

Avquick Meta Search Engine For Quick Audio Visual Searching

N.R.Meddage, V.G.S. Pradeepika

Abstract: Video search engine optimization is a new trend in the internet. Increasing demand for video searching has persuaded business companies to spend more on video search engines as a marketing strategy. Therefore designing video search engines have become a lucrative and imaginary trend. Most of existing video search engines uses their own video repositories (databases) for storing video data to fulfill video requirements of Internet users and it costs more money because video files need more storage capacity and wide range of movies are needed to satisfy users and it is unaffordable for individuals and small business organizations. Metasearch engine is the ideal solution to develop video search engine with minimum effort and cost. Developing a metasearch is supported by a variety of web technologies and programming skills. This research project concerns about creating a server-based metasearch engine, "AVQUICK" to search multiple video search engines to extract top videos for given user queries and to re-rank results to give better video experience for users. AVQUICK is developed using VB.net programming language and it is not expected to use own databases. Different "Information Retrieval" (IR) techniques and web content mining techniques are used for searching engines, analyzing HTML document, extracting text and to re-rank collected results.

Index Terms: Query, Crawler, Ranker, Frontier, Information, Modifier, Filter, Web, Search, Merger, information Retrieval

1 INTRODUCTION

Video search engine optimization is a new trend in the internet. Increasing demand for video searching has persuaded business companies to spend more on video search engines as a marketing strategy. A metasearch engine is a search tool that sends user requests to several other search engines and/or databases and aggregates the results into a single list or displays them according to their source. Metasearch engines enable users to enter search criteria once and access several search engines simultaneously. Metasearch engines operate on the premise that the Web is too large for any one search engine to index it all and that more comprehensive search results can be obtained by combining the results from several search engines. This also may save the user from having to use multiple search engines separately. Therefore designing video search engines have become a lucrative and imaginary trend. Most of existing video search engines uses their own video repositories (databases) for storing video data to fulfill video requirements of Internet users and it costs more money because video files need more storage capacity and wide range of movies are needed to satisfy users and it is unaffordable for individuals and small business organizations. Metasearch engine is the ideal solution to develop video search engine with minimum effort and cost. Developing a metasearch is supported by a variety of web technologies and programming skills.

Problem Statement

Early days traditional search engines were used by on-line users and they were less user friendly because of operations associated such as Booleans, queries, proximity searches, text based and link analysis[1] [6] [7]. During last decade Internet search engines have been so popular that search engines were concerned as 'information-seeking vehicles and online advertising media'. A survey '10th WWW user survey' conducted by 'Georgia Tech University' reveals that 85 percent of web users use Internet search engines to fulfil their day to day searching requirements and it is reported that Search Engine Optimization (SEO) companies have achieved online advertising revenue of \$16.5 billion during the year 2005[2]. Search engines are different types and they use different searching mechanisms to search for different digital assets like videos, pictures, audio files, text documents, etc. Among the all different types of digital sources, there is a crucial place for online videos because the growing the demand. Article published in internet by 'Wayne Friedman' tells that the demand and usage of on-line videos have been increased unbelievably. According to the article, they have found from a survey that 37% of traditional TV consumers like shorter on-line videos than full length TV shows. Further it says 43% of online videos are uploaded by regular on-line video users. 32% of videos come from news stories, 31% from movie previews and 26% from comedy or bloopers. Article also says that 34% of TV users prefer to use Internet half of their TV watching time while they watch TV. The most important fact collected from the survey is "43% of all Internet users use to watch some on-line videos"[3]. This reveals that the popularity and usage over on-line videos grow rapidly. So it is important to design search engines to search for videos effectively and efficiently. Although there are numerous video search engines, most of them can't satisfy almost all the search queries of users due to different sizes of repositories associated. In order to overcome this problem meta search engines were introduced. Basically meta search engines send the user queries to different search engines to extract multiple results from them and then they re-rank the result using their own ranking techniques [4]. Therefore developing meta search engines can be recognized as low cost, more benefitted and less effort required aspect to provide better searching experience for internet users rather than creating brand new

- Meddage N.R. Sri Lanka Institute of Advance Technological Institute, 320 T.B. Jaya Mawatha Colombo 10, Sri Lanka Tel: 0712366113
E-mail: nadeeshika@gmail.com
- Pradeepika V.G.S. Sri Lanka Institute of Advance Technological Institute, 320 T.B. Jaya Mawatha Colombo 10, Sri Lanka Tel: 0718035218
E-mail: mmrisith@yahoo.com

search engines with own databases.

Requirement Analysis

When developing software products it is crucial to collect right set of requirements. In ordinary software development projects different methods like interviews, questionnaires and observations are carried out to identify user requirements. 'Mark Whidby' a director of a famous metasearch engine (dogpile.com) says that they have listened to their users and done research to collect requirements to enhance dogpilemetasearch engine [5]. But according to the nature of developed metasearch engine AVQUICK above three methods are impossible to gather requirements. Only possible method is studying the other existing metasearch engines

Development Methodology

The Architecture of new metasearch engine AVQUICK contains seven components (User Interface, Query Modifier, Crawler, Multi-Frontier, URL Filter, Out Links collection, Page ranker & Merger). The architecture below (Figure 1:) illustrates how the functionality of AVQUICK flows.

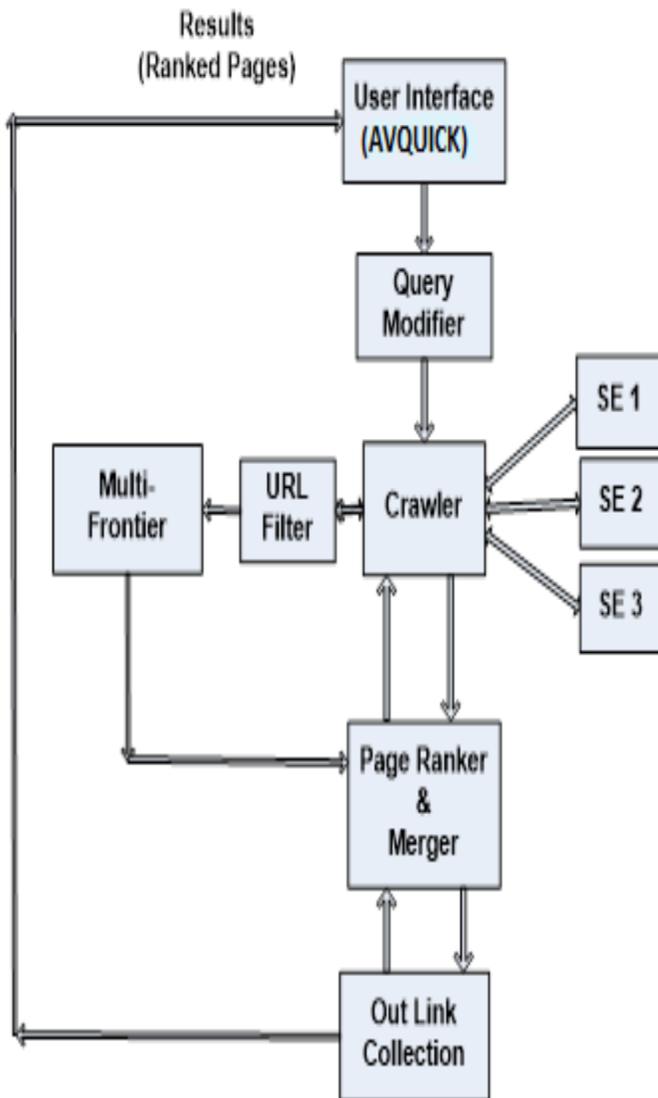


Figure 1 AVQUICK Metasearch Engine Architecture

User Interface can be simply accessed through web address of AVQUICK metasearch engine and it allows the users to enter their queries in natural language through keyboard. Then user interface passes all key words (query) to **Query Modifier** component. Different search engines require different queries. Query modifier creates right query for each search engine using key words of user query and they are passed to crawler component. **Crawler** sends relevant queries to suitable engines to collect results. Always the crawler collects Top 10 results from each engine by ignoring other results. HTML data of collected results are analyzed to extract valid URLs for videos (results) by omitting any unimportant URLs like advertisements and this done by **URL Filter**. The valid URLs (Results) extracted are added to **Multi-Frontier**. Frontier stores URLs from different search engines separate. **Page-Ranker & Merger** are two sub components. **Page Ranker** analyses the pages linked with result pages by using URL links on result pages. It searches pages in result using their URLs to collect URLs on those pages. Similar manner all new URLs (Out Links) collected are used to search new pages to collect URLs available in them. This cycle is followed repeatedly until it reaches expected number of total URLs. Then "Page Ranker" calculates the frequency of result page (URLs) appearing in collected URL collection [10]. The URLs with higher frequency are recognized as "Top ranked pages" (Videos). **Merger** combines the ranked results from different engines and passes to the user interface to display to user.[8], [9]. AVQUICK consists of TWO main functional modules (Searching/ Ranking) to satisfy video requirements of Internet users. This section illustrates how the searching and ranking is mapped using VB.net

Searching Process

AVQUICK searches for number of search engines linked and it uses number of reusable snippets (classes/ procedures/ user defined procedures/ user defined functions). Appendix1 illustrates how the AVQUICK performs searches on youtube and extracts the results step by step. Searching on other engines also take place in the same order. The user can enter his query with one or more key words and hit 'enter key' and press 'search button' to begin search. When user clicks on search button or hits enter, query is passed from text box to string type variable as below.

```
VDQuery = txtSearchBox.Text
```

The user query is used to create right queries to access search engines and to perform searches[11]. Below it demonstrates how the queries for different search engines are formed. The queries for search engines have constant and varying parts. Varying part is formed using the query (keywords), while the constant part(s) are added to get complete URL (query)[12].

Creating query for Youtube

In youtube the blanks between keywords are replaced with '+'. Inorder to achieve this challenge 'Replace function' (Regular Expressions) in VB.net is used.

Eg:

```
URLyoutube=
```

"http://www.youtube.com/results?search_query=" &
 Replace(VDQueryYoutube, " ", "+") &"&search_type=&aq=f"

Creating query for Myspace

Any blanks between keywords of query is replaced with "%20"

Eg:

URLmyspace = "http://www.myspace.com/search/videos?q=" &
 Replace(VDQueryMyspace, " ", "%20") &"&sl=t"

Then next search is carried out using query. "Webrequest" class searches the search engine and extracts the results (Results page). The result page is transferred to HTML data and HTML source is passed to a String variable for further analysis. Next extracted HTML data for result pages are analyzed to collect URLs for video pages. Searching for URLs starts from first character of HTML data and finishes at last character. There are important and unimportant URLs such as Advertisements, redundant URLs, etc and searching must extract only important URLs from HTML data. Before start searching for URLs some initiations have to be done. Unique phrases (characters) before the URL and after URL have to be mentioned. The "Starting" and "Ending" positions are different from engine to engine as bellow. Starting word and end word are used to track the place where URL exists and to chop the URL from starting position and ending position. If the URL needs any characters from Start word, the "Excess Length" is used to toggle text to chop URL from right starting point.

Youtube

TempText= tempdoc
 Startword= " href="/watch
 Endword= "
 ExcessLength=16

MySpace

TempText= tempdoc
 Startword= searchid=
 Endword= "
 ExcessLength=8

HTML data may contain more than one URL. So relevant URLs are extracted and some variables have to be initiated. Once the start of URL found, the HTML data has to be chopped and right side part has to be assigned as HTML data to proceed. This is done by Left (Left chop) function in VB.net. Extracted URL may contain pre-recognized unnecessary characters and they have to be removed. Bellow "Replace" function replaces all the unwanted characters (&feature=browch) with blank(""). Here removal is needed twice because URLs may come with two different unwanted words.

URLyoutube = Replace(URLyoutube, "&feature=browch", "")

URLyoutube = Replace(URLyoutube, "&feature=fvst", "")

Each time when a URL is gathered, it will be assigned to a String variable (pageID), to be added to a list of URLs (List) pageID = URLyoutube Once a URL is extracted, the remaining HTML data has to be searched to collect remaining URLs. So

HTML data (tempdocYoutube) is replaced with the HTML data starting from end of URL to end of HTML. This is done by bellow line. Right function chops HTML data and assigns right side of HTML to variable (tempdocYoutube).

tempdocYoutube = Right(tempdocYoutube, Len(tempdocYoutube) - IEnd)

As mentioned above searching is carried out to extract the "Title" for the Video extracted above. Beginning of Video title can be recognized as "" while ending is "<img title="" ". As above VB.net uses Left, Right functions for chopping HTML data. As identified during the researches and studies conducted about search engines plugged, same URL appears twice in Youtube result page[14]. So in order to avoid adding same URL twice, previously extracted URL and currently extracted URL are compared each other. "PageID" is the new URL while "lastURL" contains previously extracted URL.

If lastURL <> pageID Then

Once the URL and Title for a video is extracted, then next they are added to a list of object called "IstURLYoutube". Objects called "pgcontent" with properties "URL" & "Title" are created using "Pagecontent class" which has been already constructed. Extracted URL has to be transformed to a complete URL before adding to object and missing parts (<http://www.youtube.com>) are added as bellow. When a new object was created and added to "IstURLYoutube" list, current URL is assigned to "lastURL". So new URL is compared with new "lastURL" before adding it. Objects were added to "IstURLYoutube" during execution of search button. The data in "List" are supposed to use in "Rank Results" button, which is later executed by user. Above stored data is volatile and cleared from memory when user clicks on "Rank Results" button. So "IstURLYoutube list" is stored in a session variable called "youtube" Next the results extracted from search engines (Top 10 videos from each) are presented to the user as search result. A "for loop" which extracts all objects and their properties from "IstURLYoutube" list is used. For loop runs repeatedly from 0 to number of objects -1. In each time object accessed, video title and URL are added to a placeholder control. Placeholder displays all the added fields together on screen

Ranking Process

Once a user carries out a search, then he/she can rank the results. "Ranking algorithm used in AVQUICK uses all Outlinks from other pages to decide how much important a pages is. Outlinks from related pages and pages of related pages for a video is concerned as voting for a video page (URL). Pages with higher voting are displayed on top as top ranked pages. In this project extracting Outlinks from related pages, more related pages of related pages had done up to two levels because more rounds take more time and more processing power[13]. Bellow graph explains that which related pages are analyzed to collect outlinks.

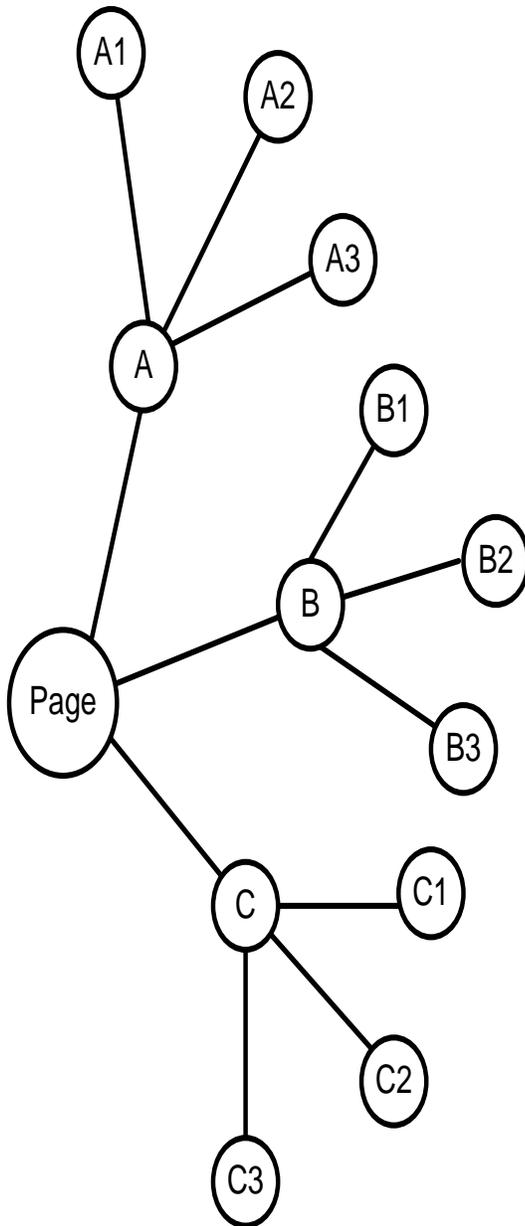


Figure 2 Graph with linked pages to analyse & collect Out Links

At level1 (A,B,C) Out Links on A,B,C pages are collected, then at level 2 (A1,A2,A3, B1,B2,B3, C1,C2,C3) out links of related pages of A,B,C (A1,A2,A3, B1,B2,B3, C1,C2,C3) are extracted. Then calculation is done to count how frequent the result videos URLs available in URL collection (out links). Videos (URLs) with higher frequency are identified as important pages and displayed on top of page as ranked Top ranked pages. The interface of this metasearch engine allows the user to see number of top ranked results. If the user selects "Top 4" then Ranking algorithm ranks all the results and displays Top two videos from each search engine. User can select option "All" to view ranked videos from top to bottom based on importance. This shows all the videos extracted for initial search. User can select ranking option (All/ Top 4) and click "Rank Results" or hit enter. user defined procedures (rankMyspace/ rankYoutube) are called to proceed ranking.

Call rankMyspace()

Call rankYoutube()

This step onwards it explains how "rankMyspace" procedure works. For loop is used to retrieve result URLs stored in session variable "session ("myspace")" during searching process.

Extracting all URL links on related pages of Video (Level 1/ Round 1)

ForMe.f = 0 To Session("myspace").Count - 1

Each URL retrieved is passed to a function (ExtractHTML) which extracts all the HTML data of the page related to given URL. The HTML source returned by function is stored in string variable "tempdocMyspace".

```
tempdocMyspace = ExtractHTML((Session("myspace")(Me.f)).url)
```

then "SearchMyspacePage" user defined procedure is called to search for URLs in extracted HTML source above. Four parameters tempdocMyspace, Starting point of URL (id=""muvVideoPlay-), End of URL ("") and Excess Length (16) are parsed.

```
Call SearchMyspacePage(tempdocMyspace, "id=""muvVideoPlay-", "", 16)
```

Above user defined procedure searches for URLs in HTML data source (tempdocMyspace) from beginning to end. Any URL found is added to IstTempURLmyspace with constant part as bellow. URLs extracted during while loop are recognized as TextValue.

```
IstTempURLmyspace.Add("http://vids.myspace.com/index.cfm?fuseaction=vids.individual&videoid=" & TextValue)
```

Once the 1st round (Level 1) of collecting outlinks finished, then all the outlinks URLs gather to "IstTempURLmyspace" are added to IstURLbaseMyspace.

```
IstURLbaseMYSAPCE.AddRange(IstTempURLmyspace)
```

Next 2nd round (Level 2) of collecting out link URLs starts. 2nd round follows the same steps and this time it needs more time because latest Out link URL collection is bigger than older one. Same steps followed in above for loop are applied to this section to collect and store all the URLs on pages collected above "myspace" is a session variable and it contains result URLs as objects with properties (URL, Title) stored. Ranking algorithm needs to compare each result URL with all the URLs in "Out link" collection to get voting or ranking weigh. First objects in session variable "session("myspace")" are parsed to blank list "myspaceURL" (list of objects) to ease functionality as bellow. myspaceURL.AddRange(Session("myspace")) 1st retrieves objects in "myspaceURL" one by one and their properties are assigned to variables for comparison purpose in next. 2nd retrieves URLs in Out Link URL collection ("IstURLbaseMYSAPCE") one by one and compares with URL of objects retrieves in above for loop to find any votes/ Rating weighs. If any similarity found

(if $z = y$),

then “tot” value for particular result URL is incremented by one

(tot = tot + 1)

At the end of 2nd round, the calculated vote/ ranking weigh (tot), a new object instant of summarizedpage class is created and property values are added to object.

```
pgcontent.URL = y
pgcontent.Title = w
pgcontent.tot = tot
```

Then next “tot” (ranking weigh) and “URL” are combined together and added to an array of string. This is done because weigh (tot) for URLs should be sorted into descending order. It is hard to sort a list of objects and this is done to overcome sorting problem. According to studies done, Weigh/ Ranking weigh (tot) can be any value between (0 – 99) and combined values (tot & URL) are stored as string and sorting faces a difficulty when there are “tot” values with single and two digits for different URLs.

Eg: below are three URLs with their votes and they are added to “list”.

```
10 urlA = 10 (tot) & urlA (result URL)
8 urlB = 8 (tot) & urlB (result URL)
50 urlC = 50 (tot) & urlC (result URL)
```

The expected ranking result show appear as follows after sorting into descending order.

1. 50 urlC
2. 10 urlA
3. 8 urlB

But actual sorted values are sorted in below order because the numbers of a string are also concerned as string and not as separate integers.

1. 8 urlB
2. 50 urlC
3. 10 urlA

In order to face this challenge in below code “0” is added in front of “tot” at string creating time, if the “tot” contains a single value. Once all URLs together with “weigh” added to string array, it needs to be sorted. “sort” function sorts all elements of array into ascending order. And “Reverse” function re-arranges the array to descending order. Below code shows how it is done.

```
Array.Sort(myspace)
Array.Reverse(myspace)
```

Then next element (URLs) of sorted array are accessed through a for loop and relevant while another for loop accesses “myspace” session variable to compare objects and identify correct object. Once the right object found the URL & Title of object with weigh (tot) chopped and separated from string of array are added together to newly created object “pg”. Each round of 1st for loop new object with URL, Title and

Weigh added to “URLweighMYSPACE” (list of objects). finally the results are ranked are presented on web layout. The user has selected whether he/she wants to see only top 4 ranked results or all ranked results by selecting option buttons on interface. So results are layout in two ways and below If, else statements used to decide the layout. All details of URLs (Title/ URL/ Weigh) and other HTML tags for layout are added to a Placeholder control (phResult). Two functions “getLiteral” and “gethyperlink” are used to place relevant literals and URLs are clickable hyperlinks. Then place holder displays the layout and ranked URLs on interface.

3 RESULTS AND DISCUSSION

Results

Meta search engine consists of two main components (search/ Ranking) and they have been formed collecting a set of functions, procedures and user defined procedures and the testing. From the design and implementation, the AVQUICK interface was initiated by typing in a query into the search box (php tutorial). Below is the output from the query, in this output you should notice that the myspace.com videos were the first to come out, follow by youtube.com videos as the sample shown. All the results (top 10 videos) extracted from search engines are shown in below screen.

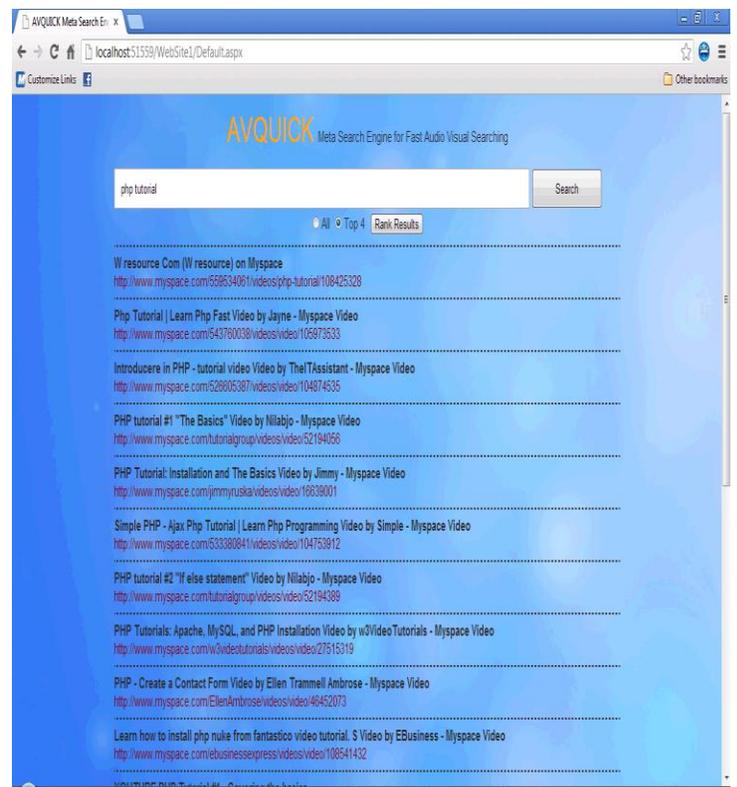


Figure 3 Interface with All the results (unranked)

Before ranking pages the user can select ranking options (Top 4 or All) from screen. Once user selects option “Top 4” and hits enter/ click Rank Results button, and after finished Ranking Top 4 ranked results are shown on interface as below

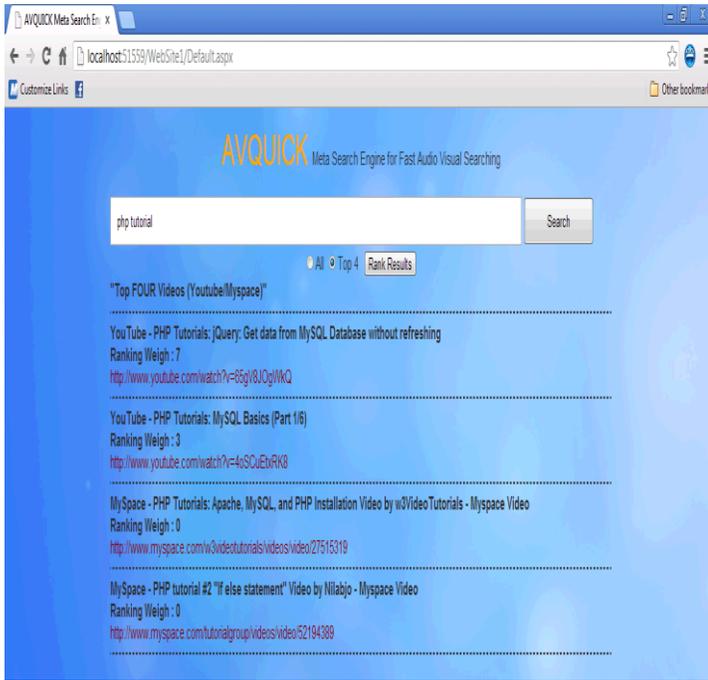


Figure 4 Interface with Top ranked 4 videos

When the user selects option "All" and proceeds ranking, then all the ranked results are displayed on Interface as below.

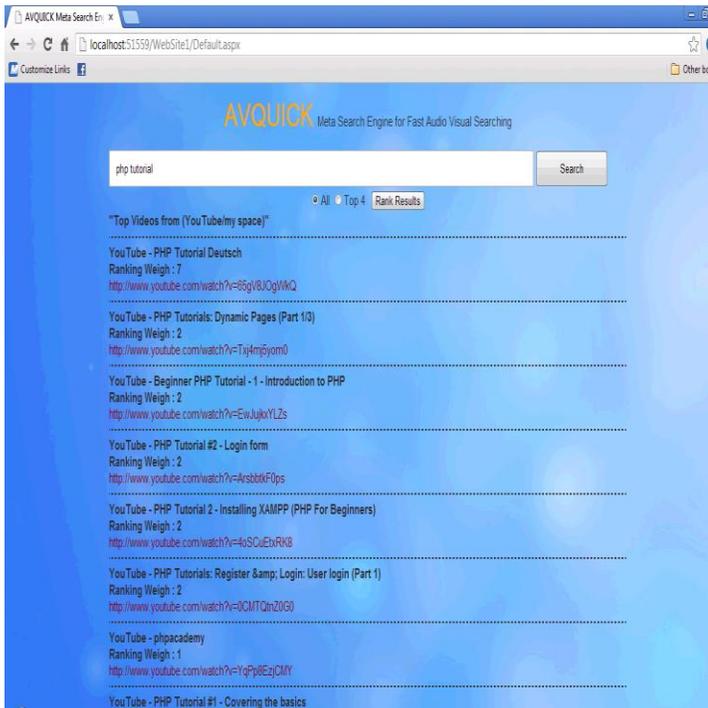


Figure 5 Interface with All ranked pages

Comparison of Results

Evaluates the metasearch engine by cross checking the "Aims & Objectives". One of the major goals of the developed metasearch engine is to search multiple video search engines to collect Top 10 videos from them and to display on the Interface. I have Used only two popular video search engines www.youtube.com and www.myspace.com were linked with

AVQUICK to search for videos. Below it demonstrates that how the AVQUICK searches on multiple engines and presents the correct results.

Used query = "Parrot"

Then AVQUICK searches search engines plugged to collect Top 10 videos from them and to put together on screen as sets one after the other.

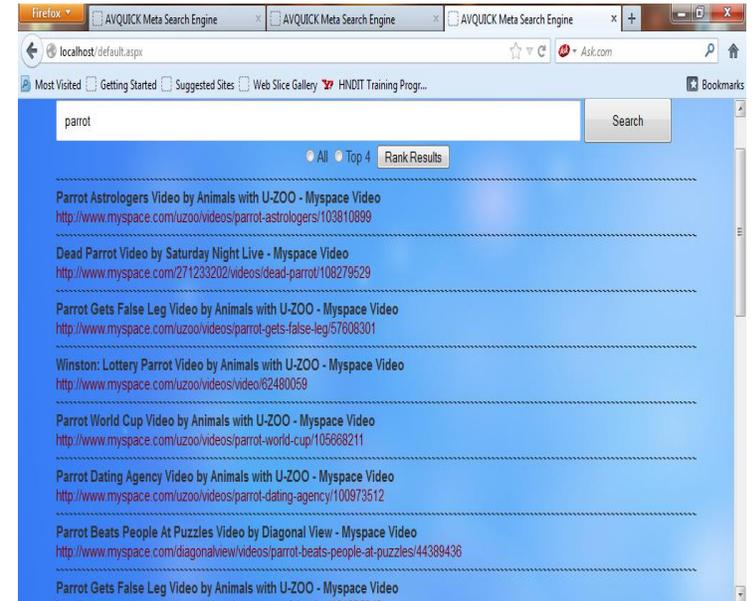


Figure 6 All results (unranked) from Myspace

Results given for the same query by myspace.com as below

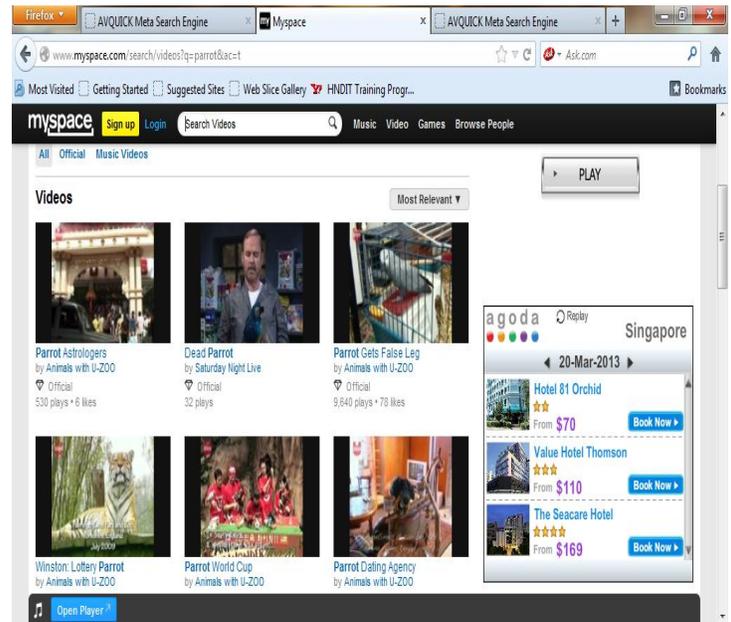


Figure 7 Top 10 results were collected from www.myspace.com.

Above same results are displayed on AVQUICK into same order as below.

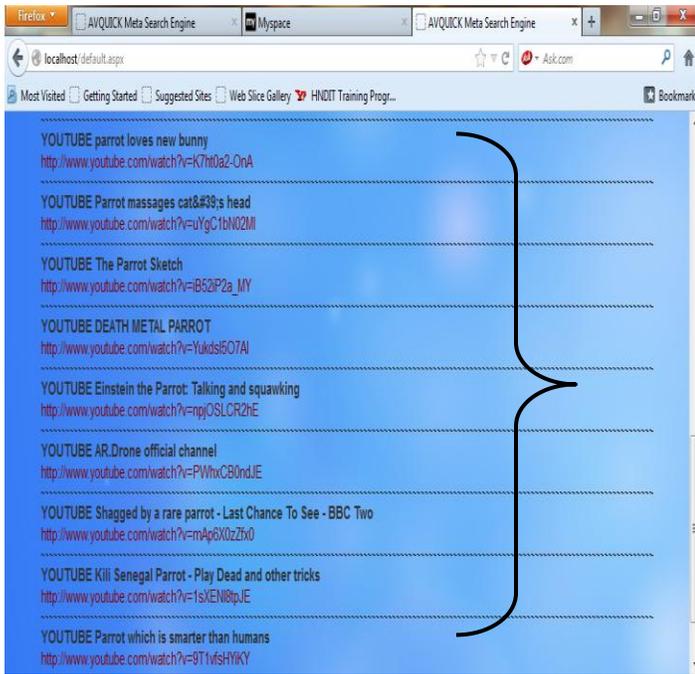


Figure 8 Top results collected from www.youtube.com

Results given by www.youtube.com for the same query used above as bellow

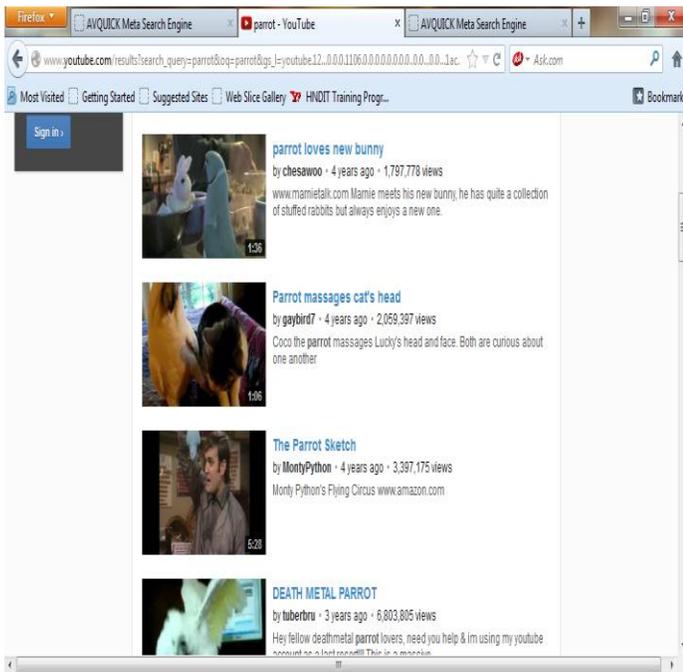


Figure 9 Top 10 results on www.youtube.com

Bellow table (table 4:1) compares the results extracted by different search engines and with results on AVQUICK to check whether expected results are extracted in the expected way.

Results (www.myspace.com)	Results (AVQUICK)
Parrot Astrologers	Parrot Astrologers Video by Animals with U-ZOO - Myspace Video
Dead Parrot	Dead Parrot Video by Saturday Night Live - Myspace Video
Parrot Gets False Leg	Parrot Gets False Leg Video by Animals with U-ZOO - Myspace Video
Winston: Lottery Parrot	Winston: Lottery Parrot Video by Animals with U-ZOO - Myspace Video
Parrot World Cup	Parrot World Cup Video by Animals with U-ZOO - Myspace Video
Parrot Dating Agency	Parrot Dating Agency Video by Animals with U-ZOO - Myspace Video
Parrot Beats People At Puzzles	Parrot Beats People At Puzzles Video by Diagonal View - Myspace Video
Parrot Gets False Leg	Parrot Gets False Leg Video by Animals with U-ZOO - Myspace Video
Parrot Custody Battle	Parrot Custody Battle Video by Animals with U-ZOO - Myspace Video
The Parrot Sketch	The Parrot Sketch Video by Saturday Night Live - Myspace Video
Results (www.youtube.com)	Results (AVQUICK)
parrot loves new bunny	YOUTUBE parrot loves new bunny
Parrot massages cat's head	YOUTUBE Parrot massages cat's head
The Parrot Sketch	YOUTUBE The Parrot Sketch
DEATH METAL PARROT	YOUTUBE DEATH METAL PARROT
Einstein the Parrot: Talking and squawking	YOUTUBE Einstein the Parrot: Talking and squawking
AR.Drone official channel	YOUTUBE AR.Drone official channel
Shagged by a rare parrot - Last Chance To See - BBC Two	YOUTUBE Shagged by a rare parrot - Last Chance To See - BBC Two
Kili Senegal Parrot - Play Dead and other tricks	YOUTUBE Kili Senegal Parrot - Play Dead and other tricks
Parrot which is smarter than humans	YOUTUBE Parrot which is smarter than humans
Parrot Types: Which one is best for You?	YOUTUBE Parrot Types; Which one is best for You?

Table 1 comparing results with search engine results

Discussion

The table 1 shows how Meta search engine is achieved its aims and objective. One of the major goals of the developed Meta search engine is to search multiple video search engines to collect Top 10 videos from them and to display on the Interface. In the table 1 1st 10 records it shows the results comes in the AVQUICK is similar to the myspace.com results. Figure 6 and figure 7 also shows the similarities in Myspace and AVQUICK. In the table 1 next 10 records are show similarity between AVQUICK and Youtube. Figure 8 and figure 9 also shows the similarities in Youtube and AVQUICK. Figure 4 shows the results for top4 ranking. And figure 5 shows ranking weight for all results.

4 FURTHER INVESTIGATION AND FURTHER DEVELOPMENT

Current metasearch engine uses "Web content mining" which considers about analyzing HTML data on web pages to extract URLs and other information with Information retrieval the project uses "Out Link" analysis techniques to gather URLs available on related pages to find weigh/ voting for result pages. Current algorithm applies two rounds because more rounds need more processing power to analyze and extract out link queries from related pages. In the future more research works are carry out to find a way of increasing "Out link" analysis and new algorithm to improve ranking efficiency. www.youtube.com and www.myspace.com have been presently linked with AVQUICK as search engines to extract videos and future works focus on adding more search engines and allowing users to customize multiple engine selection on user Interface. In future a Video recommendation component is expected to add to existing AVQUICK and hence user accounts are introduced and user log records will be stored. Current metasearch engine mainly focused on fulfilling video Meta search requirements of users. But in future existing metasearch engine is expected to extended into different areas like text searching, MP3 searching, etc.

5 CONCLUSION

AVQUICK is a server-based metasearch engine which searches a range of search engines to collect videos to provide higher video satisfaction for Internet users through its' ranking algorithm. At the initial stage AVQUICK accepts the user query and multiple queries are created using user query and then they are sent to different search engines to collect Top 10 videos from each engine. Initial results acquired are displayed on Interface. User can rank the results by selecting ranking options on interface (view all ranked results/ view Top 4 results) and clicking "search" button or hitting enter. "Ranking algorithm used in AVQUICK uses all Outlinks from other pages to decide how much important a pages is. Outlinks from related pages and pages of related pages for a video is concerned as voting for a video page (URL). Pages with higher voting are displayed on top as top ranked pages. In this project extracting Out links from related pages, more related pages of related pages done up to two levels because more rounds take more time and more processing power. Bellow graph explains that which related pages are analyzed to collect outlinks. This process can be followed repeatedly to collect more "Out Links" for analysis which consumes more time and processing power. This project has a limited time and limited resources (limited Internet bandwidth/ Processing speed) and therefore the rounds (cycles) to collect Out link

URLs have been restricted up to 2 cycles. Once the expected amount of out link URLs were gathered, then calculation methodology is applied to count the frequent the result videos URLs available in out link URL collection. Results with higher frequency are identified as important pages and they are displayed on top of the Interface as ranked pages. Ranked results on interface are displayed as two different layouts by allowing user to select their choice. User can select option "Top 4" to view only Top 4 ranked pages from all search engines and user can select option "All" to view ranked videos from top to bottom based on importance. This shows all the videos extracted for initial search.

REFERENCES

- [1] Traditional Search engines - Available from; http://www.readwriteweb.com/archives/search_20_vs_tr.php; Accessed Date: 02nd march 2014;
- [2] The impact of search engine optimization on online advertising market; Bo Xing, Zhangxi Lin ; August 2006; **Publisher:** ACM;
- [3] Video Searching of Internet Users – Available from; http://www.mediapost.com/publications/?fa=Articles.showArticle&art_aid=109260; Accessed Date: 5th march 2014;
- [4] Ranking process of Metasearch Engine – Available from http://en.wikipedia.org/wiki/Metasearch_engine; Accessed date: 5th march 2014;
- [5] Dogpile.com Updates Search Algorithm, Design and content- Available from ; <http://blog.searchenginewatch.com/080424-084455>; Accessed Date: 01/08/2009;
- [6] Updated internet and Web Technologies - Available from; http://en.wikipedia.org/wiki/Search_engines; Accessed Date: 2nd march 2014;
- [7] Usage and Demand for Search engine – Available from <http://www.ibtimes.com/prnews/20090623/demand-for-seo-companie1s-nearly-double-in-the-last-two-years.htm>; Accessed Date: 2nd March 2014;
- [8] Different kinds of software products – Available from <http://www.stsc.hill.af.mil/crosstalk/1995/01/Comparis.asp>; Accessed Date: 07th May 2014;
- [9] Capturing Requirement – Available from <http://209.85.229.132/search?q=cache:-3A7c88igZYJ:www.comp.glam.ac.uk/pages/staff/tdhutchings/chapter4/CHAPTER4.PPT+ssadm+stages&cd=3&hl=en&ct=clnk&gl=uk>; Accessed date: 3rd April 2014;
- [10] Compare and contrast evaluate the strengths and weakness of system development methodologies – Available from <http://www.google.co.uk/imgres?imgurl=http://www.sqa.org.uk/e-learning/SDM01CD/images/pic001.jpg&imgrefurl=http>

[://www.sqa.org.uk/e-learning/SDM01CD/page_02.htm&h=483&w=376&sz=17&tbnid=rEk7XrausCaQnM:&tbnh=129&tbnw=100&prev=/images%3Fq%3Dssadm%2Blife%2Bcycle%2Bdiagram&usg=__NQRIUCXAab0u6JWpD8-AuYxsayU=&ei=hjh0Su_YBY-sjAfNuuSnBg&sa=X&oi=image_result&resnum=2&ct=image](http://www.sqa.org.uk/e-learning/SDM01CD/page_02.htm&h=483&w=376&sz=17&tbnid=rEk7XrausCaQnM:&tbnh=129&tbnw=100&prev=/images%3Fq%3Dssadm%2Blife%2Bcycle%2Bdiagram&usg=__NQRIUCXAab0u6JWpD8-AuYxsayU=&ei=hjh0Su_YBY-sjAfNuuSnBg&sa=X&oi=image_result&resnum=2&ct=image); Accessed date: 3rd April 2014;

- [11] Building an open source meta-search engine; Authors: A. Gulli, A. Signorini; Publisher: ACM; Date: May 2005;
- [12] Helios Metasearch Engine – Available from; http://www.cs.columbia.edu/~wh2138/helios/final_report.pdf; Accessed date: 6th February 2014;
- [13] Expert Agreement and Content Based Reranking in a Meta Search Environment using Mearf; Authors: B. UygurOztekin, George Karypis, Vipin Kumar; Publisher: ACM; Date: May 2002;
- [14] Mamma, the Mother of all metasearch engine – Available from ; <http://www.mamma.com/info/about.html>; Accessed Date: 6th February 2014;
- [15] An advanced direct searching technique applied on compressed video content repositories; Authors: FotisAndritsopoulos, SerafeimPapastefanos, VassilikiMpilili, Christos Theocharatos; September 2008; **Publisher:** ACM;
- [16] An efficient indexing structure for multimedia data; Authors: Thierry Urruty, ChabaneDjeraba, Joemon M. Jose; **Publisher:** ACM; Date: October 2008;