IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 12, Issue 2 (May. - Jun. 2013), PP 93-100
www.iosrjournals.org

Efficient Approach for Knowledge Management Using Deep Web Information Retrieval System

Soniya Agrawal[1], Dharmesh dubey[2]
[1]*Department of Computer Science and Engineering, SDBCT, Indore, (M.P), India*
[2] *Assistant professor, Department of Information technology, SDBCT, Indore, (M.P), India*

Abstract: *Along with the development of network technology, web information are rapidly growing, and the way of information storage is gradually changed from the html to the database, thus web information can be divided into the surface web and deep web. Deep web is a concept corresponding to the surface web. It means from ordinary search engine that is difficult to discover the information content of a web page. The traditional crawler only crawl the content on the surface of a web, which makes the current traditional search engine, did not retrieve deep web data. Deep web compared with surface web has the advantage of large volume, high quality, theme single-minded, good structured. In view of several advantages, the establishment of deep web data integration system is becoming a research hotspot. The deep web query interface is the only entrance of the background database, so how to determine which web form is the query interface is important to the deep web information access. However, because the page proportion on the internet which contains querying interface is very small, using the traditional breadth-first strategy and keyword filtering method to crawl, it will download a lot of unrelated pages, spend a lot of resources, we need a way to efficiently find and collect the query interfaces through deep web crawling strategy. We proposed novel query planning approach, for executing different types of complex attribute through queries over multiple inter-dependent deep web data sources. increase accelerate query searching based on attribute selection, execution and propose optimization techniques, including query plan merging and grouping optimization.*

Keywords*: Novel query planning approach, Deep web, Semantic Deep Web, Ontologies, attribute.*

I. Introduction

Centralized search engines like Bing and Google use crawlers to download web content and build an inverted index so that users can quickly search within the content. Crawlers are given a set of seed pages and recursively download content by following the links on the downloaded pages. However, many pages on the web are hidden behind web forms and are inaccessible to crawlers. These pages are commonly referred to as the deep web [1]; in contrast, those pages accessible by following links are referred to as the surface web. The number of deep web pages is estimated to be up to two orders of magnitude larger than the surface web [1]. This content is often of high quality and highly relevant to the user's information need. In other words, it is of crucial importance to provide adequate deep web search functionality. For the remainder of this proposal, the inaccessibility of deep web pages to web crawlers will be referred to as the deep problem. Another problem related to web search is its immense size and continuous growth, which poses many challenges and hard requirements on the scalability of any web search solution [6].In 1999, it was estimated that no web search engine indexes more than 16% of the surface web, and that the web consisted of 800 million pages [10]. In 2005, a new estimate put this number at 11.5 billion pages [11], and in 2008, Google announced the discovery of one trillion unique URLs on the web at once. In 2012 unlimited URLs on the web at once the extremely large number of web pages and the continuous web growth will be referred to as the big problem. The following scenario illustrates some of the hurdles when searching for deep web content. Imagine that you are planning a short trip. You are gathering information about possible routes and trying to determine the preferred means of public transport: whether to go by bus, metro, train, or a taxi. In addition, you are comparing them to the costs and bents of traveling by car. To gather such information, you must submit a structured query to a complex web form (i.e. a form with multiple input fields). Chances are that you will have to re-type your complete query several times, once for each site about a particular means of transportation. This repeated process is tiresome. Furthermore, you must first find the right sites to query; otherwise you might not even find the (best) solution. It would be much easier if one could submit a single free-text query to a simple search interface, and search many complex forms at the same time (especially if one does not know about their existence).In [5], the attributes of the query interfaces were obtained manually. The automatic attribute extraction has been proven to be a difficult task [7]. In this paper, we present a Semantic Deep Web approach to identifying the Deep Web sources that are most relevant to the user search needs, utilizing ontology. Ontologies, together with agent technologies, are primary ingredients of what Berners-Lee et al. [8] have called the Semantic Web. Ontology provides semantic

support by using controlled terms for concepts in a domain. We combine the interfaces of the Deep Web and the ontology approach of the Semantic Web. Our approach makes the following Contributions:

- We concern ontology-based semantic similarity for estimate the consequence of a web query interface and its essential data Sources to address inaccurate and incomplete user search needs.
- The query interface attributes are repeatedly extracted from together the perspective of the customer (text labels) and the perspective of the (Web application) system (query form Attributes).
- The text labels that customer observe in the query interface are used in decisive the appropriate attributes, declining the partiality of Search results towards mainly accepted data sources.

II. Literature Survey

Liu Jing [1] in this research, they were proposed a deep web interface discovery method based on ordinal regression model. Their strategy puts Web Page Classifier, Link Info Extractor and Link Features Learner together. In the process of crawling, they classified web pages through Ordinal Regression Model based page classifier, estimate whether the web pages layer the same as the corresponding link layer, and feed the result back to Link Features Learner. Then, Link Features Learner extracts features of active link, and makes use of these features extract most promising links in each layer. In this approach uses adaptive learning so that it can adjust links info extractor in time. Page Classifier just chooses the most promising links in each layer to avoid crawling links which have no reference to query form.

Guangyue Xu in at al[2] proposed a narrative approach to classify deep webs, a significant step for significant integration of such sources. Aggravated by the characteristics of the deep web, propose a two phase framework by combining the topic model and the String Kernel method. Unlike traditional solutions, this system captures the query form's semantic structure and operates in an unsupervised manner. Moreover, their system achieves satisfying performance on the TEL-8 dataset from UIUC. Chelsea Hicks in at al[3]Compared to the surface Web, the deep Web contains vast additional information. In exacting, construction a generalized search engine that can index deep Web across all domains remains a difficult research problem. In this paper, we highlight these challenges and demonstrate via prototype implementation of a generalized deep Web discovery framework that can achieve high precision.

Fajar Ardian in at al[4] In this research, they have described a novel technique for efficiently maintaining common keys in a sequence of versions of archived continuous query results from deep Web sites. This is crucial for developing robust techniques for modeling evolutionary Web data, query processing, and tracking entities over time. Proposed an algorithm called COKE to discover common keys from the archived versions of structured query results represented as relations. It generates minimal common keys without computing the minimal key set of the new relation.

Hexiang Xu in a at al[8]numerous deep Web sources are structured by providing structured query interfaces and consequences. Classifying such structured sources into domains is one of the critical steps toward the integration of heterogeneous Web sources. Considering the semantic relations, such as synonym, hypemym/hyponym, meronym/ holonym and homonym etc., and the characteristic of attributes, in this research, they were present a deep Web model and machine learning based classifying model. Challenges deep web In regard to the deep problem, the challenges are given below and are divided into two different groups: query con-version (the first three) and user interfacing (the last two). Query description a formal syntax in which web administrators can identify the established language of the exacting resource. How can we maintain this intuitive and simple, while allowing enough freedom to specify almost any kind of query, and strict enough to allow easy parsing?

Query translation Due to possible spelling errors, ambiguity, or unknown words to the system, extracting the intended meaning of free-text queries is challenging. A query could be interpreted in different ways. How to devise a feasible approach that achieves reasonable performance (e.g. correctly interprets and translates over, say, 75% of the queries)? Interpretation ranking as stated in the previous point, a query could be interpreted in many ways. How to rank these interpretations in order to minimize the user's effort to scan through all interpretations, thus quickly finding the right one?

User ignorance How to bridge the gap between the expectations of the user and the capabilities of the system? Is it feasible to automatically suggest available search facets while typing (i.e. the aspects in which the search query can be narrowed further to obtain more specific results)? How to automatically choose suggestions such that the customer:

1) Is guided while formulating more distinctive queries, and
2) Can finish formulating the query faster?

System ignorance How to automatically expand the system's knowledge about valid queries? For example, given a query that contains unknown words, the system present several annotated interpretations. Then, if the same query is given many times and a particular interpretation is often selected, the system could learn a new

rule which includes the unknown word. Of these challenges, the main focus of this research will be on the query description, query translation, and ranking.

III. Attribute For Deep Web Data Sources

Before describing our approach to automatically extracting the Proper attributes from Web query interfaces of the Deep Web, we first clarify the meaning of the ambiguous term "attribute." In the General sense, an attribute of a Web query interface is any item of information that describes the Deep Web source. The more Specific meaning of "attribute" is derived from the HTML/XML Syntax. A tag of HTML consists of a mandatory name between angular brackets, which may be following by optional attribute/value pairs.

System perspective Attributes**:** extracted from within html tags. The simple <form> in figure 2 shows the html code of a number of common form elements. The <label>, <select>, <option> and <input> elements all contain system perspective attributes

Customer perspective attributes the results of analyzing the text of the web form, especially as it is associated with text entry areas.

IV. Our Approach: Automatic Attribute Extraction

Our approach to extracting the proper attributes for a, there is an overlapping area between the Customer perspective Attributes and System perspective Attributes. That is, the final attributes will be determined by comparing cpa and spa. Therefore, in order to automatically extract the attributes for each Web data source, we have developed a three-stage algorithm. Given a set of Web data sources, the spa are obtained from the inner identifiers of all the Web data sources. Secondly, the cpa are obtained from the free text within the query interface. Finally, the final attributes (FAs) of each Web data source are determined based on spa and cpa by utilizing an ontology. Note that, spa and cpa extraction, and FA determination are all achieved automatically.

Phase I: is the main algorithm of our Automatic Attribute Extraction (AAE). The details of each step will be described in Sections 4, 5 and 6. Note that, our algorithm does not treat PVAs and cpa in a symmetrical way. Only one spa set is matched against several cpa sets.

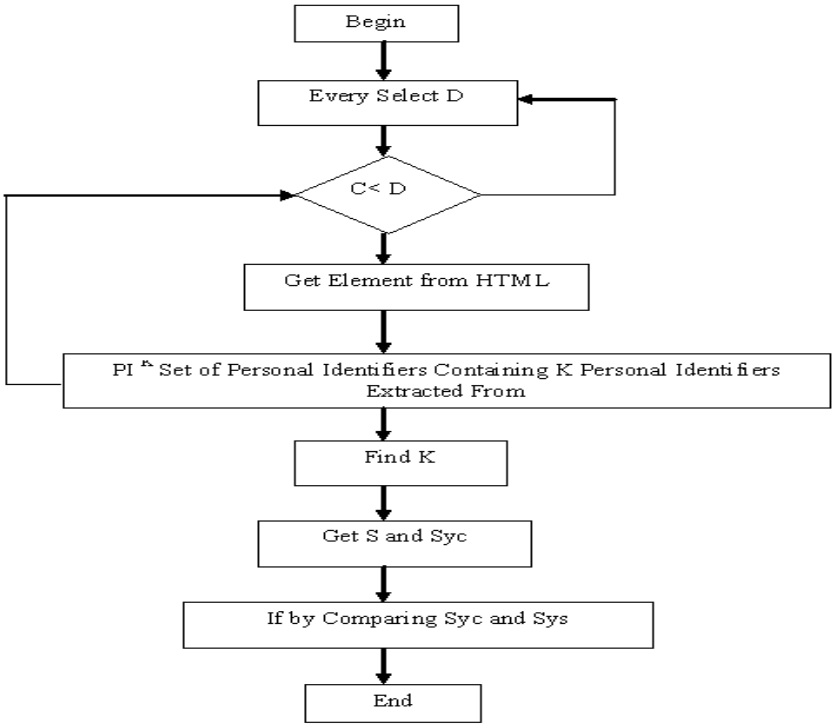
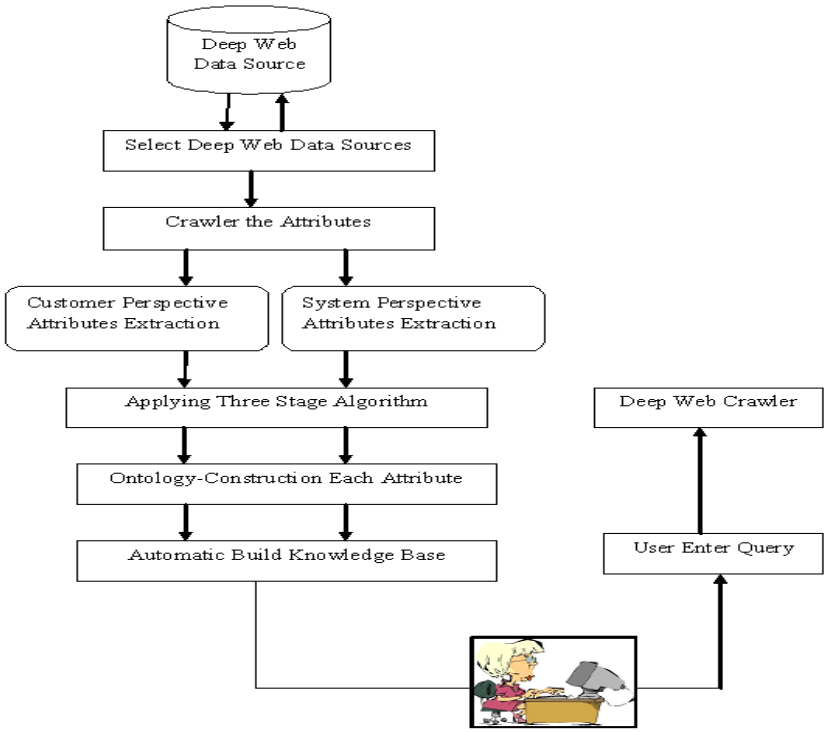


Figure 1: select deep web data sources

Let *Di* be a set of Web query interfaces for data sources containing HTML form elements {h_1, h_2 ... h_n}. Let *k Iii* is a set of inner identifiers containing *k* inner identifiers extracted from *Di*. Let *KW* be a set of keywords found between the begin tag <LABEL> and/or <SELECT> and the end tag </LABEL> and/or </SELECT> from all the Web data sources. Let spa and cpa_i be a set of developer and customer perspective attributes, respectively.



A set of synonyms for spa and a set of synonyms for cpa_i are acquired from WordNet. Note that, each D_i has its own cpa_i and $socpa_i$. The final attributes of D_i are determined from {spa ∧ sospa ∧cpa_i ∧ $socpa_i$}.

V. Pva Extraction

While inner identifiers can be easily obtained from HTML elements by a program, they cannot be directly used for further analysis since they are usually comprised of several words and symbols. Therefore, the inner identifiers have to be further separated into several independent words. The phase II shows steps for separating a set of inner identifiers of a Web data source *DSi* . The automatic derivation of KW is relatively simple by text extraction, thus we do not describe it in detail. Let *IICAi* be a set of inner identifier-based candidate attributes.

Phase II: extrication the position of private Identifiers function *eppif*(PI_i): *PICAi* ; instigate eliminate the duplicate private identifiers in set PI_i . for every private identifier in PI_i do start if the private identifier contain individual symbols (:,/,{,},@,[,],>,$,&,#,+,\,.,=,?,;,*,_,{,",},<, etc.) then divide the private identifier into numerous sub-strings; if each sub-string include private Capital letter(s) (i.e., camel case) then break every sub-string into numerous sub-strings; for every sub-string do begin for every key word of KW do begin if the key expression is situated in the sub-string then break every sub-string into numerous sub-strings with reverence to the key word; end end obtain the divided private identifier which is a string enclose numerous sub-strings end for every divided private identifier do begin count the quantity of sub-strings, *ss*, in the separated private identifier for index *i* = 1 to *ss* do begin extract a string which is self-possessed of *i*-word uninterrupted words from the divided private identifier, and add the string into a set of $pica_i$; end end Remove the duplicated strings in $pica_i$ return $pica_i$.

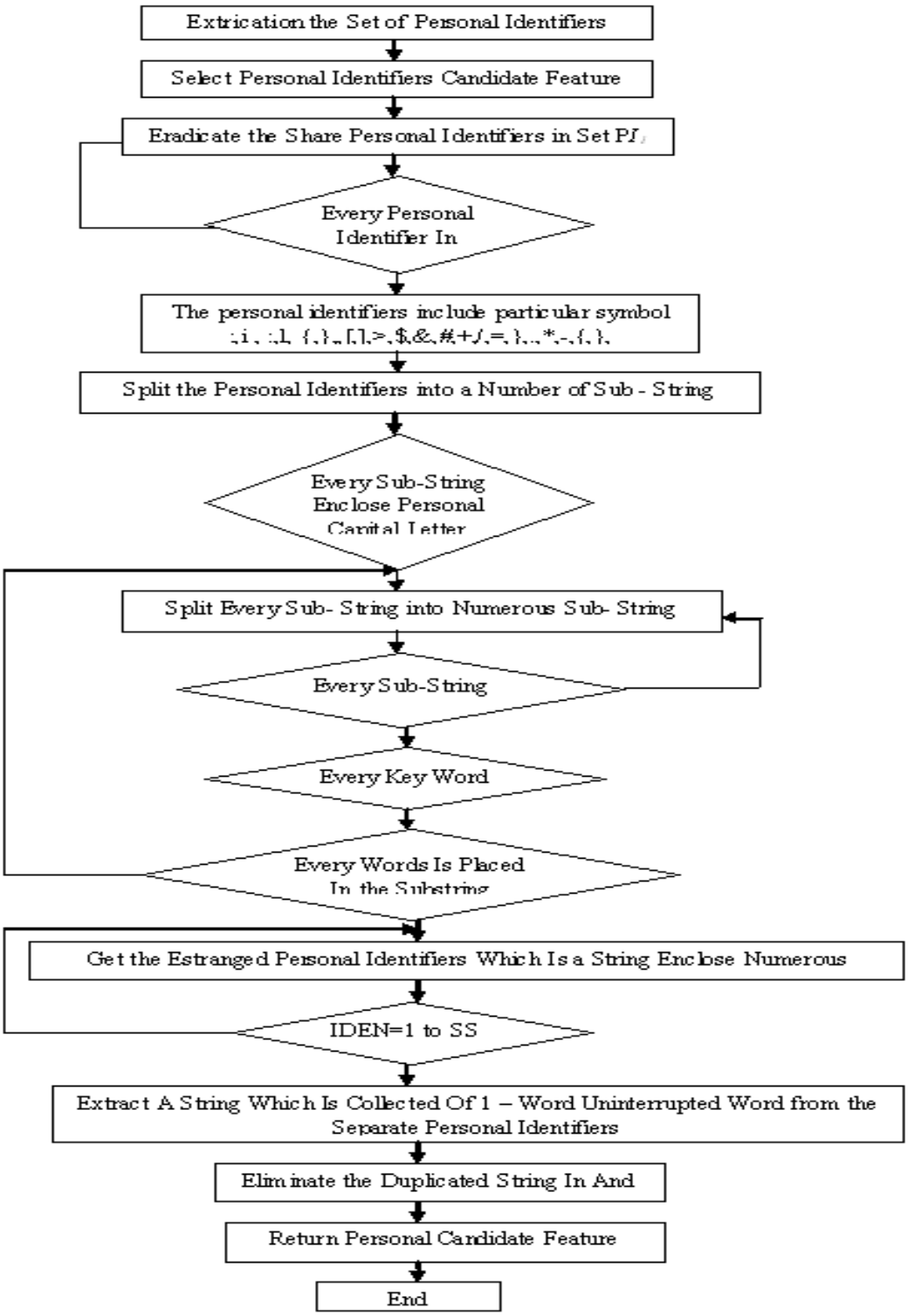


Figure 2: attribute for deep web data sources

Phase III: acquire spa function *ospa*(all of sets PI_i): spa; begin for each PI_i do begin acquire $pica_i$ by calling function *sspi*(PI_i) add all of $pica_i$ into a set of *spa* end for *pica* in *spa* do begin if *pica* appears one time in the *spa* then Remove the *pica* from the set *spa* if *pica* enclose numerous copies in *spa*, then keep one copy and eliminate the duplicate ones end return *spa* end. It necessitate that the free text between two HTML tags, which potentially embodies semantics, is added into the set *ftca*. The text between <OPTION> and </OPTION> is unobserved since it does not describe attributes but inanes.

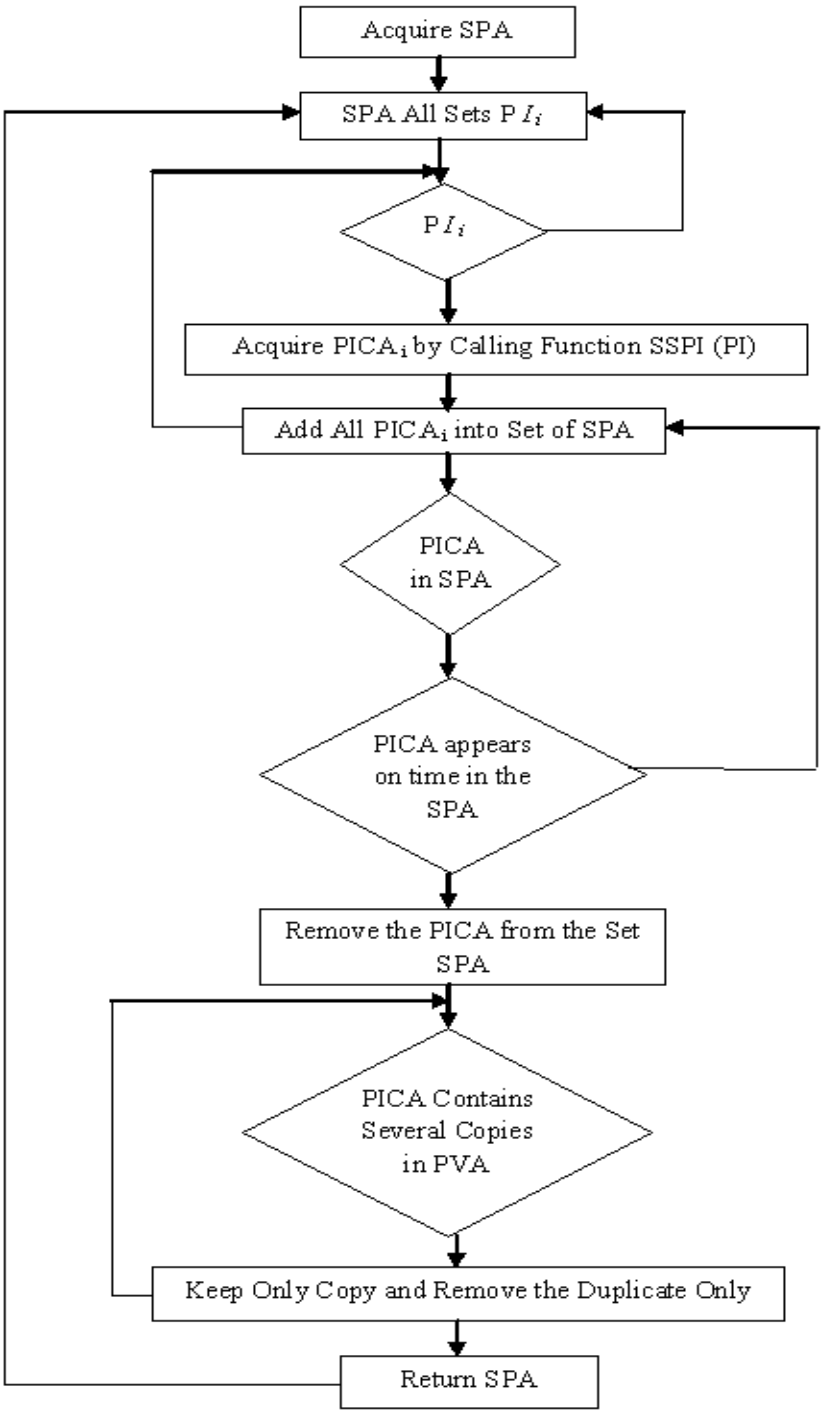


Figure 3: Acquire System perspective attributes

Phase IV: Obtaining UVA function ocpa (hi): cpai; begin eliminate every the text between <option> and </option> from h_i. For every h_i do begin acquire every free text between two HTML tags and add them as strings into set $ftca_i$. for every string in set $ftca_i$ do begin if a string include particular symbols then divide the string into sub-strings with deference to the symbols, to acquire numerous free text based candidate attributes add all ftca s into set cpa_i end eliminate the duplicate ftca from cpa_i end return cpa_i end history research behavior on the Semantic Web and on the Deep Web have sophisticated with modest communication. We recommend generating a Semantic Deep Web by calculation an ontology layer to the Deep Web. We communication that logic-oriented research on the Semantic Web has dealt with a Deep Semantic Web, which is different from our Semantic Deep Web.

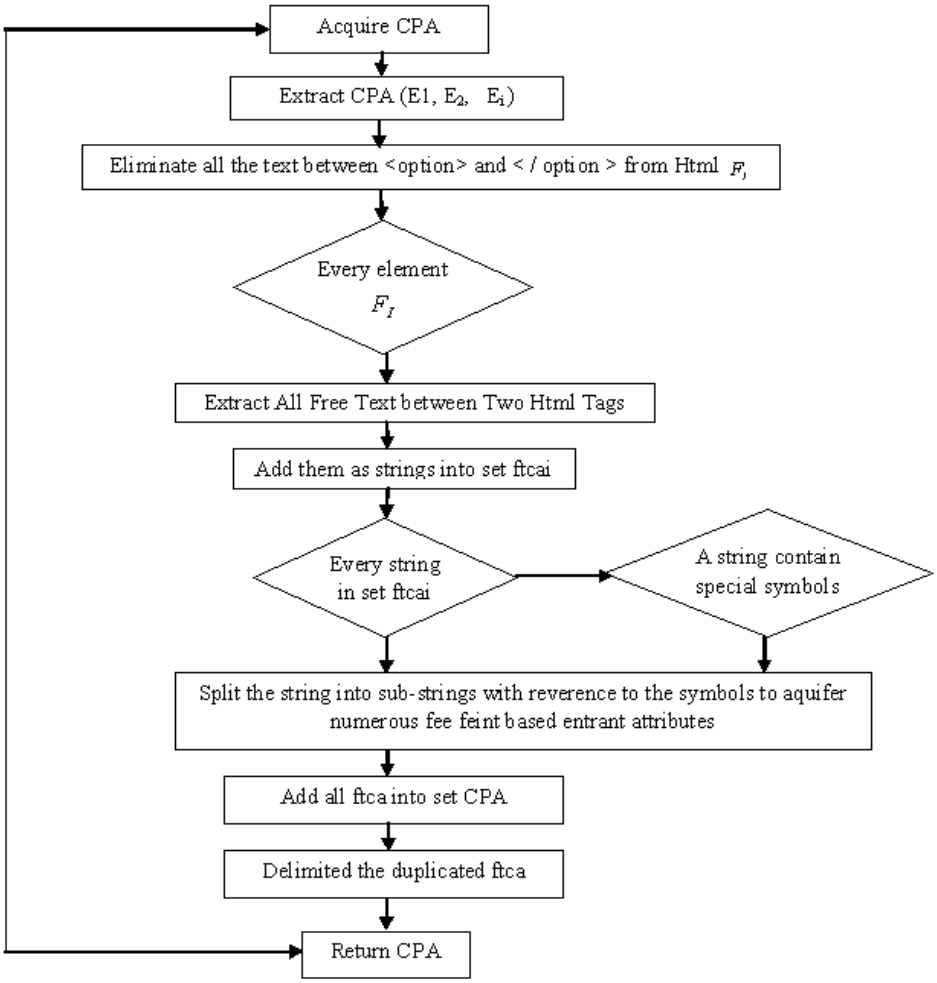


Figure 3: Acquire customer perspective attributes

VI. Ontology-Based Attribute Growth

Ontology is employed for giving out text from the inquiry interfaces of Deep Web sources in this paper, while the Deep Web is deliberate as a source for construction domain ontologies in [9, 10]. In calculation, we exercise ontology to competently filter words from the Deep Web data sources. The ontology adds a semantic layer to the Deep Web. In this paper, Word Net [11] is operating for finding matches between spa and cpa, based on synonyms. It is also used for Eliminating stop words to allow correct attribute retrieval. Among the Word Net categories, nouns, verbs, adjectives, and adverbs, we center on nouns base on the study that semantics are mostly carried by them [12]. Major search engines each were able to index part of the deep web. However, almost two thirds of the deep web was not indexed by any engine, indicating certain inherent barriers for crawling and indexing the deep web. Current approaches related to searching deep content include universal search of enterprise verticals domain specific mediators like and surfacing i.e. automatically filling in and submitting web forms, and indexing the resulting web pages. We will propose through our research Mediator frameworks that often set up by collaborating companies that allow access to their databases. Our propose frameworks do not crawl and index; instead, they broadcast every query to all databases. Our propose mediator often has a complex web form to ease the conversion of the query to the specific query format of each database. The last approach, surfacing, more general solution towards deep web search (since there is no collaboration between companies). However, there are deep web sites for which surfacing is not suitable, for example, sites that offer traveling schedules. Such indexed web pages would get outdated quickly. A prototype will be built that converts free-text queries into structured queries that are suited for some deep web site, i.e. which has a form with multiple input fields. The interface will look like that of a simple search engine (e.g. a text box and a search button), so that the user can freely enter any text to search for. Comparative user studies will then be performed to assess how users finish a pre-defined set of search tasks with a standard system and the newly built prototype. Among the measurements will be: task completion time, user satisfaction, and the use of query suggestions, result ranking, and the query translation effectiveness (i.e., the percentage of correctly translated queries).

VII. Conclusions and Future Work

In this research, a narrative routine attribute extraction approach was accessible which establish the attributes of Deep Web data sources by operate WorldNet. Two categories of attributes, system perspective attributes and customer perspective Attributes were defined in this paper to achieve our goal. The system perspective Attributes were acquire by separating the private identifiers extracted from attribute/value pairs of HTML tags, everywhere the attribute is name, or id according to pre-collected constructive key words. The customer perspective Attributes were extracted from English text labels. The final attributes of a Web data source were gritty by checking the extend beyond region among system and customer perspective Attribute sets.

ACKNOWLEDGMENT

We would like to express our gratitude to all those who gave us the possibility to complete this paper. We want to thank the Computer Science & Engineering Department of the SDBCT, Indore for giving me permission to commence this paper in the first instance, to do the necessary research work and to use departmental data. We are deeply indebted to our Master of Engineering supervisor Mr. Dharmesh dubey from the Information Technology Department SDBCT, Indore whose help, stimulating suggestions and encouragement

Reference

[1.] Liu Jing," A Regression Model-Based Approach to Accessing the Deep Web" 978-1-4244-7255-0/11/- IEEE -2011.
[2.] Guangyue Xu and Weimin Zheng, Haiping Wu and Yujiu Yang," Combining Topic Models and String Kernel for Deep Web Categorization" Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010).
[3.] Chelsea Hicks, Matthew Scheffer, Anne H.H. Ngu, Quan Z. Sheng," Discovery and Cataloging of Deep Web Sources" IEEE IRI 2012, August 8-10, 2012, Las Vegas, Nevada, USA.
[4.] Fajar Ardian, Sourav S Bhowmick," Efficient Maintenance of Common Keys in Archives of Continuous Query Results from Deep Websites" ICDE Conference 2011.
[5.] Yoo jung an, James geller, Yi-ta wu, Soon ae chun," semantic deep web: automatic attribute extraction from the deep web data sources" SAC'07, March 11-15, 2007, Seoul, Korea.
[6.] Ritu Khare Yuan An Il-Yeol Song," Understanding Deep Web Search Interfaces: A Survey" SIGMOD Record, March 2010 (Vol. 39, No. 1).
[7.] S. Lawrence and C. L. Giles. Accessibility of information on the web. Intelligence, 11(1):3239,-2000.
[8.] A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web, pages 902{903, New York, NY, USA, 2005. ACM.
[9.] Hexiang Xu,Chenghong Zhang, Xiulan Hao, Yunfa Hu," A Machine Learning Approach Classification of Deep Web Sources" Fourth International Conference on Fuzzy ystems and Knowledge Discovery (FSKD 2007). [1] M.P. Singh. Deep Web structure. IEEE Internet Computing, 6, 5 (Sep.-Oct. 2002), 4-5.
[10.] T.M. Ghanem and W.G. Aref. Databases deepen the Web. Computer, 37, 1 (Jan. 2004), 116-117.
[11.] UC Berkeley. Invisible or Deep Web: What it is, why it exists, How to find it, and Its inherent ambiguity. Available at http://www.lib.berkeley.edu/TeachingLib/Guides/Internet/ InvisibleWeb.html, July 2006.
[12.] M. K. Bergman, the Deep Web: Surfacing Hidden Value. Available at http://www.brightplanet.com/resources/details/deepweb.html, May 2006.
[13.] Yoo Jung An, James Geller, Yi-Ta Wu, Soon Ae Chun," Semantic Deep Web: Automatic Attribute Extraction fromthe Deep Web Data Sources" SAC'07, March 11-15, 2007, Seoul, Korea.