RESEARCH ARTICLE OPEN ACCESS

Smart Web Crawling System

Krutika R. Dipte[1], Karuna S. Mate[2], Prof. Swati Uge[3]
1(Computer Engineering, Smt. Radhikatai Pandav College of Engineering, Nagpur
2 (Computer Engineering, Smt. Radhikatai Pandav College of Engineering, Nagpur
3(Computer Engineering, Smt. Radhikatai Pandav College of Engineering, Nagpur

Abstract:
As deep web grows at a very fast pace, there has been increased interest in techniques that help efficiently locate deep-web interfaces. However, due to the large volume of web resources and the dynamic nature of deep web, achieving wide coverage and high efficiency is a challenging issue. In this paper implemented a three-stage framework, namely Smart Crawler, for efficient harvesting deep web interfaces. Web crawler performs site-based searching for center pages with the help of search engines, avoiding visiting a large number of pages. In this paper we have made a methodology on how web crawler works and what are the methodologies available in existing system from different researchers.

Keywords — **Smart Crawler, Deep web, web mining, feature selection, ranking.**

I. INTRODUCTION

In this paper, implemented an effective deep web harvesting framework, namely Smart Crawler, for achieving both wide coverage and high efficiency for a focused crawler. Based on the observation that deep websites usually contain a few searchable forms and most of them are within a depth of three our crawler is divided into three stages.

A novel three-stage framework to adder
ss the problem of searching for hidden-web resources. Our site locating technique employs a reverse searching technique (e.g., using Google's "link:" facility to get pages pointing to a given link) and incremental three-level site prioritizing technique for extracting relevant sites, achieving more data sources. During the in-site exploring stage, we design a link tree for balanced link prioritizing, eliminating bias toward web pages in popular directories.

Also propose an adaptive learning algorithm that performs online feature selection and uses these features to automatically construct link rankers. In the site locating stage, high relevant sites are prioritized and the crawling is focused on a topic using the contents of the root page of sites, achieving more accurate results. During the in site exploring stage, relevant links are prioritized for fast in-site searching.

II. ARCHITECTURE OF SMART CRAWLER

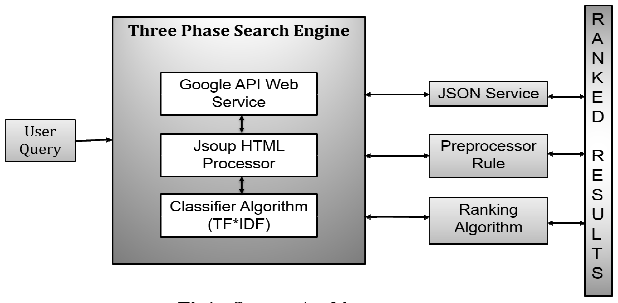


Fig1: System Architecture

To proficiently and adequately find profound web information sources, Crawler is outlined with a three-arrange engineering, website finding and in-webpage investigating, as appeared in above Fig.1, The principal webpage finding stage finds the most important website for a given theme, the second in-website investigating stage reveals accessible structures from the website and after that the third stage apply innocent base characterization positioned the outcome.

In particular, the site finding stage begins with a seed set of destinations in a site database. Seeds locales are competitor destinations given for Crawler to begin creeping, which starts by following URLs from picked seed locales to investigate different pages and different areas. At the point when the quantity of unvisited URLs in the database is not as much as an edge amid the slithering procedure, Crawler performs "invert seeking" of known profound sites for focus pages (very positioned pages that have numerous connects to different spaces) and sustains these pages back to the site database. Site Frontier gets landing page URLs from the site database, which is positioned by Site Ranker to organize profoundly important locales. Our exploratory outcomes on an arrangement of agent spaces demonstrate the dexterity and exactness of our proposed crawler system, which productively recovers profound web interfaces from huge scale locales and accomplishes higher collect rates than different crawlers.

A. *Algorithm:*

1. KNN:

The k-nearest neighbor's calculation (k-NN) is a non-parametric technique. Utilized for grouping and relapse. In the two cases, the info comprises of the k nearest preparing cases in the element space. The yield relies upon whether k-NN is utilized for arrangement or relapse:

• In k-NN grouping, the yield is class participation. A protest is arranged by a larger part vote of its neighbors, with the question being doled out to the class most normal among its k closest neighbors (k is a positive number, commonly little). In the event that k = 1, at that point the question is basically doled out to the class of that solitary closest neighbor.

• In k-NN relapse, the yield is the property estimation for the question. This esteem is the normal of the estimations of its k closest neighbors.

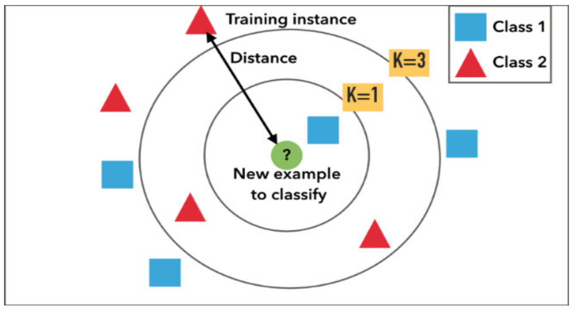


Fig2: KNN Algorithm

2. Naïve Bays:

It is an arrangement strategy in view of Bayes' Theorem with a presumption of autonomy among indicators. In basic terms, a Naive Bayes classifier expects that the nearness of a specific element in a class is irrelevant to the nearness of some other component. For instance, an organic product might be thought to be an apple in the event that it is red, round, and around 3 crawls in distance across. Regardless of whether these highlights rely upon each other or upon the presence of alternate highlights, these properties freely add to the likelihood that this organic product is an apple and that is the reason it is known as 'Credulous'. Gullible Bayes display is anything but difficult to assemble and especially valuable for huge informational collections. Alongside straightforwardness, Naive Bayes is known to beat even exceedingly modern arrangement strategies. Bayes hypothesis gives a method for ascertaining back likelihood P(c|x) from P(c), P(x) and P(x|c). Take a gander at the condition below:

$$\underbrace{P(c\,|\,x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x\,|\,c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

$$P(c\,|\,X) = P(x_1\,|\,c)\times P(x_2\,|\,c)\times\cdots\times P(x_n\,|\,c)\times P(c)$$

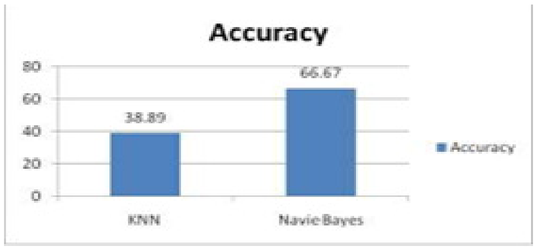


Fig3: Comparison of KNN & NB Algorithm

3. Implemented Algorithm:

Stage 1: Accept Query into Q
Stage 2: Read Results into Array List all utilizing Google API
Stage 3: Perform pre-handling and dismiss all labels and other media than content.
Stage 4: Calculate Word check of each page recovered.
Stage 5: Calculate Term Frequency = happen/add up to
Stage 6: Rerank pages in light of Term Frequency.

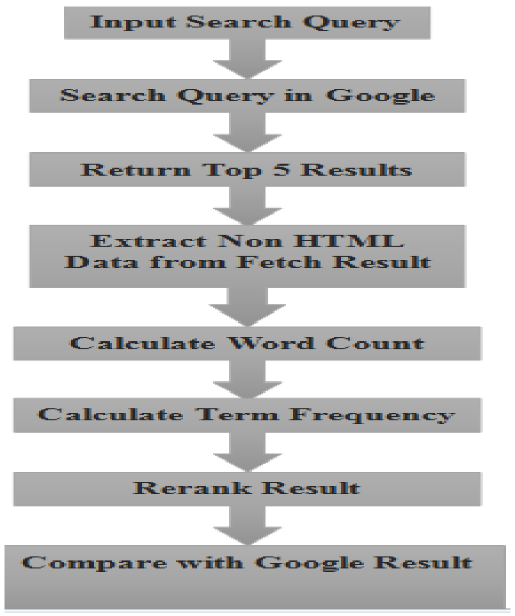


Fig4: Flow Chart

III. IMPLEMENTATION METHODOLOGY

A. ***First Phase: - Fetching Results from Google***

In first phase the proposed system fetches results from Google search engine with the help of Google developer API and JSON (Java Script Object Notation).

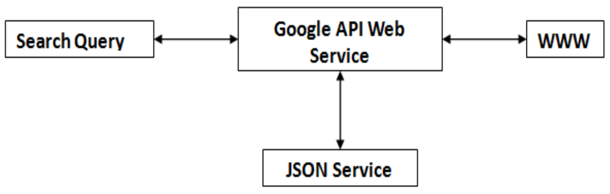


Fig5: First Phase

B. ***Second Phase: - Fetching the Word count from HTML Pages***

In second phase the proposed system opens the web pages internally in application with the help of Jsoup API and preprocesses it. Then it performs the word count of query in web pages.

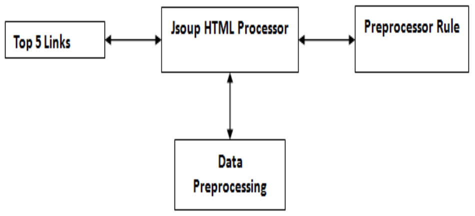


Fig6: Second Phase

C. ***Third Phase: - Frequency analysis***

In third phase the proposed system performs frequency analysis based on TF and IDF. It also uses a combination of TF*IDF for ranking web pages.

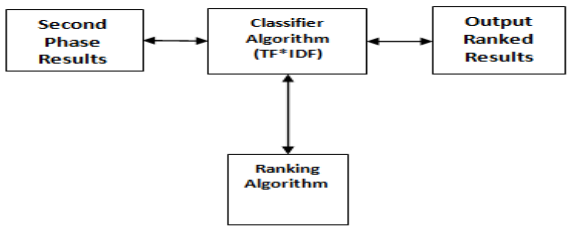


Fig7: Third Phase

IV. RESULTS AND DISCUSSION

A. ***System Simulation Snapshots:***

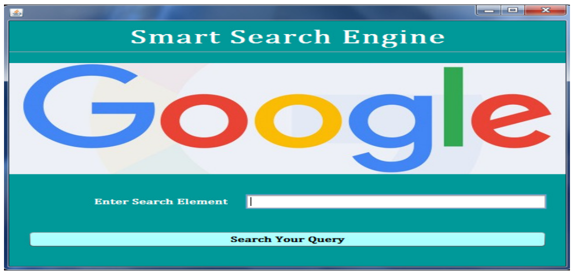


Fig8: GUI Design for Crawling System

This is the first form of design for proposed system. In this page user needs to enter the query that's gets searched on Google search engine with help of Google API framework. Using these first 5 results of Google search engine can be searched.

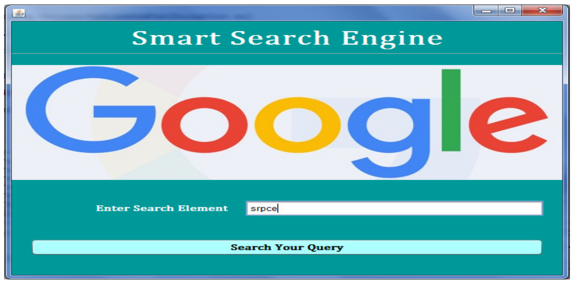
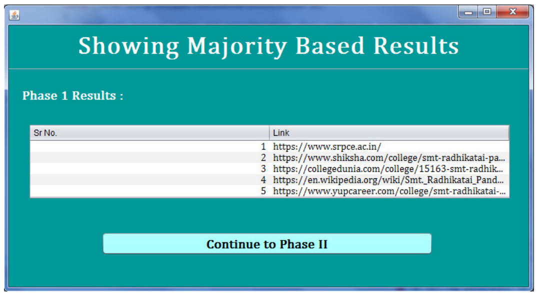


Fig9:Enter Search Element

This is an example for searching this keyword on Google. As the keyword is entered it is send to 1google search engine using query based JSON API. The next form shows the result of searching this keyword on proposed system.

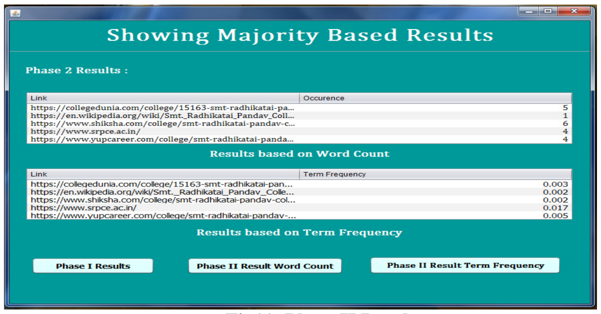


Phase 1 Results :

Sr No.	Link
1	https://www.srpce.ac.in/
2	https://www.shiksha.com/college/smt-radhikatai-pa...
3	https://collegedunia.com/college/15163-smt-radhik...
4	https://en.wikipedia.org/wiki/Smt._Radhikatai_Pand...
5	https://www.yupcareer.com/college/smt-radhikatai-...

Fig10: Showing Majority Based Result

These are the results that are fetched from Google Search Engine. Basic preprocessing is done as Google API return not only link but page details like meta tag titles URL Images and promotional links. All have been removed and original ranked results are showed.


Showing Majority Based Results
Phase I Results
Phase II Result Word Count
Phase II Result Term Frequency

Phase 2 Results :

Link	Occurence
https://collegedunia.com/college/15163-smt-radhikatai-pa...	5
https://en.wikipedia.org/wiki/Smt._Radhikatai_Pandav_Coll...	1
https://www.shiksha.com/college/smt-radhikatai-pandav-c...	6
https://www.srpce.ac.in/	4
https://www.yupcareer.com/college/smt-radhikatai-panda...	4

Results based on Word Count

Link	Term Frequency
https://collegedunia.com/college/15163-smt-radhikatai-pan...	0.003
https://en.wikipedia.org/wiki/Smt._Radhikatai_Pandav_Colle...	0.002
https://www.shiksha.com/college/smt-radhikatai-pandav-col...	0.002
https://www.srpce.ac.in/	0.017
https://www.yupcareer.com/college/smt-radhikatai-pandav-...	0.005

Results based on Term Frequency

Fig11: Phase II Result

As we move on to the next phase the proposed algorithm starts re-ranking the results based on k-NN and Naïve Bayes algorithm. Using these algorithms two different frequencies are calculated. First is word-count and second is term frequency that is calculated using JSOUP API.

B. *Graph Based Analysis:*

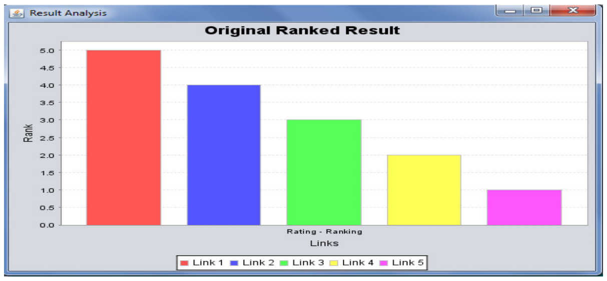


Fig12: Link Result

The above graph shows the ranking of links as provided by Google search engine. The bar-chart has been drawn using JFree chart library of java and is always changing based on results provided by Google.

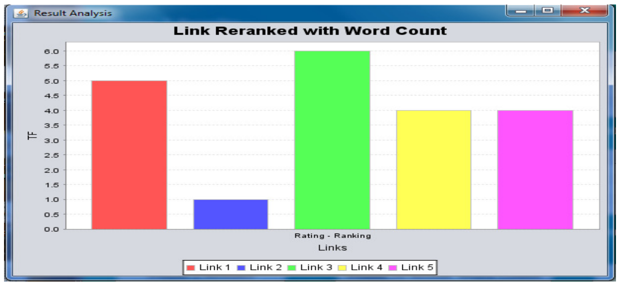


Fig13: Link Result With Word Count

The above graph shows the ranking of links based on wordcount based k-NN algorithm in which all the links are searched based on the keyword being searched in the first phase of algorithm. Using this we can verify the results of Google using k-NN algorithm.

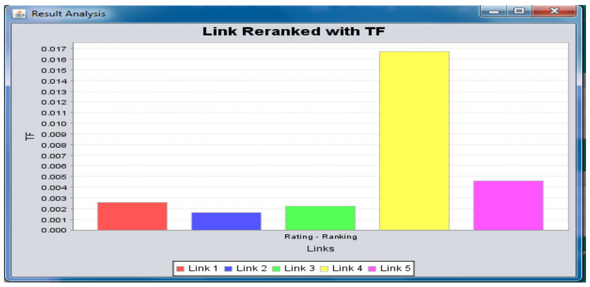


Fig14: Link Result with Term Frequency

The above graph shows the ranking of links based on TF based NB algorithm in which all the links are searched based on the keyword being searched in the first phase of algorithm. Using this we can verify the results of Google using NB algorithm.

V. CONCLUSION

In this paper implemented a three-phase structure, for proficient gathering profound web interfaces. In the main stage, web crawler performs website based scanning for focus pages with the assistance of web search tools. In the second stage the implemented framework opens the site pages inside in application with the assistance of Jsoup API and pre-processes it. In the third stage the implemented framework performs recurrence examination in view of TF and IDF. It likewise utilizes a blend of TF*IDF for positioning site pages. To take out predisposition on going by some profoundly significant connections in concealed web catalogues, in this implementation outline a connection tree information structure to accomplish more extensive scope for a site. Test comes about on an arrangement of delegate spaces demonstrate the spryness and precision of our implemented crawler structure, which effectively recovers profound web interfaces from substantial scale locales and accomplishes higher gather rates than different crawlers utilizing Naïve Bayes calculation.

REFERENCE:

[1]. *Feng Zhao, Jingyu Zhou, Chang Nie, Heqing Huang, Hai Jin "Smart Crawler: A Two-stage Crawler for Efficiently Harvesting Deep-Web Interfaces" in IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 9, NO. 4, JULY/AUGUST 2016.*

[2]. *Jianxiao Liu, Zonglin Tian, Panbiao Liu, Jiawei Jiang, "An Approach of Semantic Web Service Classification Based on Naive Bayes" in 2016 IEEE International Conference on Services Computing, SEPTEMBER 2016.*

[3]. *Bo Tang, Student Member, IEEE, Steven Kay, Fellow, IEEE, and Haibo He, Senior Member, IEEE "Toward Optimal Feature Selection in Naive Bayes for Text Categorization" in IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 9 Feb 2016.*

[4]. *Amruta Pandit , Prof. Manisha Naoghare, "Efficiently Harvesting Deep Web Interface with Reranking and Clustering", in International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 1, January 2016.*

[5]. *Anand Kumar , Rahul Kumar, Sachin Nigle, Minal Shahakar, "Review on Extracting the Web Data through Deep Web Interfaces, Mechanism", in International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 1, January 2016*

[6]. *Sayali D. Jadhav, H. P. Channe "Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques" in International Journal of Science and Research, Volume 5 Issue 1, January 2016.*

[7]. *Akshaya Kubba, "Web Crawlers for Semantic Web" in International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 5, May 2015.*

[8]. *Monika Bhide, M. A. Shaikh, Amruta Patil, Sunita Kerure,"Extracting the Web Data Through Deep Web Interfaces" in INCIEST-2015.*

[9]. *Y. He, D. Xin, V. Ganti, S. Rajaraman, and N. Shah, "Crawling deep web entity pages," in Proc. 6th ACM Int. Conf. Web Search Data Mining, 2013, pp. 355–364.*

[10]. *Raju Balakrishnan, Subbarao Kambhampati, "SourceRank: Relevance and Trust Assessment for Deep Web Sources Based on Inter-Source Agreement" in WWW 2011, March 28–April 1, 2011.*

[11]. *D. Shestakov, "Databases on the web: National web domain survey," in Proc. 15th Symp. Int. Database Eng. Appl., 2011, pp. 179–184. [12] D. Shestakov and T. Salakoski, "Host-ip clustering technique for deep web characterization," in Proc. 12th Int. Asia-Pacific Web Conf., 2010, pp. 378–380.*

[12]. S. Denis, "On building a search interface discovery system," in Proc. 2nd Int. Conf. Resource Discovery, 2010, pp. 81–93.