are simple mostly application use form as input) and Transformation of user query (which includes spell checking, query suggestion and text transformation techniques) and at last displays the ranked documents in the result.

2) Ranking: Each Search engine uses its own ranking algorithm. Each document is scored using ranking algorithm Basic scoring equation is Σ qi di. Where q and d stand for query and document terms weights for term.

3) Evaluation: It deals with monitoring and measuring the effectiveness and efficiency by logging the user queries and interactions. Raking analysis and Performance analysis is done here.

D. Architecture of semantic search engine: Semantic search engine architecture is same as that of the traditional search engine the major difference in their relevancy algorithm and ranking algorithm and as well in user interaction which includes the query transformation. So the two major processes indexing and querying is different for both search engines.

1) Ontology development

Ontology is defined as a formal and explicit condition of a conceptualization [xiv]. Basically, there is no definition which is accepted universally. There are two main objectives of the ontology 1. Knowledge Sharing. 2. Machine Understanding. Tools and languages are used for creating the ontology. One of the recommended languages is OWL (Web ontology Language) Ontology Crawler.

Ontology crawler finds new ontology by crawling through the web and places them in the database. Crawler will not dump all the found ontologies into DB. Ontology translator will first translate them and then ontology mapper will map them into DB. Here basically the quality of the search engine depends on the ―Knowledge which is stored in the database.

2) Ontology Annotator

Annotator is a web app for CMS or tagging documents to manage metadata like tags and annotation. Once the ontologies are created, now its aim is to understand the web pages with this metadata. It is capable to read from plain text files or databases [xv].

3) Web Crawler

Web crawler finds the interpreted web pages. Just like the TSE which looks for the keyword and create word index. But SSE looks for the semantics and main idea in the marked web pages and builds knowledge base [xv].

4) Query Builder

It is a powerful tool to construct search queries because users don't have the knowledge of ontologies perform semantic search. It loads the ontologies from the database. All the accessible ontologies is provided to select the precise context by user [xv].

5) Query Convertor

Here the query is searched in three ways: by concepts, by links, and by searching for keyword with synonyms too.

6) Semantic Content Retriever

Results are extracted from the earlier obtained semantically indexed web content. Then the retrieved content checked by keywords as well as the concepts, and the result that is presented to user is the intersection of the documents containing both the keywords and the concepts.

7) Semantic Ranker

The ranking of results must be done using ranking algorithm so that most relevant result is on the top.

8) Relevancy Algorithm

What we usually mean by relevancy/relevance algorithm is algorithm that which we use to check how similar the contents of a full-text field are to a full-text queries string [xvi].

IV. ALGORITHM

Semantic relations are analyzed by the synonym/meaning (not labels, only concepts). Semantic search focuses on concepts instead of terms [xvii]. We have presented an algorithm for semantic matching, and then we will discuss its implementation.

Search engine use the search interface to take input from the user. User types query in search bar. Web crawler is responsible to fetch the appropriate pages from the database against the user query. Meta data, a few sentences and title is mostly found in fetched pages. Then the process of indexing will occur, complete process of indexing is already will explained later in algorithm. Those retrieved pages are indexed according to the ranking algorithm.

Then the result is shown to the user. User clicks to open the link. User activities are also logged for the evaluation purpose and evaluation is done to improve the ranking algorithm. The key intuition of Semantic Matching is to find semantic relations. This all is done using a four steps approach,

Step1: Parsing: convert query to lower case.

Step2: Searching and mapping: Find all semantics of inserted word. Map all words to related semantics and find all related results.

Step3: Rendering: Fetch all the related results of web and semantic search from custom repository.

Step4: Rank the results according to their relevancy and show to user.

Given below is the proposed algorithm, variables used in algorithm have following meanings:

q: query
w: word
s: semantic,
S[ ]: Contains words and all related semantics (Words and semantics association)
R+: strong relationship of equivalence (=) between word and semantics

```
START

Step-1 (Searching)
1- Search query
2- q = Receive search query

Step-2 (Parsing)
3- pass q to server
4- w = parse "q" to lower case

Step-3 (lookup and Mapping)
5- lookup for "w" in Database
6- If (w)
        i.    S[ ]= Get All Semantics (w)
        ii.   Relation R+ (strong) is being made if there is
              equivalence (=) between words. Otherwise R- is
              made which is idk (I don't know)
        iii.  WS[ ]=Make Association between Words And Its
              Semantics(w, s)
        iv.   C[ ]= Get Content For Words And Association (w, s)
7- Else
        i.    Show error message "no such word found"

Step-4 (Ranking)
8- WC[ ] = Extract for Web from Content (C)
9- SC[ ] = Extract for Semantic from Content (C)
10- For(WC[])
        i.    Ranked_WC[ ]= (wci , wci+1 , wci+3 , .... , wcn-1,
              wcn)
11- For(SC[ ])
        i.    Ranked_SC[ ]= (wci (scj(sck( scl( scm (scn ))))))
        ii.   (wci+1 (scj+1 (sck+1 ( scl+1 ( scm+1 (scn+1 ))))))
                        .
                        .
                        .
                        .
        iii.  iii. (wcz(scz (scz( scz( scz (scz ))))))
Where i, j, k, l, m, n represent indexes of word and related
semantics and z represents end index of word and its semantics

Step-5(Rendering)
12- Render Ranked_WC[ ] show output
13- Rnder Ranked_SC[ ] show output

END
```

R-: weak relationship of equivalence between word and semantics
WS[ ]: Association between Words and Its Semantics
C[ ]: Contain content for word association
WC: Contain web result

A. Step-1 Searching

Searching is first step where query is received from using input screen. Query is being stored in variable "q".

1- Search query
2- q = Receive search query

This q could be in any format, lowercase, uppercase or both. Following could be the query:

q
Accommodation
Accommodation
ACCOMODATION

Fig. 1. Query to search

B. Step-2 Parsing

Basic aim of step 2 is to convert query of user into a consistent format so it is easy for system to understand. When query is passed to the search engine through user interface, query "q" can be in any format it can be lowercase, uppercase or mixture of both. But for matching from database purpose it is required that query is in a proper format same as the format in database. For this purpose, parsing is required. Parsing can be done for both lower and upper case. In this search engine query will be converted into lowercase.

1- pass q to server
2- w = parse "q" to lower case

After converting "q" in lower case that will be saved in a new variable "w" stands for word. Now basically "w" is main word which will be further used for searching purpose.

Following shows how "q" is parsed into lowercase "w"

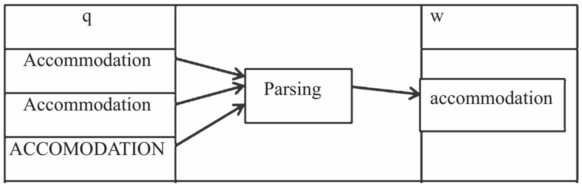
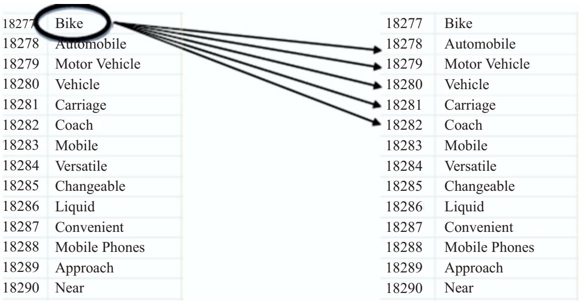


Fig. 2. Query Parsing in lowercase letters

C. Step-3 Lookup and Mapping

After query is being parsed successfully now next work is being started. System will look in databases to find related semantics of "w". *Lookup for "w" in Database.* For searching purpose, it will go through all indexes words table and all those which are semantics of w will be placed in.



18277	Bike	18277	Bike
18278	Automobile	18278	Automobile
18279	Motor Vehicle	18279	Motor Vehicle
18280	Vehicle	18280	Vehicle
18281	Carriage	18281	Carriage
18282	Coach	18282	Coach
18283	Mobile	18283	Mobile
18284	Versatile	18284	Versatile
18285	Changeable	18285	Changeable
18286	Liquid	18286	Liquid
18287	Convenient	18287	Convenient
18288	Mobile Phones	18288	Mobile Phones
18289	Approach	18289	Approach
18290	Near	18290	Near

Fig. 3. Words and semantic mapping

First, it maps words to its real-life semantics. This search includes logical operator equivalence (=) for example W = S, if equivalence relation connects them Translation of rel (w, s): w=s into propositional logic It will be w ↔ s, which means that they are exactly equal to each other. Since w=s holds if and only if w is subset of s and s is subset of w holds.

Now by considering equivalence relation assign relations: *R+*: strong relationship of equivalence (=) between "w" and semantics

	18278	18279	18280	18281	18282	18284	18285	18286	18287	18288	18290
18277	R+	R+	R+	R+	R+	R+	R-	R-	R-	R-	R-
18283	R-	R-	R-	R-	R-	R-	R+	R+	R+	R+	R-
18289	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R+

Fig. 4. Query to search

Take primary key of word table as foreign key in association table and make association between words and related semantics:

```
1-  If (w)
      i.   S[ ]= Get All Semantics (w)
      ii.  Relation R+ (strong) is being made if there is equivalence
           (=) between words. Otherwise R- is which is idk (I don't know)
iii.  WS[]=MakeAssociationbetweenWordsAndItsSemantics(w, s)
  iv.   C[ ]= Get Content For Words And Association (w, s)
2-  Else
      i.   Show error message "no such word found"
```

Content will have following things against each entry:

cache Id
display Link
Formatted URL
html Formatted URL
html Snippet
html Title
kind
link
snippet
title

All of this information will be stored in C [ ], against w and its related semantics which are mapped. This result will be further used for displaying result purpose.

D. Step-4 Ranking

Ranking is the process of sequencing collected results according to some criteria. In this algorithm results are ranked based on their relevancy with word. Basically, in this prototype there will be two results. One result is simple results (in our case Google results) and second results are for semantic search which is our own result.

WC[ ] array is used to store results which are web based.

1- WC[ ] = Extract for Web from Content (C)
2- SC[ ]= Extract for Semantic from Content (C)

Results for WC[ ] are ranked by following order:

$(Wci_{,} wci_{+1}, wc_{i+3,} \ldots, wc_{n-1,} wc_{n})$

These results have no such ranking; they are simply ranked according to how they placed in repository. Staring from i index, here i=0, these results are purely for "w" which is entered by user. Results are shown up to nth index; here n is last index of word content. This ranked array is now stored in *Ranked_WC[ ]* which is further use for rendering purpose.

For second results which are semantic result they need to rank carefully. Basic ranking starts from here. These results require to be ranked according to their relevancy. For this purpose, following ranking formula are used:

1- $wci(sc_j\,(sc_k(\,sc_l(\,sc_m\,(sc_n))))))$

2- $(wc_i{+}1\,(sc_j{+}1\,(sc_k{+}1\,(\,sc_l{+}1\,(\,sc_m{+}1\,(sc_n{+}1\,))))))$

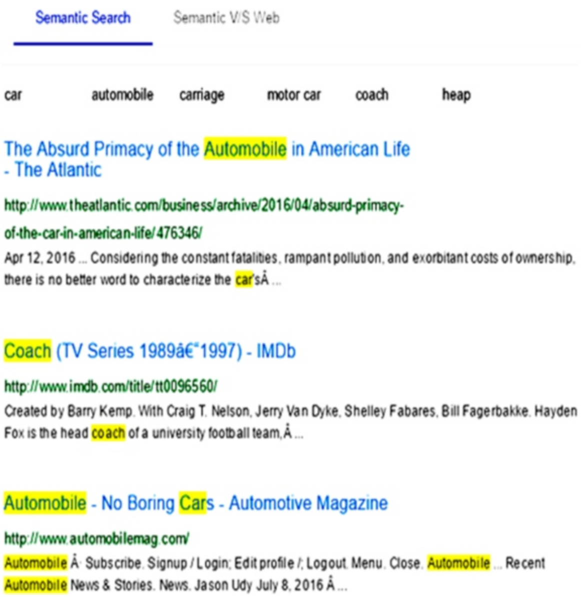
3- $(wc_z(sc_z\,(sc_z(\,sc_z(\,sc_z(sc_z\,))))))$

Where i, j, k, l, m, n represent indexes of word and related semantics content. Where z represents end index of word and its semantics. Here in first iteration first content against word and its each semantic will be stored (*i, j, k , l , m , n)*, after that next results (*i+1, j+1, k+1 ,l+1 , m+1 , n+1*) will be ranked and place. This ranking will allow best mixture and variety of word and its related semantics. At the end zth index of all semantics and word will be placed, which is last index of word and each semantic.

Word	$wc_{i,}wc_{i+1,}wc_{i+2,\ldots\ldots,}wc_{z-1,}wc_z$
Semantic1	$sc_{j,}sc_{j+1,}sc_{j+2,\ldots\ldots,}sc_{z-1,}sc_z$
Semantic2	$sc_{k,}sc_{k+1,}sc_{k+2,\ldots\ldots,}sc_{z-1,}sc_z$
Semantic3	$sc_{l,}sc_{i+1,}sc_{l+2,\ldots\ldots,}sc_{z-1,}sc_z$
Semantic4	$sc_{m,}sc_{m+1,}sc_{m+2,\ldots\ldots,}sc_{z-1,}sc_z$
Semantic5	$sc_{m,}sc_{m+1,}sc_{m+2,\ldots\ldots,}sc_{z-1,}sc_z$

E. Step-5 Rendering

Rendering is the process of showing results/output on screen. After ranking results now, they are displayed. As already mention there are two basic results one is web and other is semantic result. *Ranked_WC[ ]* is rendered and result simple from Google is being displayed. Then results stored in *Ranked_SC[ ]* is rendered. Now semantic based result is shown. There is also a third screen which will use both arrays *Ranked_WC[ ]* and *Ranked_SC [ ],* screen will be split in two section one will make use of *Ranked_WC [ ]* and second will make use of *Ranked_SC [ ],* this will show basic difference between web results and our prototype results.



The Absurd Primacy of the Automobile in American Life - The Atlantic

http://www.theatlantic.com/business/archive/2016/04/absurd-primacy-of-the-car-in-american-life/476346/

Apr 12, 2016 ... Considering the constant fatalities, rampant pollution, and exorbitant costs of ownership, there is no better word to characterize the car'sÂ ...

Coach (TV Series 1989â€"1997) - IMDb

http://www.imdb.com/title/tt0096560/

Created by Barry Kemp. With Craig T. Nelson, Jerry Van Dyke, Shelley Fabares, Bill Fagerbakke. Hayden Fox is the head coach of a university football team,Â ...

Automobile - No Boring Cars - Automotive Magazine

http://www.automobilemag.com/

Automobile Â· Subscribe. Signup / Login; Edit profile /; Logout. Menu. Close. Automobile ... Recent Automobile News & Stories. News. Jason Udy July 8, 2016 Â ...

Fig. 6. Semantic Search results

V. Experimental Work

In our research work we have made a tool named as SSE 1.0. That works on limited dataset of words and their related semantics it gives result of each semantic against query of word, and it provide relevant results. We have done an experiment and involved famous search engines yahoo, Google and Bing. Compared those collected results based on meanings/synonyms of words. This comparison is done with our search engine which is built in .NET.

When we enter word "car" our search engine, it will not just consider word "car" itself. But it will also consider its meanings present in ontology. So, in this case it will look at all its possible related semantics and will show result accordingly.

For every first ten results which are ranked as top-rated links, there is 20% semantics included in both Google and yahoo, but in Bing this ratio is 60% whereas in SSE 1.0 variety in results is up to 80%. For next results from 11 to 30 there is 31.57% semantic support in Google, whereas in yahoo this ratio is 36.84%, in Bing ratio for 11-30 results is 42.11%, for SSE 1.0 this ratio is 47.36. For next limit from 31-50 Google: 36.84 yahoos: 47.36 Bing: 42.11 SSE 1.0 63.16. And for the next result (up to 50), following is the ratio of semantic base result.

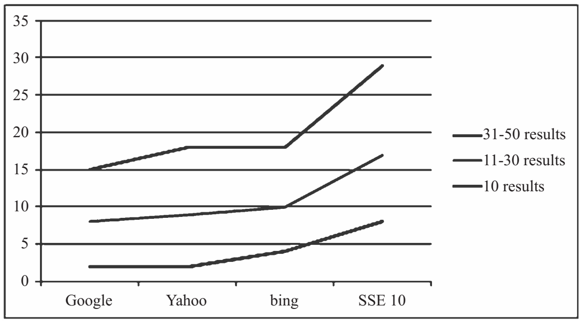


Fig. 7. Comparison between Google, yahoo, Bing and SSE 1.0

On x-axis search engines are listed, on y-axis number of total results are present. These lines are showing number of semantics included by each search engine. Staring from first 10 results, then next 11 to 30 results and then next 31 to 50 results.

Relevancy of result is next issue; we have tried to resolve this issue up to some extent. This is future work for this research as well to increase relevancy of results and to include phrases as well.

VI. Conclusion and Future Work

We have presented a semantic matching algorithm with its experiment. We have compared our algorithm with three famous search engines. Results proved strength of our algorithm, but relevancy is still a big issue in proposed algorithm. Future work includes making such a system which will focus on concern of searcher. Hopefully, that would bring more relevant results. This algorithm works by considering all semantics, it does not focus on the relation between word and semantic is strong or weak. That is why the next work should to develop an algorithm which is capable enough to determine strength of relationship between semantics. The results of this algorithm can be improved by determining the intersecting relation between word and its meanings. First work has already been done. This algorithm is implemented in limited set of data. So, next semantic search engine can be implemented in detail on large and real-world data.

Acknowledgment

We are thankful to Dr. Javaria Ikram on her valuable suggestions for improving the quality of this research article. She is a visiting faculty at International Student Center, School of Computer Sciences, Beijing Institute of Graphics Communication China.

References

[i] What is World Wide Web [Online].Available: http://www.webopedia.com/TERM/W/World_Wide_Web.html

[ii] S. Redondo (November 22, 2014). What is Semantic Search and Why Should I Care? (SEO 101) [Online]. Available: https://www.searchenginejournal.com/seo-101-semantic-search-care/119760/

[iii] G. Madhu 1 and A. Govardhan 2 Dr. T. V. Rajinikanth 3 (January 2011). Intelligent Semantic Web Search Engines: A Brief Survey. International journal of Web & Semantic Technology (IJ WesT) Vol.2, No.1

[iv] R. Guha, R. McCool, E.Miller, "Semantic Search" WWW 2003.*Retrieved July 13, 2012*.

[v] X. Wei, F. Peng, H. Tseng, Y. Lu, X. Wang, B. Dumoulin. “Search with Synonyms: Problems and Solutions”Coling 2010: Poster Volume, pages 1318–1326,Beijing, August 2010

[vi] J. M. Kassim and M. Rahmany, (5-7 August 2009). Introduction to Semantic Search Engine. Faculty of Technology and Information Science, University Kebangsaan Malaysia. 2009 International Conference on Electrical Engineering and Informatics, 5-7 August 2009, Selangor, Malaysia ©2009 IEEE IS-1

[vii] Q. M. Ilyas, Y. Z. Kai and M. A. Talib, (2004). A Conceptual Architecture for Semantic Search Engine. Department of Electronics and information Engineering Huazhong University of Science and Technology, Wuhan - 430074, P.R. China © IEEE

[viii] V. Richard Benjamins1, Jesús Contreras1, Oscar Corcho 2 and Asunción Gómez-Pérez2. Six Challenges for the Semantic Web.Intelligent Software Components, S. A.

[ix] Semantic Search [Online]. Available: https://www.techopedia.com/definition/23731/semantic-search

[x] P. Midwinter. Is Google a Semantic Search Engine? [Online] Final Year Project Semantic Search Engine Available: http://readwrite.com/2007/03/26/is_google_a_semantic_search_engine.

[xi] W. R. Agualimpia, Francisco J. L. Pellicer1, Pedro R. M-Medrano, J. N. Iso1, and F. Javier Z. Soria1 (2010).Exploring the Advances in Semantic Search Engines Computer Science and Systems Engineering Department, University of Zaragoza.

[xii] What is Relevance [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/guide/current/relevance-intro.html

[xiii] CS6200, Information Retrieval. David Smith College of Computer and Information Science Northeastern University

[xiv] T. R. Gruber, ―A Translation Approach to Portable Ontology Specifications‖. Knowledge Acquisition 1993, pp. I99 - 220.)

[xv] Y. Z. Kai, Q. M. Ilyas."A conceptual architecture for semantic search engine", 8th International

Multitopic Conference 2004 Proceedings of INMIC 2004.

[xvi] Onix text Retrieval Toolkit API Reference [Online]. Available: http://www.lextek.com/manuals/onix/ranking.html

[xvii] G. Zou, B. Zhang, Y. Gan, J. Zhang (2008), An Ontology-based Methodology for Semantic Expansion Search. Fifth International Conference on Fuzzy Systems and Knowledge Discovery © 2008 IEEE DOI 10.1109/FSKD.2008.475