B.E. Project A Report on

User Feeds Relevant Results

Project Members

Shah Manav Bharat, C-453
Surve Akshay Vishwas, C-456

This synopsis has been prepared in fulfillment of Semester-VII
Project A Report as prescribed by the University of Mumbai.

October 2005

Don Bosco Institute of Technology
Premier Automobiles Road, Kurla (W), Mumbai-400070
Website: http://donboscoit.ac.in

Acknowledgement

We thank Prof. Amiya Tripathy, our guide, for channeling our efforts in the right direction. We would also like to thank Dr. S. Krishnamoorthy , *Research and Development* Director Dr. Revathy Sundararajan of Don Bosco Institute of Technology, for their invaluable support. We also thank all the people involved directly or indirectly with this project.

Our sincerest thanks to Donald Knuth, who wrote TEX, without which producing this document would have been a thankless task, and to Hans Hagen who wrote ConTEXt.

Finally, we would like to thank the DUCS *(DBIT Undergraduate Computer Society)*, without which, we would probably never know of TEX.

Dedication

"We dedicate this to each and every Spider who has crawled the Web in search of relevant information."

Contents

1 Synopsis

Internet search engines, use *Web crawlers* (also called web robots, spiders, or wanderers) to download data from the Web. The crawled data is stored on centralized servers, where it is parsed and indexed. The importance of a Web page is an inherently subjective matter, which depends on the readers interests, knowledge and attitudes. But there is still much that can be said about the relative importance of Web pages.

Search engines use a ranking algorithm to determine the order in which matching web pages are returned on the results page. They build indices mostly based on keyword occurrence, link popularity and frequency for query negotiation using these indices. Using these connectivity-based algorithms, they measure the quality of each individual page so that users will receive a ranked page list for their queries.

However this method works well when users' queries are clear and specific. But, in real world, web search queries are often short and ambiguous, and web pages contain a lot diverse and noisy information.

Morever, this is a machine job. It is found that a page can be best judged for its content and quality when viewed by human experts. *UFeedRevs* incorporates the missing human edge over machine supervised search results.

User Feedback systems can be used to foster good behavior and to encourage adherence to rank web pages by a search engine. This report describes a new system called ***UFeedRevs*** - User Feeds Relevant Results, a system which is based on Normal distribution functions to combine feedback and derive new rankings of the web page from existing ones.

UFeedRevs is hereby presented as a distributed search engine framework, in which our core web server answers queries over its own data. Results from search engine will be merged with precious user feedback ratings to generate a ranked list of relevant search results.

The preliminary experiments on a real data set would demonstrate that the system achieves comparable accuracy to produce high quality of query results.

We describe *UFeedRevs* as -

A method for rating Web pages relatively but precisely, effectively measuring the human interest and attention devoted to them.

Keywords:

Search Engine, Relevancy Techniques, Reputation System, Online User Feedback, Relevance Feedback, Learning System, Information Retrieval

2 Introduction

Millions of people all around the world use the Web for their daily life needs. The rate of the growth of the Web is exponential. Traditional methods of information retrieval do not work efficiently for the Web. Not surprisingly, a survey for Realnames[1] reports that 44% of users are frustrated by navigation and search engine use.

Search engines use a ranking algorithm to determine the order in which matching web pages are returned on the results page. They build indices mostly based on keyword occurrence, link popularity and frequency for query negotiation using these indices, as described by Watters & Amoudi[2]. However this method works well when users' queries are clear and specific. However, in real world, web search queries are often short and ambiguous, and web pages contain a lot diverse and noisy information.

User Feedback systems can be used to foster good behavior and to encourage adherence to rank web pages by a search engine. This report describes a new system called ***UFeedRevs*** - User Feeds Relevant Results, a system which is based on Normal distribution functions to combine feedback and derive new rankings of the web page from existing ones. *UFeedRevs* would favor:

- Subjective analysis of ranked Web pages
- Incorporating human-element with mechanized search engines
- Harnessing the power of the millions of users who use the system
- Truly Distributed System

"*The advantage of using UFeedRevs is flexibility and simplicity as well as its foundation on a new era where an user helps himself find relevant results by providing honest feedback.*"

3 Motivation

The World Wide Web with its short history has become a major area of interest for different people. Millions of people all around the world use the Web for their daily life needs and many companies invest on developing a better system for information retrieval on the Web. The rate of the growth of the Web is exponential. However, the more the Web grows the need for search engines arises.

A search engine works on computing algorithm that helps user to find his required piece of information by scanning keywords from user, commonly known as "user query." However, it has been found that results that appear as in response to user query on the result pages are always not in parlance with user requirements. Also, a page is judged by the algorithm based on some standard structured criteria. Thus, a machine always lacks the ability to provide user with his expected results

4 Related Work

The incredible development of Web resources and services has become a motivation for many studies and for companies to invest on developing new search engines or adding new features and abilities to their search engines.

4.1 Page Structure Analysis

The structure of the page is analyzed. Pages with the search terms appearing in the HTML title tag are often assumed to be more relevant than others to the topic. Search engines will also check to see if the search keywords appear near the top of a web page, such as in the headline or in the first few paragraphs of text. Keywords with those with a higher frequency are often assumed to be more relevant than other web pages.

By analyzing how pages link to each other, a search engine can both determine what a page is about and whether that page is considered to be "relevant" and thus deserving of a ranking boost.

However, it is clear that the contents of a Web page could not be sufficient for capturing the huge amount of information. It is thought that Web page layout is a good resource for improving search results, but could not form the sole basis for providing relevant information.

4.2 MetaSearch Engines

A metasearch engine is a search engine that sends user requests to several other search engines and returns the results from each one. They allow users to enter their search criteria only one time and access several search engines simultaneously. The ease of use and high probability of finding the desired page(s) make metasearch engines popular.

Search engines frequently have different ways they expect requests submitted i.e. there are irrelevant 'matches'. Results can vary between metasearch engines based on a large number of variables.

4.3 Web click-thru data

Search engines may watch what results a user selects for a particular search, then eventually drop high-ranking pages that aren't attracting clicks, while promoting lower-ranking pages that do pull in visitors. This is described in detail here[4]

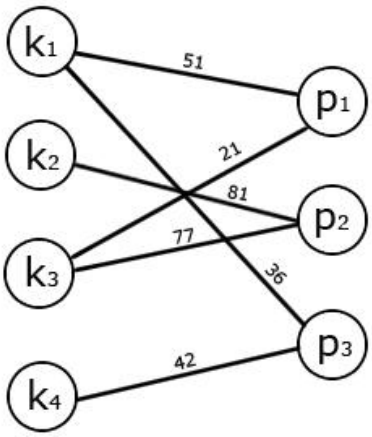

Figure 4.1 Web click-thru graph

The clicks-through data may be very noisy and incomplete for the simple reason that a web page could be found relevant or not only after it is visited. But while the page is being visited, a vote is already counted for the same.

Thus it would introduce inaccurate metadata to associated web pages.

4.4 Hypertext based large scale web-search engines

Apart from the problems of scaling traditional search techniques to data of this magnitude, there are new technical challenges involved with using the additional information present in hypertext to produce better search results. Also the problem of how to effectively deal with uncontrolled hypertext collections where anyone can publish anything they want. This question of how to build a practical large-scale system which can exploit the additional information present in hypertext in answered in [3].

4.5 Adaptive Web Search Based on User's Implicit Preference

In this scheme, search results adapt to users with different information needs. Search systems that adapt to each user's preference[5][6] can be achieved by constructing user profiles based on modified collaborative filtering[11].

However, the system in this case is exhaustive. Also, once in a while user might feel the need to retrieve information as against as he defined in his profile.

4.6 Active Feedback for Effective Web Search

It is based on the concept of balanced tree to present some critical questions for guiding users to have the proper feedback in further searching[7].

An active feedback approach analyzes the distribution of Web pages and provide users the suggestion for proper feedback. Users are guided for fast searching, and are asked interactive questions. Besides, as the system has tried to understand what kind of impacts the feedback will introduce, the item clicked for feedback would highly benefit the systems performance[5]. But user involvement increases with the system working.

4.7 PageRank Algorithm

PageRank is a family of algorithms for assigning numerical weightings to hyperlinked documents (or web pages) indexed by a search engine. It uses link analysis scheme for ranking pages.

PageRank relies on the uniquely democratic nature of the web by using its vast link structure as an indicator of an individual page's value. Google interprets a link from page A to page B as a vote, by page A, for page B. But Google looks at more than the sheer volume of votes, or links a page receives; it also analyzes the page that casts the vote. Votes cast by pages that are themselves "important" weigh more heavily and help to make other pages "important."

The PageRank system is used by the popular search engine Google to help determine a page's relevance or importance. It was developed by Google's founders Larry Page and Sergey Brin[13] while at Stanford University in 1998.

5 Problem Statement

The World Wide Web is expected to be more usable in the future. Information extraction, ambiguity in addresses and names, personalization and multimedia searching are major issues which need to be tackled. The next generations of search tools are expected to be able to extract data to offer high quality responses to users' questions.

Following are some of the questions that we attempt to answer:

- Are the First Results always the Best Results?
- How can one interpret user queries which are usually ill-formed and ambiguous?
- How can one decide that a page is relevant to the query?
- Can algorithms decide accurately on linking of the user query and the web page which is a subjective matter?
- Can a newly uploaded web page compete with current best search results?
- Can users themselves rank the search results?
- Are the users search requirements of a user related to his background?

Providing relevant searches are done by machines. However, it is believed that the best judgment about the importance and quality of Web pages is acquired when they are reviewed and recommended by human experts. Discussion thread recommendation or peer reviews are expected to be used by search engines to improve their results.

Thus we hereby propose search results being ranked not only based on the automatic ranking algorithms but also by using feedback from the users spread across the globe to whom we would cater relevant results.

6 Our Proposed Method - *UFeedRevs*

Recent survey claims search engines to index more than 19 billion pages. So , it is indeed clear that today, focus shifts from size to ***relevancy!*** *UFeedRevs* is here to redefine Web Searches.

Current search engines follow a predefined so called *secret algorithm* to compute web search results. This, however is a machine job. But, it is found that a page can be best judged for its content and quality when viewed by human experts. *UFeedRevs* incorporates the missing human edge over machine supervised search results.

6.1 System Working

UFeedRevs makes use of feedbacks from all the users to re-rank web pages generated as a result of user query. Following is a simple description of the system:

i. User enters *search query*.

ii. Search engine provides current search results.

iii. Collect user feedbacks*(optional)*.

iv. Filter user feedbacks.

v. Compute feedback rating.

vi. Re-rank web pages.

UFeedRevs collects feedbacks from users across the globes. So, in true sense, it acts like a distributed system. Millions of users would give feedbacks. Thus, a pattern of *relevant results* can be identified for the queries which would form the basis for future *relevancy deterministic criteria*.

6.2 Design Plan

It is important to note that *UFeedRevs* is a learning engine. It starts with results ranked same as any other conventional search engine, but later on based on user feedback updates the search results ranking.

Fundamental aspects of System Design to consider would be:

I. Collecting user feedback

After search engine delivers search results in response to the user query, users may visit different search result pages. After they analyze content of the web page, they play a key role in deciding how relevant is the page for the query.

For this purpose, user is provided with a scale ranging from 0 to 100. User, in one single click rates the page. For example, 75 is a nice score for a page to be relevant.

II. Feedback storage process

UFeedRevs uses a propagation mechanism "*centralized approach feedback*" that allows entities to obtain page ranks when required. In this, feedback values are stored in a central server referred to as *Core Server*, and whenever there is a need, users forward their query to the Core Server for the computed page ranks.

III. Filtering user feedbacks

Feedback rating for a query - page relationship would be established based on 'N ' number of feedbacks. Here, 'N' is a sufficiently large number[11]. A large number of feedbacks from the user can be approximated to "Normal Distribution[8]" which is described later in section 6.4

Now, we batch 'n' (<<N) number of collected feedbacks together, analyze those feedbacks for ambiguous or misleading feedbacks. Remove those, using a Feedback filter, explained in detail in The System Architecture. Refer section 6.3

IV. Computing feedback rating

After filtering of user feedbacks, information is passed on to a feedback engine which contains filtered feedbacks. This Feedback Engine computes "Feedback Rating" for the web pages under consideration which is forwarded to Feedback Rating Database. This database contains indexed information about web pages rated by the users.

V. Re-ranking process

When the same query is fired at the Core Server, it considers the feedback ratings from the database and produces re-ranked set of search results.

6.3 System Architecture

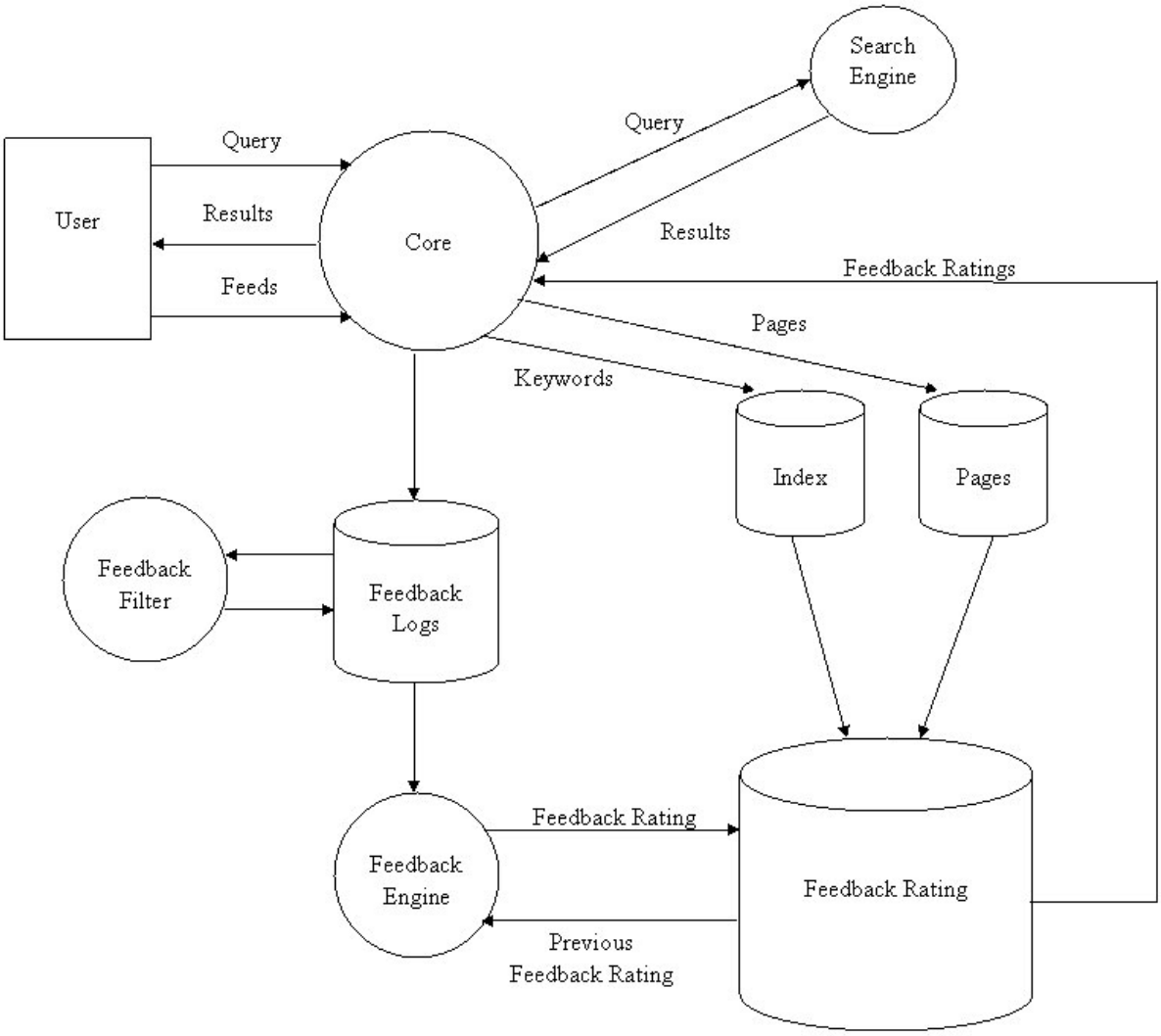

Figure 6.1 System Architecture

The architecture of UFeedRevs is composed of the distributed servers and data structures[9]. For testing and implementation purposes *UFeedRevs* can be used as a supplementary ranking system along with an actual search engine. For such purposes we need to duplicate the database of the indexes.

Major data structures involved include:

- Index The index consists of all the keywords indexed by the search engine. For testing purposes we can consider and duplicate only the list of keywords typed in by the user.
- Pages

  Consists of the URL's of the documents crawled by the search engine indexed by the document id. For testing purposes we can restrict the number of pages to a fixed number with respect to each set of results delivered by the search engine.
- Feedback Logs Consist of the feedback logs generated by the Core. A log would consist of the feedback rating, keywords, document id and additional information like the source identification.
- Feedback Ratings This data structure would consist of the actual feedback rating of a keyword and the document generated by the Feedback Engine[12].

Major components of the above distributed architecture include:

- Core

  The Core is and abstraction of the major functions of UFeedRevs which acts as an interface to the UFeedRevs system. The Core interacts with the users and also the external search engine. It routes the user query to the external search engine and showcases the results to the user along with the feedback rating. It generates a log of feedbacks in the Feedback Log database.
- Feedback Filter

  Feedback Filter is an offline crawler which is used to filter the inconsistencies of the Feedback Logs database.
- Feedback Engine Feedback Engine is the heart of the UFeedRevs system which would calculate the actual feedback rating for the documents and store it in the Feedback Ratings database.

6.4 System Mechanics - Normal Distribution

As described in previous section, a large number of feedbacks are considered for re-ranking process. As the system is distributed widely across the globe, a lot of varying feedbacks can be approximated to "Normal Distribution"

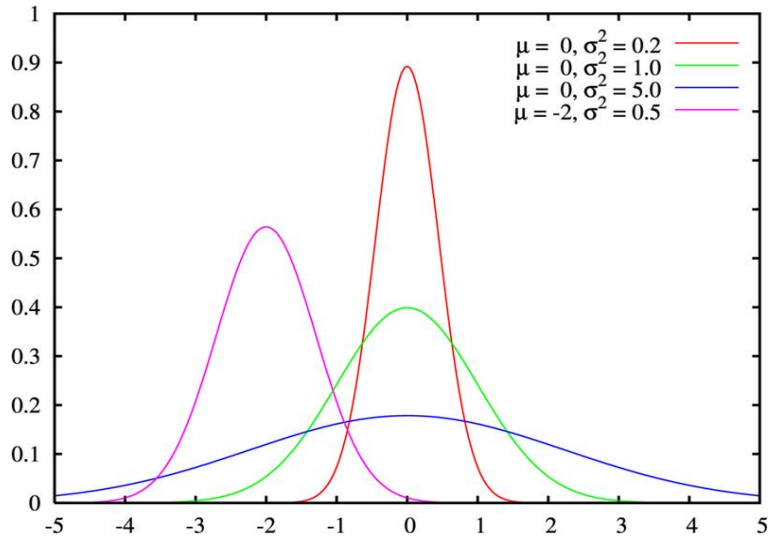

Figure 6.2 Normal Prob. Density Function

The *probability density function* of the *Normal distribution* with mean μ and variance σ^2(equivalently, *standard deviation*) is an example of a Gaussian function,

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

where

Mean μ= N p

Variance σ^2 = N p q

P = Probability that the event is true

q = 1 - p

If a random variable X has this distribution, we write X ~N(μ, σ^2). If μ= 0 and σ= 1, the distribution is called the standard normal distribution and the probability density function reduces to

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right).$$

Some notable qualities of the normal distribution:

- The density function is symmetric about its mean value.
- The mean is also its mode and median.
- 68.27% of the area under the curve is within one standard deviation of the mean.
- 95.45% of the area is within two standard deviations.
- 99.73% of the area is within three standard deviations.
- The inflection points of the curve occur at one standard deviation away from the mean.

Normal distributions have many convenient properties, so random variates with unknown distributions are often assumed to be normal. It is often a good approximation due to a surprising result known as the Central limit Theorem.

Central Limit Theorem:

The Central Limit Theorem states that -

" *If the summed variables have a finite variance then they will be approximately normally distributed.*"

The practical importance of the central limit theorem is that the normal distribution can be used as an approximation to some other distributions. This clearly explains the fact why our system, *UFeedRevs* uses Normal Distribution.

6.5 Aging and Flushing

Old feedback may not always be relevant for the actual reputation rating, because the web page may change its behavior over time. What is needed is a model in which old feedback is given less weight than more recent feedback. This translates into gradually forgetting old feedback. This can be achieved by introducing a *forgetting factor* which can be adjusted according to the expected rapidity of change in the observed entity.

As and when new feedbacks are made to enter the system, the mean of older feedbacks are calculated (for N feedbacks). The weights due to older feedbacks are

then flushed off from the system and new feedback information are appended. Mean is computed once again.

Thus, mean of the older feedbacks persist. However, older feedbacks are forgotten i.e. almost removed.

6.6 Design Goals

- Our main goal is to *improve the quality of web search engines.* As the collection size of web page grows, we need tools that have very high precision (number of relevant documents returned, say in the top tens of results). Indeed, we want our notion of "relevant" to only include the very best documents since there may be tens of thousands of slightly relevant documents.

- By implementing *UFeedRevs*, there is quite a bit of recent optimism that users having community feeling coming together and help in filtering out the best results.

- Another important goal was to *build systems that reasonable numbers of people can actually use* for defined purposes. Usage was important to us because we think some of the most interesting research will involve the vast amount of usage data that is available from modern web systems.

- Our final design goal was to build an *architecture that can learn by itself. UFeedRevs* is to set up an environment where users from across the globe can come in quickly, while *UFeedRevs* process large chunks of the web and produce interesting results that would have been very difficult to produce otherwise without involvement of the human factor.

7 Evaluation Metrics

Our goal is to apply the algorithm in a distributed Internet search engine system, where it should be able to provide users the same quality results as what the original algorithm does, but with an icing of user feedback.

In this section, several metrics are described, which can be used to compare two ranked lists.

7.1 Precision

The Precision is applied to measure the performance of our proposed algorithm. Given a query Q, let R be the set of the relevant pages to the query and | R | be the size of the set; let A be the set of top 20 results returned by our system. Precision is defined as:

$$Precision = \frac{|R \cap A|}{|A|} \tag{7.1}$$

Precision measures the degree to which the algorithm produces an accurate result.

7.2 Probation

Given a query, we ask the ten volunteers to identify top 10 authoritative pages according to their own judgments. The set of 10 authoritative web-pages is denoted by M and the set of top 10 results returned by search engines is denoted by N.

$$Probation = \frac{|M \cap N|}{|M|} \tag{7.2}$$

Probation measures the ability of the algorithm to produce pages that are most likely to be visited by users. Probation measurement is more relevant to users' degree of satisfactory on the performance of a web search engine.

7.3 Relevance of search results

The results that are displayed for a particular query should be relevant to the query. Relevancy is a relative term and therefore the user is the key who would decide on amount of relevancy incurred by the web page for a particular query.

7.4 Result overlap

These results would then be compared with results of some standard web search engines. This overlap criteria is beneficial to check out how much has the user feedback created an impact on the search system.

Given the same data set, the distributed search engine system is expected to return a very similar, if not identical, ranked page list of the user feedback results.

8 Expected Results

We are expecting *UFeedRevs* would perform exceptionally well under controlled and ideal environments. Considering the user bias and SEO's[1] we would need to make the system more robust.

The tool would add a more democratic voice to search technology on the Web.

[1] Search engine optimizations are techniques to improve Website ranks in the search engines. Today, they are called cheapest and most effective marketing tool available.

9 Probable Drawbacks

UFeedRevs is expected to perform well under standard defined conditions. However, it may be subjected to following foreseen drawbacks:

- Unfairly high ratings ("*ballot stuffing*"):

  A company web site can incorporate collaboration with a group of users in order to be given unfairly high ratings by them. This will have the effect of inflating a company reputation, therefore allowing the web page of company a higher ranking than it deserves.

- Unfairly low ratings ("*bad-mouthing*"):

  Web page of company can incorporate collaboration with a group of users in order to "bad-mouth" other deserving web page rank. In such a situation, the users provide unfairly negative ratings to the targeted web pages, thus lowering their reputation.

10 Intended Areas of Application

UFeedRevs is expected to form a building block in redefining web search. The more we think about *UFeedRevs*, the more we get entangled in thought process imagining about its wide - scope applications, just like a spider caught in its own web.

Following are some of the intended areas of Application:

- General Search

  Purpose of *UFeedRevs* is to incorporate the missing human element into existing search engine mechanism. It will provide with search results which are viewed and rated "good" by other users.

- Content Specific Search

  Many documents of web are user-centric and require explicit feedback from humans. Machines cannot rank these accurately. For example, users can grade a particular tutorial or an essay as of high quality.

- Academic Search Engine

  The Web has become increasingly commercial over time. With *UFeedRevs*, we have a strong goal to push more development and understanding into the *academic realm.*

- Query Knowledge Discovery

  User queries are often ill-formed and ambiguous. *UFeedRevs* identifies a pattern among users i.e human beings that when there is an ill formed query 'Q', these are the pages(P_0, P_1, P_2 ..... P_n) that user expects from the system.

11 Conclusion and Future Work

The thought process in designing *UFeedRevs* has been grossly intuitive. We believe our approach to the problem of providing relevant search results to the user is innovative. The system processes involved like learning, ageing, flushing and analysis of feedbacks is derived from extensively researched areas of reputations systems, machine learning, rating systems and collaborative filtering.

Our future work has two directions. On the implementation front we intend to build the *UFeedRevs* system as highly scalable distributed system on terms of the above architecture. On the theoretical side, we would try to better understand the behavior of the system in accordance to various parameters thereof. We would like to build suitable mathematical models that approximate the behavior of the system and thus help in enhancing the overall functioning of the system.

References

[1] Sadi, S., & Jamali, H.R. (2004). "Shifts in search engine development: A review of past, present and future trends in research on search engines". Webology, 1(2), Article 6. Available at: http://www.webology.ir/2004/v1n2/a6.html

[2] Watters, C. & Amoudi, G. (2003). GeoSearcher: location-based ranking of search engine results. Journal of the American Society for Information Science and Technology, 54(2), 140-151.

[3] Brin, S. & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. Proceedings of the 7th International WWW Conference, Brisbane, Australia, 107-117.

[4] Wei-Ying Ma, Zheng Chen & Hua-Jun Zeng. Microsoft Research Asia, 5F, Sigma Center, 49 Zhichun Road, Beijing 100080, P.R.China@microsoft.com

[5] Kazunari Sugiyama, Kenji Hatano & Shunsuke Uemura. Adaptive Web Search Based on User's Implicit Preference. Nara Institute of Science and Technology 8916-5 Takayama, Ikoma, Nara 630-0192, Japan.

[6] Masatoshi Yoshikawa. Adaptive Web Search Based on User Profile Constructed without Any Effort from Users. Nagoya University, Furo, Chikusa, Nagoya, Aichi 4648601, Japan. yosikawa@itc.nagoyau.ac.jp

[7] Ray-I Chang, Jan-Ming Ho. Active Feedback for Effective Web Search. September 2005, Technical Report No. TR-IIS-05-013 http://www.iis.sinica.edu.tw/LIB/TechReport/tr2005/tr05.html

[8] Eric W. Weisstein. "Normal Distribution." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/NormalDistribution.html

[9] W. Buntine, J. Lofstrom, J. Perkio, S. Perttu, V. Poroshin, T. Silander, H. Tirri, A. Tuominen and V. Tuulos, July 5, 2005. A Scalable Topic-Based Open Source Search Engine [hiit-2004-14.pdf, Architecture stuff]

[10] Chrysanthos Dellarocas, Immunizing Online Reputation Reporting Systems Against Unfair Ratings and Discriminatory Behavior. Chrysanthos Dellarocas Sloan School of Management Massachusetts Institute of Technology Room E53-315, Cambridge, MA, 02139, USA +1 (617) 258-8115 dell@mit.edu

[11] Boris Chidlovskii, Natalie S. Glance and M. Antonietta Grasso. Collaborative Re-Ranking of Search Results. Xerox Research Centre Europe 6 chemin de Maupertuis 38240 Meylan, France@xrce.xerox.com

[12] R. Guha. Open Rating Systems. IBM Research Almaden, USA and Knowledge Systems Lab, Stanford guha@cs.stanford.edu

[13] Sergey Brin, Rajeev Motwani, & Terry Winograd.
PageRank Citation Ranking: Bringing Order to the Web Lawrence Page.
http://dbpubs.stanford.edu:8090/pub/showDoc.pdf

Index

Colophon

This document was produced using ConTEXt, an open-source document preparation system. Occasionally, documentation of the concerned was referred to for text-editing. The Content list was placed using the completecontent and the index was prepared using placeindex.

GNU Emacs was the text editor used.