An Efficient Clustering System for the Measure of Page (Document) Authoritativeness

[1]F. U. Ogban, [2]P. O. Asagba, [3]Olumide Owolabi
[1]Department of Mathematics, Statistics, & Computer Science, Faculty of Science
University of Calabar, Nigeria
[2]Department of Computer Science, Faculty of Physical & Information Technology, University of Port Harcourt, Nigeria
[3]Computer Center, University of Abuja, Nigeria

ABSTRACT

A collection of documents $\mathbf{D_1}$ of a search result $\mathbf{R_1}$ is a cluster if all the documents in $\mathbf{D_1}$ are similar in a way and dissimilar to another collection say $\mathbf{D_2}$ for a given query $\mathbf{Q_1}$. Implying that, given a new query $\mathbf{Q_2}$, the search result $\mathbf{R_2}$ may pose an intersection or a union of documents from $\mathbf{D_1}$ and $\mathbf{D_2}$ or more to form $\mathbf{D_{3.}}$ However within these collections say $\mathbf{D_1}$**,** $\mathbf{D_2}$**,** $\mathbf{D_3}$ etc, one or two pages certainly would be better in relevance to the query that invokes them. Such a page is regarded being 'authoritative' than others. Therefore in a query context, a given search result has pages of authority. The most important measure of a search engine's efficiency is the quality of its search results. This work seeks to cluster search results to ease the matching of searched documents with user's need by attaching a ***page authority value*** (***pav***). We developed a classifier that falls in the margin of supervised and unsupervised learning which would be computationally feasible and producing most authoritative pages. A novel searching and clustering engine was developed using several measure-factors such as anchor text, proximity, page rank, and features of neighbors to rate the pages so searched. Documents or corpora of known measures from the Text Retrieval Conference (TREC), the Initiative for the Evaluation of XML Retrieval (INEX) and Reuter's Collection, were fed into our work and evaluated comparatively with existing search engines (Google, VIVISIMO and Wikipedia). We got very impressive results based on our evaluation. Additionally, our system could add a value – ***pav*** to every searched and classified page to indicate a page's relevance over the other. A document is a good match to a query if the document model is likely to generate the query, which will in turn happen if the document contains the query words often. This approach thus provides a different realization of some of the basic ideas for document ranking which could be applied through some acceptable rules: number of occurrence, document zone and relevance measures. The biggest problem facing users of web search engines today is the quality of the results they get back. While the results are often amusing and expand users' horizons, they are often frustrating and consume precious time. We have made available a better page ranker that do not depend heavily on the page developer's inflicted weights but considers the actual factors within and without the target page. Though very experimental on research collections, the user can within the collection of the first ten search results listing, extract his or her relevant pages with ease.

Keywords: page Authoritativeness, page Rank, search results, clustering algorithm, web crawling.

1.0 INTRODUCTION

The Internet is growing with an increasing rate, and it is obvious that it will be difficult to search for information in this gigantic digital library. Eric Schmidt, the CEO of Google (www.google.com;2013), the world's largest index of the Internet, estimated the size at roughly 5 million terabytes of data and it's constantly expanding by 100 terabytes per month. By figure 1, the estimated size of the indexed pages of the World Wide Web ("Internet"), by Wednesday May 1st 2013, indicates that there are about 14.41 Billion pages on the World-Wide Web, on about 22 million servers.

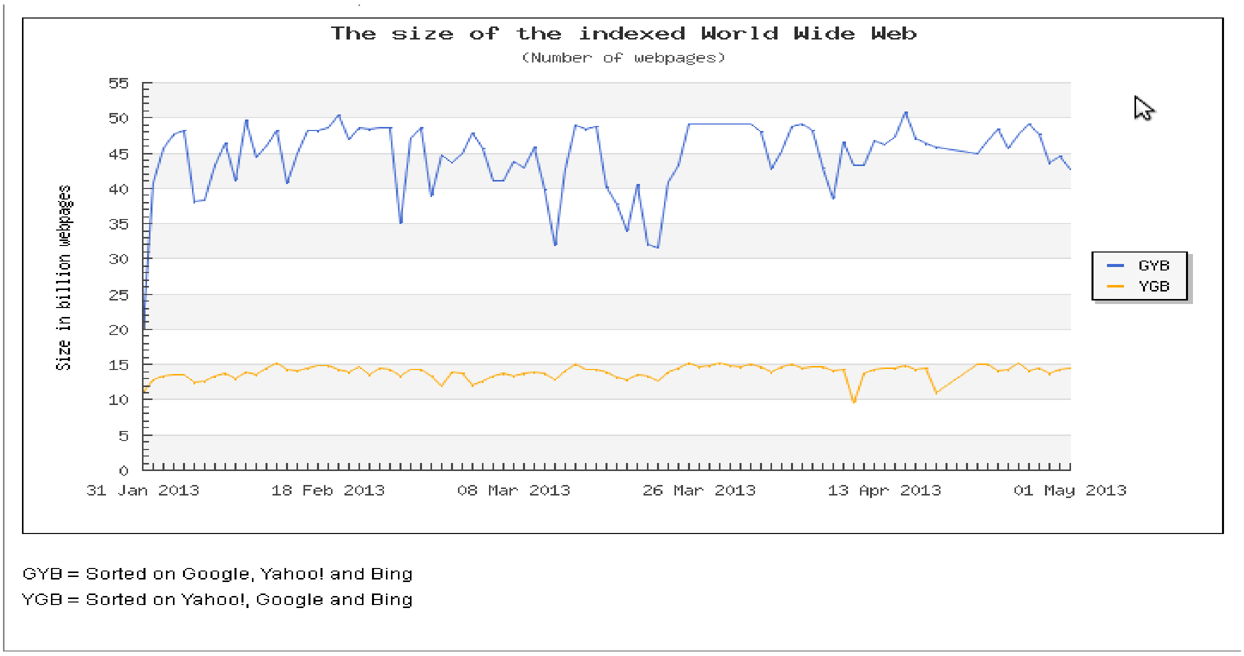

Fig. 1: Size of the World Wide Web and growth rate. (**Source**: WorldWideWebSize.com, 2013).

Retrieval of text information is a difficult task. The problem can be either that the information is misinterpreted because of natural language ambiguities or the information need can be imprecisely or vaguely defined by the user. This calls for improved automatic methods for searching and organizing text documents so information of interest can be accessed fast and accurately.

Classification of web page content is essential to many tasks in web information retrieval such *as* maintaining web directories and focused crawling. The uncontrolled nature of web content presents additional challenges to web page classification as compared to traditional text classification, but the interconnected nature of hypertext also provides features that can assist the process. Classification plays a vital role in many information management and retrieval tasks. On the Web, classification of page content is essential to focused crawling, to the assisted development of web directories, to topic-specific web link analysis, and to analysis of the topical structure of the Web. Web page classification can also help improve the quality of web search.

1.1 General review and problem definition

The increasing growth rate of the Internet's content, especially now that everyone wants to own a personal 'blog', has made it very difficult to search for 'relevant' information in this gigantic digital library. The estimated size of about 997 million pages on the World-Wide Web, on about 8 million servers is huge. Retrieval of text information is a difficult task. The problem can be either that the information is misinterpreted because of natural language ambiguities or the information need can be imprecisely or vaguely defined by the user. This calls for improved automatic methods for searching and organizing text documents so information of interest can be accessed fast and accurately. This work aims at developing a classifier that falls in the margin of supervised and unsupervised learning which would be computationally feasible and aimed at producing most authoritative pages. The learning scheme lies somewhere between supervised and unsupervised.

Page classification also known as web page classification is the process of assigning a page to one or more predefined category label. The field is often posed as a supervised learning problem (Mitchell, 1997) in which a set of labeled data is used to train a classifier which can be applied to label future examples.

Web page classification can be divided into multiple sub-problems: subject classification, functional classification, sentiment classification, etc. While subject classification is concerned about the subject or topic of the page; for example judging whether a page is about "art", "business" or "sport" is an instance of subject classification, functional classification cares about the role that the page plays. For example, deciding a page to be a "personal homepage," "course page," or "admission page" is an instance of a functional classification. Sentiment classification focuses on the opinion that is presented in a web page, that is, the author's attitude about some particular topic. Other types of classification include genre classification (Zu and Stein; 2004), search Engine spam classification (Gyongyi and Garcia-Molina; 2005b), (Castillo, Donato, et al; 2007) and so on.

2.0 RELATED WORKS

Query ambiguity is among the problems that undermine the quality of search results. For example, the query term “bank” could mean the border of a water area or a financial establishment. Various approaches have been proposed to improve retrieval quality by disambiguating query terms. (Chekuri et al;1997) studied automatic web page classification in order to increase the precision of web search.

Search results are usually presented in a ranked list. However, presenting categorized, or clustered, results could be more useful to users. An approach proposed by (Chen and Dumais; 2000) classifies search results into a predefined hierarchical structure and presents the categorized view of the results to the user. Their study established that the category interface is liked by the users better than the result list interface, and is more effective for users to find the desired information, compared to the approach suggested by (Chekuri et al., 1999), which is less efficient at query time because it categorizes web pages without checks.

PageRank calculates the authoritativeness of web pages based on a graph constructed by web pages and their hyperlinks, without considering the topic of each page. Since then, much research has been explored to differentiate authorities of different topics. (Haveliwala, 2002) proposed Topic-sensitive PageRank, which performs multiple PageRank calculations, one for each topic. When computing the PageRank score for each category, the random surfer jumps to a page in that category at random rather than just any web page. This has the effect of biasing the PageRank to that topic. This approach needs a set of pages that are accurately classified. (Nie et al., 2006) proposed another web ranking algorithm that considers the topics of web pages. In that work, the contribution that each category has to the authority of web pages is distinguished by means of soft classification, in which a probability distribution is given for a web page being in each category. In order to answer the question “to what granularity of topic the computation of biased page ranks make sense,” (Kohlschutter et al., 2007) conducted analysis on Object Dynamic Pages (ODP) categories, and showed that ranking performance increases with the ODP level up to a certain point. It seems further research along this direction is quite promising. Although, there are surveys on textual classification that mention web content, they lack an analysis of features specific to the web. (Sebastiani, 2002) mainly focused on traditional textual classification. (Chakrabarti, 2000) and (Kosala and Blockeel, 2000) reviewed web mining research in general as opposed to concentrating on classification. (Mladenic, 1999) reviewed a number of text-learning intelligent agents, some of which are web-specific. However, her focus was on document representation and feature selection. (Getoor and Diehl, 2005) reviewed data mining techniques which explicitly consider links among objects, with web classification being one of such areas. (Fiirnkranz, 2005) reviews various aspects of web mining, including a brief discussion on the use of link structure to improve web classification. Closer to the present article is the work by (Choi and Yao, 2005), which described the state of the art techniques and subsystems, used to build automatic web page classification systems.

3.0 The Fuzzy C-means Algorithm

Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. This method (developed by Dunn in 1973 and improved by Bezdek in 1981) is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{c} u_{ij}^{m} \left\| x_i - c_j \right\|^2 , 1 \le m < \infty \qquad 1$$

where m is any real number greater than 1,
u_{ij} is the degree of membership of x_i in the cluster j,
x_i is the ith of d-dimensional measured data,
c_j is the d-dimension center of the cluster, and
||*|| is any norm expressing the similarity between any measured data and the center.

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\| x_i - c_j \|}{\| x_i - c_k \|} \right)^{\frac{2}{m-1}}} , \qquad 2$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \bullet x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \quad 3$$

This iteration will stop when

$$\max_{ij}\left\{u_{ij}^{(k+1)} - u_{ij}^{(k)} |\right\} < \ell \quad 4$$

Where ℓ *is a termination criterion between 0 and 1,*
and k *is the iteration steps.*

This procedure converges to a local minimum or a saddle point of J_m.
The algorithm is composed of the following steps:

1. Collect and initialize the objective function $U = \lfloor u_{ij} \rfloor$ in a matrix of equation 2.
2. At K-step, Calculate the centers of the vectors $C^{(k)}=[c_j]$ with $U^{(k)}$ as in equation 3.
3. Update $U^{(k)}$, $U^{(k+1)}$ as is in equation 2.
4. Test for the difference between $U^{(k)}$, $U^{(k+1)}$ using the termination criteria of equation 4.
   - if the absolute value of the difference is less than ℓ then stop
   - else return to step 2
5. Stop iteration.

Fuzzy partitioning algorithm

It is worthy of note that, data are bound to each cluster by means of a Membership Function, which represents the fuzzy behavior of this algorithm. To do that, we simply have to build an appropriate matrix named U whose factors are numbers between 0 and 1, and represent the degree of membership between data and centers of clusters. In most retrieval systems, this matrix is referred to as the ***term versus documents*** matrix.

4.0 Methodology

A document is a good match to a query if the document model is likely to generate the query, in other words, the document must have the query words more often. This approach thus provides a different realization of some of the basic ideas for document ranking which could be applied through some acceptable rules: (i) A document or zone (Topic, Abstract, and Introduction etc) that mentions a query term more often has more to do with that query and therefore should receive a higher score in ranking. (ii) The exact ordering of the terms in a document is ignored but the number of occurrences of each term is material and we assign to each term in a document a weight for that term, which depends on the number of occurrence of the term in the document – term frequency. (iii) All terms are considered equally important when it comes to assessing relevancy on a query.

Finite Automata and Language Model

A language model is a function that puts a probability measure over strings drawn from some vocabulary. That is for a language model **M** over an alphabet $\sum$, the sum of the probability measure P over a string s is equal to 1

$$\sum_{s \in \sum} P(s) = 1 \quad 5$$

One simple kind of language model is equivalent to a probabilistic finite automaton of figure 2 consisting of just a single node with a single probability distribution over producing different terms, so that

$$\sum_{t\in V} P(t) = 1$$

,

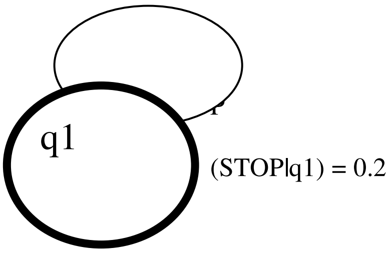

Fig. 2 Probabilistic finite automata language model for equation 5

After generating each word, we decide whether to stop or to loop around and then produce another word, and so the model also requires a probability of stopping in the finishing state. Such a model places a probability distribution over any sequence of words. Probabilities over sequences of terms can be built using the chain rule equation 6 to decompose the probability of a sequence of events into probability of each successive event conditioned on earlier events:

$$P(t_1t_2t_3t_4) = P(t_1)P(t_2 | t_1)P(t_3 | t_1t_2)P(t_4 | t_1t_2t_3) \qquad 6$$

The simplest form of language model simply throws away all conditioning context as in equation 7 and estimates each term independently. Such a model is referred to as **Unigram language model:**

$$P_{uni}(t_1t_2t_3t_4) = P(t_1)P(t_2)P(t_3) \qquad 7$$

There are many more complex kinds of language models such as **bigram language model**, equation 8 which conditions on the previous term:

$$P_{bi}(t_1t_2t_3t_4) = P(t_1)P(t_2 | t_1)P(t_3 | t_2)P(t_4 | t_3) \qquad 8$$

And even more complex grammar-based language models such as probabilistic context-free grammars. Such models are vital for tasks like speech recognition, spelling correction, and machine translation, where you need the probability of a term conditioned on surrounding context. However, most language modeling work in Information Retrieval has used unigram language model. This is because Information retrieval does not directly depend on the structure of the sentence to the extent that other tasks like speech recognition do. Besides, in unigram language models, the order of words is irrelevant, and so such models are often called “Bag of words” models.

The Query Likelihood Model

The original and basic method for using language models in information retrieval is the query likelihood model. In it, we construct from each document **d** in the collection, a language model **M_{d.}** Our goal is to rank documents by $P(d | q)$, where the probability of a document is interpreted as the likelihood that it is relevant to the query. Using Bayes rule in this context, we have:

$$P(d | q) = P(q | d)P(d) / P(q) \qquad 9$$

However, since $P(q)$ and $P(d)$ is the same for all documents, they can be ignored. Thus above equation would be:

$$P(d | q) = P(q | d) \qquad 10$$

But we could implement a genuine prior, which could include criteria like authority, length genre, newness, and number of previous people who had read the document. Given these simplifications, we return results ranked by simply $P(q | d)$, the probability of the query would be observed as a random sample from the respective documents model. The most common way to do this is using the multinomial unigram Language Model, which is equivalent to a multinomial Bayes model where the documents are the classes, each treated in the estimation as a separate “language”:

$$P(q | m_d) = K_q \prod_{t\in V} P(t|m_d)^{tf_{t,d}} \qquad 11$$

For retrieval based on a language model, we treat the generation of queries as a random process. The approach is to:

i. Infer a Language Model for each document

ii. Estimate $P(q \mid m_{di})$ the probability of generating the query according to each of these documents models.

iii. Rank the document according to these probabilities.

The intuition of the basic model is that the user has a prototype document in mind, and generates a query on words that appear in this document. Often, users have a reasonable idea of terms that are likely to occur in documents of interest and they will choose query terms that distinguish these documents from others in the collection. Collections statistics are an integral part of the language model, rather than being used heuristically as in many other approaches.

5.0 Page Ranking Improvement Design

The most obvious solution will be to get as many incoming links as you can, while shutting down your site and not linking to anyone else. Wikipedia is one example of such closed system, as every outgoing link on Wikipedia is ‘no follow’. However things are not as easy, as Google has tweaked their algorithm over the years and in effort to fight this kind of Page Rank conservation they have probably invented numerous algorithms to detect and even punish such sites.

The analysis of hyperlinks and the graph structure of the Web has been instrumental in the development of web search. Such link analysis is one of many factors considered by web search engines in computing a composite score for a web page on any given query. We begin by reviewing some basics of the Web as a graph then proceed to the technical development of the elements of link analysis for ranking. Figure 3 shows our proposed page ranker, where matched and indexed results (as input) are ranked based on several factors including accepting several incoming links but not linking to other pages.

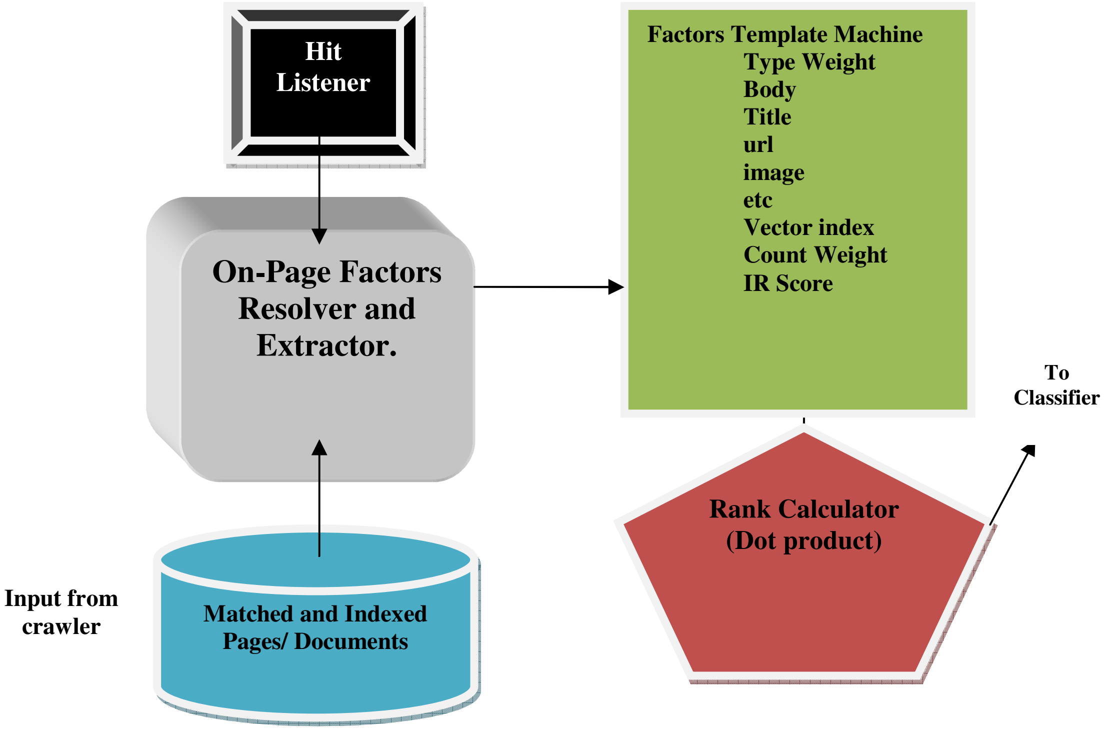

Fig. 3: Our proposed page ranker

Matched and indexed pages

This is a collection of documents arising from a successful crawling, matching and indexing of the resultant given query vis a vis the web based corpus. The result of this processes in presented in either an inverted file format or the forward file format to ease hashing for stem words and expressions. In our page ranker, the collection of matched and indexed pages acts now as an input to the ranker.

On-page Factor Revolver and Extractor

On every accepted page of match, there are several page oriented factors that proves to be the major ingredients of the page. These factors such as tags, keywords, words frequency, topics, bolded text, hyperlinks etc are the bases for the measure of the reason for a page being more important to a query than the other. This module extracts and resolves these comparative measures into units for easy weighting and analysis.

Hit Listener: Given any query for a search to commence, this module monitors the pages that matches the given query in a hyperlink induces text (HIT) function format. This listener as called considers two basic information retrieval processes -the Authority and the Hub measures. A hyperlink from say page **A** to page **B**, gives page **B** an authority and page **A** regarded as a hub. This inbound and outbound linkage is used to indicate a power of influence on and within pages. This listener measures such influences towards the determination of better pages.

Factor template machine

A template machine is a place holder designed to take grab of collections in a predefined structure. Such build structures will then become a platform for ease comparison and a bench mark for any newly formed template instances.

Rank Calculator

Majority of the indices for the measures to be generated by the discussed modules above, are presented in the form of vectors of similarities and dissimilarities. The rank calculator therefore is designed to measure the rank of a particular page using the mathematical frame of dot products.

6.0 RESULTS

Several results were produced from the series of data inputs. One of such is that shown in Table 1. It is an experimental measure of the comparative average results (in thousands) of ten categories of queries (10 queries in each category) administered to the four search engines(Vivosimo, Google, wikipedia and Our System) considering the total match found (TMF), the Total relevant pages (TRP) and the non-relevant pages(NRP) .

Table 1: Showing a comparative average results (in thousands) of ten categories of queries (10 queries in each category) administered to the four search engines

S/N	QUERY	Vivosimo			Google			Wikipedia			Our system		
		TMF	TPR	NRP	TMF	TPR	NRP	TMF	TPR	NRP	TMF	TPR	NRP
1	A	434	2	18	678	3	17	569	3	17	311	5	15
2	B	299	4	16	344	1	19	442	4	16	233	6	14
3	C	308	5	15	563	3	17	466	4	16	214	4	16
4	D	400	7	13	623	4	16	611	2	18	321	5	15
5	E	311	4	16	345	7	13	412	6	14	291	4	16
6	F	300	5	15	476	3	17	407	5	15	266	4	16
7	G	298	3	17	449	4	16	412	2	18	201	5	15
8	H	379	6	14	512	5	15	500	5	15	227	6	14
9	I	403	2	18	570	3	17	417	3	17	301	3	17
10	J	279	3	17	455	4	16	399	5	15	193	4	16

TMF – Total match found **TPR –** Total pages relevant (first 20 links) **NRP –** None relevant pages

Precision

In the field of information retrieval, **precision** is the fraction of retrieved documents that are relevant to the search:

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|} \quad 12$$

Precision takes all retrieved documents into account, but it can also be evaluated at a given cut-off rank, considering only the topmost results returned by the system. This measure is called **precision at n** or **P@n**. For example, for a text search on a set of documents, precision is the number of correct results divided by the number of all returned results. Precision is also used with recall, the percent of *all* relevant documents that is returned by the search. The two measures are sometimes used together in the F1 Score (or f-measure) to provide a single measurement for a system. Note that the meaning and usage of "precision" in the field of Information Retrieval differs from the definition of accuracy and precision within other branches of science and technology. (Fiirnkranz, 2005)

Recall

Recall in information retrieval is the fraction of the documents that are relevant to the query that are successfully retrieved. This is shown in equation 13.

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|} \quad 13$$

For example for text search on a set of documents recall is the number of correct results divided by the number of results that should have been returned. In binary classification, recall is called sensitivity. So it can be looked at as the probability that a relevant document is retrieved by the query. It is trivial to achieve recall of 100% by returning all documents in response to any query. Therefore, recall alone is not enough but one needs to measure the number of non-relevant documents also, for example by computing the precision. A measure that combines precision and recall is the harmonic mean of precision and recall, the traditional F-measure or balanced F-score: This is shown in equation 14.

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad 14$$

This is also known as the F_1 measure, because recall and precision are evenly weighted. These are shown in Table 2.

Table 2 showing the calculated recall, Precision and F-measure of Table 1

S/N	QUERY TYPE	Vivosimo			Google			Wikipedia			Our system		
		TMF	TPR	NRP	TMF	TPR	NRP	TMF	TPR	NRP	TMF	TPR	NRP
1	A	434	2	18	678	3	17	569	3	17	311	5	15
2	B	299	4	16	344	1	19	442	4	16	233	6	14
3	C	308	5	15	563	3	17	466	4	16	214	4	16
4	D7	400	7	13	623	4	16	611	2	18	321	5	15
5	E	311	4	16	345	7	13	412	6	14	291	4	16
6	F	300	5	15	476	3	17	407	5	15	266	4	16
7	G	298	3	17	449	4	16	412	2	18	201	5	15
8	H	379	6	14	512	5	15	500	5	15	227	6	14
9	I	403	2	18	570	3	17	417	3	17	301	3	17
10	J	279	3	17	455	4	16	399	5	15	193	4	16
	Sum	3411	41	159	5015	37	163	4635	39	161	2558	46	154
	Precision		0.01202			0.007378			0.008414			0.017983	
	Recall		0.257862			0.226994			0.242236			0.298701	
	F		0.022969			0.014291			0.016264			0.033923	

TMF – Total match found **TPR –** Total pages relevant (first 20 links) **NRP –** None relevant pages

DISCUSSION OF RESULTS

The tabulated result of table 1 is an experimental measure of the comparative average results (in thousands) of ten categories of queries (10 queries in each category) administered to the four search engines(Vivosimo, Google, wikipedia and Our System) considering the total match found (TMF), the Total relevant pages (TRP) and the non-relevant pages(NRP) . These results were based on a TREC collection of known relevance for a given query type. The choice of the search engines is established by their diversity in purpose and usage thus the measures would be unbiased. By table 1, the performance shows a clear increase in the number of relevant pages clustered by our system than that of others.

In table 2 we went further to calculate the precision and recall of the systems. It is an experimental measure of the comparative average Recall, Precision and F-measure of ten categories of queries (10 queries in each category) administered to the four search engines(Vivosimo, Google, wikipedia and Our System) considering the total match found (TMF), the Total relevant pages (TRP) and the non-relevant pages(NRP). The result indicates an improvement in the precision, recall and f-measure of the system as captured below:

	Vivosimo	**Google**	**Wikipedia**	**Our system**
Precision	**0.01202**	**0.007378**	**0.008414**	**0.017983**
Recall	**0.257862**	**0.226994**	**0.242236**	**0.298701**
F	**0.022969**	**0.014291**	**0.016264**	**0.033923**

7.0 Conclusion

The biggest problem facing users of web search engines today is the quality of the results they get back. While the results are often amusing and expand users' horizons, they are often frustrating and consume precious time. The most important measure of a search engine is the quality of its search results. Experience with Google has shown it to produce better results than the major commercial search engines for most searches, using PageRank, anchor text, and proximity. The results are clustered by server. This helps considerably when sifting through result sets. Notice that some searched results in Google have no title. This is because it was not crawled. Instead, Google relied on anchor text to determine this was a good answer to the query.

Our work improved the quality of web search engines. Before now, most people believed that a complete search index would make it possible to find anything easily. According to Best of the Web 2000 -- Navigators, "The best navigation service should make it easy to find almost anything on the Web (once all the data is entered)." However, the Web of today is quite different. Anyone who has used a search engine recently can readily testify that the completeness of the index is not the only factor in the quality of search results. These we have not only proved but have proffer a classification process to match similar authoritative pages together with a **pav**. Our system with the F-measure of 0.033923 against that of Vivosimo at 0.022969, Google at 0.014291 and wikipedia at 0.016291 shows a better performance. There exist similar improvements in the recall and precision as indicated.

REFERENCES

Bezdek J. C. (1981): "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York

Castillo C, Donato D, Gionis A, Murdock V, and Silvestri F. (*2007*) incorporated web **...** V.,classification (e.g., (GyÃngyi and Garcia-Molina 2005b; *Castillo*, *Donato*, *Gionis*, *Murdock*, and. Silvestri *2007*)) and so on. This survey focuses on subject

Chakrabarti, S. (2000, January). Data mining for hypertext: a tutorial survey. SIGKDD Explorations Newsletter 1 (2), 1–11.

Chen H. and S. T. *Dumais* (*2000*). Bringing order to the web: Automatically categorizing search results. (pdf file) In Proceedings of the ACM SIGCH1 Conference on Human: Proceedings of CHI'00,

Chekuri, C., M. Goldwasser, P. Raghavan, and E. Upfal (1997). Web search using automated classification. In Proceedings of the Sixth International World Wide Web Conference, Santa Clara, CA. Poster POS725.

Cho, J., Garcia-Molina, H., and Page, L. (1998). "Efficient crawling through URL ordering". *In Proceedings of the seventh conference on World Wide Web* (Brisbane, Australia).

Fiirnkranz, J. (2005). Web mining. In Maimon and L. Rokach (Eds.), The Data Mining and Knowledge Discovery Handbook, pp. 899–920. Berlin: Springer.

Getoor *&* Diehl, (*2005*) Lise *Getoor*, Christopher P. *Diehl*. **...** Indrajit Bhattacharya , Lise *Getoor*, Iterative record linkage for cleaning and integration,

Gyongyi*,* Zoltan and *Garcia-Molina*, Hector (*2005*) Link Spam Alliances. In: 31st International Conference on Very Large Data Bases (VLDB *2005*),

Haveliwala, T. H. (2002). Topic-sensitive PageRank. In Proceedings of the Eleventh International World Wide Web Conference, New York, NY, pp. 517–526. ACM Press.

Kohlschutter, C., P.-A. Chirita, and W. Nejdl (2007). Utility analysis for topically biased PageRank. In WWW '07: Proceedings of the 16th International Conference on World Wide Web, New York, NY, pp. 1211–1212. ACM Press.

Mladenic. D. (1999). "Text Learning and Related Intelligent Agents: A Survey." IEEE Intelligent Systems 14(4): 44-54. This paper proposes five variants of the odd ratio algorithm and compares other algorithm

Mitchell, T. M. (1997). Machine Learning. New York: McGraw-Hill.

Nie, L., B. D. Davison, and X. Qi (2006). Topical link analysis for web search. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval, New York, NY, pp. 91–98. ACM Press.

zu Eissen and Stein,. (2004) [24]; Lim et al., 2005 [17]), and from qualitative analyses (e.g. Shepherd and Watters. Web Agent Design Based on Computational Memory and Brain Research.