Meta-Search Utilizing Evolutionary Recommendation: A Web Search Architecture Proposal

Dušan Húsek
Institute of Computer Science of Academy of Sciences of the Czech Republic, CZ 18207 Prague, Czech Republic
dusan@cs.cas.cz

Amir Keyhanipour, Behzad Moshiri
Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, Iran
a.keyhanipour@ieee.org, moshiri@ut.ac.ir

Suhail S. J. Owais
Information Technology
Al-Balqa' Applied University - Ajloun University College, P.O. Box 6, JO 26810
Ajloun, Jordan
suhailowais@yahoo.com

Václav Snášel , Pavel Krömer, Jan Platoš
FEECS, Dept. of Computer Science
VŠB – Technical University of Ostrava
17. listopadu 15,CZ 708 33
Ostrava–Poruba, Czech Republic
{pavel.kromer.fei, vaclav.snasel}@vsb.cz

Abstract

An innovative meta-search engine named WebFusion has been presented. The search system learns the expertness of every particular underlying search engine in a certain category based on the users' preferences according to an analysis of click-through behavior. In addition, an intelligent re-ranking method based on ordered weighted averaging (OWA) was introduced. The re-ranking method was used to fuse the results' scores of the underlying search engines. Independently, a progressive application of evolutionary computing to optimize Boolean search queries in crisp and fuzzy information retrieval systems was investigated, evaluated in laboratory environment and presented. In this paper we propose an incorporation of these two innovative recent methods founding an advanced Internet search application.

1. Motivation

WWW consists of more than ten billion publicly visible web documents [1] distributed on millions of servers world-wide. It is a fast growing and continuously changing dynamic environment. Individual general-purpose search engines providing consensual search services have been unable to keep up with this growth. The coverage of the Web by each

! !

of the major search engines has been steadily decreasing despite their effort to comprehend larger porting of web space. Several investigations show that no single standalone search engine has complete coverage and it is unlikely any single web search engine ever will [2]. Rather insufficient coverage of the web by standalone general search engines forced research into the area of meta-search systems as tools sending user queries to multiple search engines and combining the results in order to improve search effectiveness [2, 3]. Therefore, a meta-search engine can be described *as an interface on top of multiple local search engines providing uniform access to many local search engines.*

Moreover, unified consensual approach to search requirements of all users becomes with growing amount of data and documents on the WWW inefficient in satisfying the needs of large number of individuals desiring to retrieve particular information. Personalized approach to the needs of each user is general trend in state-of-the-art web applications including search engines. Personalization, based on stored knowledge of users' general needs, area(s) of interest, usual behavior, long and short term context and search practices can be evaluated when improving web search applications, no matter if they are standalone search engines or more advanced meta-search systems lying on the top of individual search applications.

This paper proposes an incorporation of two recent methods for improving web search – meta-search based on implicit evaluation of local search engines' expertness and evolutionary query optimization based on user profiles.

2. WebFusion

Each search engine covers a specific region on the Web. In this regard, meta-search engines use few simultaneous search engines at once to embrace larger parts of the search space. Different techniques of information retrieval which are used in underlying search systems have caused different efficiency of various search engines in particular expertise areas [4]. Majority of contemporary meta-search programs send the query to all of the underlying search engines in their knowledge base which may cause some problems such as resource consumption. The problem is also that each underlying search engine returns too many Web pages, which takes much time for meta-search engine to merge these returned lists and to determine the relevant documents.

Several ranking mechanisms, often based on probabilistic contents, are used to provide search results in a relevant order according to the users' queries [5]. An important drawback of most of these techniques is that they do not consider the users' preferences. Therefore, the relation between the outcomes and the preferences of the users could not be completely matched [6]. An evidence of user preferences can be found in query logs, created during web search and browsing activities.

There were several investigations concerning query log analysis for search improvement. Joachims [7] proposed a method of utilizing click-through data in learning of a retrieval function. He introduced a novel method for training a retrieval function on the basis of click-through data called Ranking SVM.

Among the more recent methods, QueryFind [8] was based on users' feedbacks with respect to the underlying search engines' recommendations. They provide more relevant results with the higher rank in the results list. Another new approach is based on exploiting the filtering capabilities of search-engines and the generalized use of weights and aggregation operators to rank documents [9]. These methods do not consider the search engines expertness in a certain category, which is promising when different sources of knowledge with different coverage are used.

2.1. Click-through data

Users' searching behaviors include queries, click through the list of the returned results, Web pages' content information and browsing activities captured in query logs. These query logs contain rich information which can be used to analyze users' behaviors and improve the results quality [7, 8]. Formally, click-through data in search engines can be seen as triples (q, r, c) consisting of the query q, the ranking r presented to the user, and the set c of links that the user has clicked. Clearly, users do not randomly click on links, but make an (somewhat) informed choice. While Click-through data is typically noisy and clicks are not "perfect" relevance judgments, the clicks are likely to convey some information [7]. Most users click on rather relevant results and we should benefit from a large quantity of query logs. Experiments show that about 82% of the queries are in fact related to the topics of the clicked Web pages [10].

2.2. OWA

The Ordered Weighted Averaging (OWA) operators [10] provide the means for aggregating scores associated with the satisfaction of multiple criteria, which unifies in one operator the conjunctive and disjunctive behavior. The OWA operator of dimension n could be described as: $F : R^n \rightarrow R$:

$$OWA(x_1, x_2, \ldots x_n) = \sum_{j=1}^{n} w_j x_{\sigma(j)} \tag{1}$$

Where σ is a permutation ordering elements $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \ldots x_{\sigma(n)}$. The weights are all non-negative and their sum is equal to one. The OWA operators can be seen as a parameterized way to go from the min to the max. In this context, a degree of maxness (initially called orness) was introduced in [11], defined by:

$$Maxness(w_1, w_2, \ldots w_n) = \frac{1}{n-1} \sum_{j=1}^{n} w_j (n-j) \tag{2}$$

For the minimum, we have $maxness(1,0,\ldots,0)=0$ and for the maximum $maxness(0,\ldots,0,1)=1$.

2.3. Search engine expertness modeling and result re-ranking

To each underlying search engine, an expertness value $expt^c_i$ has been assigned. The value illustrates the expertness of the search engine i in the category c. The expertness value is updated by bootstrapping technique based on previous expertness value and the current usage behavior of the user [12]. In this way, search engines in which the higher ranked results are clicked earlier, are more rewarded than the others. The expertness updating formula and the reward function are given as follows:

$$expt_i^c = (1-\alpha)expt_i^c + \alpha \cdot \gamma_l$$
$$\gamma_l = \frac{\sum_{i \in hitted\_results} (N-i) \cdot t_i}{\sum_{i=1}^{N} (N-i) \cdot i} \tag{3}$$

Where N is the number of returned results, i is the index of a result in the ranked list and t_i is the index of hit result i. The learning rate α is also modified by the following formula:

$$\alpha = \exp(\beta \cdot t) \tag{4}$$

Where t is the iteration number and β is the regulator of the learning rate that was in performed experiments fixed to 0.2. Each iteration starts by query submitting and it will be finished after the user selects the results and closes the session. The learning rate allows at the beginning of learning process more exploration than exploitation and the exploration-exploitation is decreasing in time.

The returned results of the underlying search engines are taken as their decisions for the submitted query which are fused in the decision level of information fusion. In decision level, the more weights are assigned to the higher results of the more expert search engines for the dispatched query in the corresponding category.

For every item, ranks that are assigned by the underlying search engines are assumed as the base of OWA method. For each result item j from the search engine i which is classified in category c, a weight is computed:

$$w(i,j,c) = expt_i^c \cdot \left(1 - \frac{j}{N}\right) \tag{5}$$

For each result item, search engines are sorted decreasingly based on these weights. Then OWA weights are assigned to these search engines according to the equation (3). Final weight of each result item is calculated:

$$w_F(j) = \sum_{i=1}^{N} w(i,j,c) \cdot w_{OWA}(i) \tag{6}$$

After assigning the weights to each returned result, they are sorted according to their weights decreasingly and the final ordered list is generated.

2.4. Evaluation

An experimental meta-search environment named WebFusion [13] was implemented in order to evaluate the effectiveness of the proposed methods. We have selected 280 sample queries in the "Computers" category. These queries were collected using a proxy application mounted on the server of the "Instrumentation and Industrial Control Lab." of the ECE department of the University of Tehran. The categories presented by Open Directory Project (http://dmoz.org/), the largest, most comprehensive human-edited directory of the Web, were used to categorize the queries sent by the users. In our experiments users were asked to manually select the category of their queries. Experiments were shown in the category of "Computers".

The performance of the WebFusion is measured by factors such as average click rate, total relevancy of the returned results and the variance of the clicked results. The first measure which is the average position of the clicked results indicates that wither interesting results are settled at the top of the ranked list. If average click rate is relatively little, it shows that users can find useful results at the first portion of the ranked list while saving time and energy. The second criterion, measures the relevancy of the content of the returned results according to judgment of the users. This criterion can be considered as the indicator of nature of responses. The last factor shows the long term performance of the meta-search engine which is determined by the behavior of the users. Consideration of these criteria together, can provide a good insight about the meta-search engine. For calculating the relevancy of the results, we have extended the relevancy measure proposed in [14] into the following:

$$rel = \frac{2 \cdot \mathrm{Rel} + \mathrm{Undecided}}{2 \cdot \mathrm{Ret}} \tag{7}$$

Sample search queries were executed on WebFusion. The results are supposed to be classified to relevant, undecided, and irrelevant documents. Each experiment participant was asked to evaluate the result items in the corresponding classes.

The comparison of WebFusion to the ProFusion (http://www.profusion.com) and some other underlying search engines shows an obvious enhancement. Average relevancy of WebFusion results is 85.5%, while this is about 76% for ProFusion. WebFusion has also better performance than the MetaCrawler (http://www.metacrawler.com). It is noticed that the average relevancy of WebFusion is 15.5% better than the best underlying search engine in the sense of relevancy. The learning process of WebFusion along the iterations can be spotted by the means of average click rate.

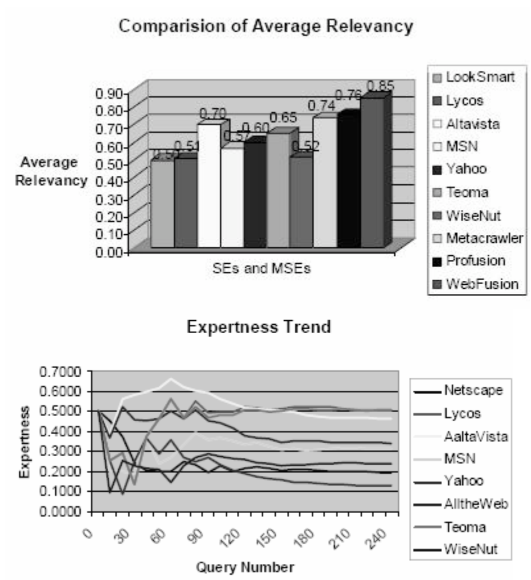

Figure 1. Average relevancy and expertness learning results

Average click rate has decreased from 11.77 to below 6.23 in 240 queries, while this is 9.701 for the ProFusion. The proposed method leads to a mapping between query categories and the underlying search engines of the WebFusion. In other words, the most expert search engines of each category are identified. Figure 1 summarizes the comparison of WebFusion's average relevancy and captures the expertness learning process for the category "Computers" along the iterations.

Concluding from above presented experiments, this approach can provide more relevant Web pages in higher ranks and hence reduce users' time and tension when finding useful information within the retrieved result sets.

3. Personalized Evolutionary Query Optimization

An individual user profile (IUP), containing stored knowledge about system user, could be utilized to improve search results by the means of personalization. Search engine equipped with user profiling can exploit user-specific acquirements to retrieve documents satisfying search queries with respect to individual user, her or his general needs, preferences, abilities, history, knowledge and current context.

Explicit profiles, defined by users themselves, are rather imprecise and not enough flexible. Instead, various techniques for implicit creation and maintenance of user profiles are being investigated [15]. In this section, an evolutionary approach exploiting information from user profiles to optimize search queries by artificial evolution will be discussed.

3.1. Evolutionary Techniques

Evolutionary Algorithms (EA) are *family of stochastic search and optimization methods based on mimicking successful strategies observed in nature* [16]. EAs emulate the principles of Darwinian evolution and Mendelian inheritance for the use in computer science.

Artificial evolution operates over a *population* of individuals (chromosomes) encoding possible solutions by applying so called *genetic operators* (selection, crossover, mutation) [16]. The individuals are evaluated using objective function assigning a *fitness* value to each individual. Fitness value mirrors the ranking of each individual as a solution of the problem. Competing individuals intensively search problems solution space towards optimal solution in more directions simultaneously [16].

EAs are well proven general adaptable concept with good results. The family of evolutionary algorithms consists of *Genetic Algorithms (GA), Evolutionary strategies (ES) and Evolutionary programming (EP)*.

Genetic Algorithms were introduced by Holland when exploring the possibilities of computer simulated evolution [16]. They are widely applied and highly successful EA variant. GAs were designed as a general model of adaptive processes and found most of its applications in the domain of optimization. Basic scheme of original *generational GA* is:

① Encode initial population of possible solutions and evaluate chromosomes (assign fitness value)

② Create new population (evolutionary search for better solutions):

   a. Select *suitable chromosomes for reproduction (parents)*
   b. *Apply* crossover *operator on parents with respect to crossover probability to produce new chromosomes (offspring)*
   c. *Apply* mutation *operator on offspring chromosomes with respect to mutation probability. Add newly constituted chromosomes to new population*

d. *Until the size of new population is smaller than size of current population go back to* a.

e. Replace *current population with new population*

③ Evaluate current population; assign fitness values to chromosomes

④ Check termination criteria; if not satisfied go back to ②

Termination criteria, crossover probability, mutation probability, population size, maximum number of processed generations and migration strategy are among the most important parameters of each GA based solution.

A high-level variant of GAs, *Genetic programming (GP)*, attracts attention. GP uses hierarchical rather than linear chromosomes and thus is able to evolve computer programs or other hierarchically structured entities such as search queries. Genetic operators are usually applied on nodes of tree chromosomes encoding hierarchical individuals. GP has a good ability to produce symbolic output in response to symbolic input [16].

3.2. Query optimization by genetic programming

More works presenting the use of evolutionary optimization in search and information retrieval have been presented. Kraft et al. [4] published GP method to optimize user search queries. Ramos and Abraham [17] published hybrid method for user profile improvement combining stigmeric paradigms and genetic programming.

In our work, Boolean queries were experimentally evolved over a sample document collection with several setups of GP parameters. Effectiveness of IR system was measured through values indicating the ability of IR system to satisfy users' information needs [2]. Among IR effectiveness measures, *precision* (*P*) and *recall* (*R*) as the most used IR performance measures [2] were applied. *P* and *R* are defined as:

$$P = \frac{|\text{Rel} \cap \text{RelRet}|}{\text{Ret}} \quad R = \frac{|\text{Rel} \cap \text{RelRet}|}{\text{Rel}}$$
$$F = \frac{2PR}{P+R} \tag{8}$$

Rel stands for a set of all relevant documents in the collection, Ret is set of all retrieved documents and RelRet is set of retrieved relevant documents. Precision describes the exactness of result set retrieved in response to user request, the ratio of number of relevant retrieved documents to number of all retrieved documents and recall describes the completeness of result set, the ratio of the number of relevant retrieved documents to the number of all relevant documents. Harmonic mean of precision and recall called F-score (*F*) was used for assembling precision and recall in one scalar value [2]. In our experiments, all three mentioned fitness measures were used.

We considered user models containing Boolean search queries representing ones individual search needs, long and short term context and document relevance estimation [18]. Several experiments were executed using data extracted from the LISA collection (http://www.dcs.gla.ac.uk/idom/ir_resources/test_collections/). The collection was indexed for Boolean IR and Extended Bolean IR systems, using Salton's indexing function based on normalized term frequency and normalized inverse document term frequency [19] in the latter case. Indexed collection contained 5999 documents and 18442 unique indexed terms. Due to stochastic character of GP process, all experiments were executed multiple times and mean experimental results evaluated.

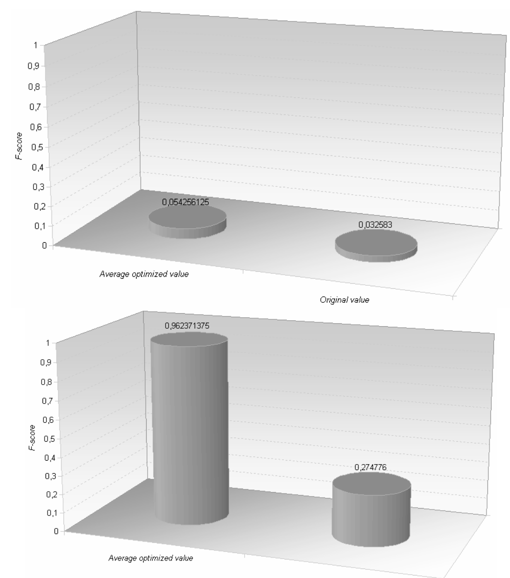

Figure 2. Query optimization improvement for random initial population and seeded initial population

Several GP parameters were initialized for all experiments and the result of query evolution in cases with different (random initial population and initial population seeded with better ranked individuals) initial populations was observed. The experiments were done using both, standard Boolean queries and fuzzified extended Boolean queries.

In the set of experiments with standard Boolean queries and seeded initial population, the average F-score value increased from initial 0,27 to optimized 0,99 (while 1 was maximum). Random initial population started with the average F-score value 0,04 and the F-score value after optimization reached 0,07. Experiments with fuzzified Boolean queries resulted into following: for seeded initial population, the F-score value boosted from average 0,27 to 0,98. Random initial population of average opening F-score 0,03 was optimized up to 0,07. An example illustrating experimental results is shown in figure 2.

The experiments have shown that GP can be used for optimizing user queries in information retrieval systems. Crucial for successful optimization was the quality of initial population. For successful optimization, initial population must contain at least some quality queries pointing to documents related to users needs. F-score fitness was preferred as a measure combining precision and recall into one value by the means of information retrieval and therefore simplifying query optimization from multi-objective to single-objective task.

4. Innovative Web Search Architecture Proposal

An advanced web search application, preventing some of imperfections summarized in first section of this paper, could incorporate both above presented novel techniques. The personalized meta-search enabled system will consist of:

- Implicit user profiles (IUP) based on click-through data created and updated by analysis of user behavior observed via click-through data. Additionally, profiles will contain previous user queries. User profiles will be exploited when optimizing search queries.
- Implicit search engine expertness profiles (SEEP) created and updated by analysis of users' click-through behavior. Search engine profiles will be exploited when retrieving and re-ranking results from underlying local search engines.
- Meta-search and re-ranking subsystem interacting with local search engines and providing data for recommendation component.
- Personalized recommendation subsystem based on queries from user profiles optimized by genetic programming and document relevance estimation. The component will offer alternative queries, created with respect to current user query, previous search behavior and personal relevancy information stored in user profiles.

Introduced architecture, consisting of independent but complementary advanced components, could notably increase search comfort and decrease the time needed for user interaction with web search applications thus improving the efficiency of the search task. Meta-search environment, utilizing and extending services provided by contemporary search engines and featuring personalized recommendation system build upon consensual systems is supposed to offer high added value at low cost.

Implicit observation of user click-through behavior is used to provide data basis for building both, SEEPs and IUPs. The simultaneous exploitation of search engine expertness profiles and individual user profiles will be deployed to improve the result sets retrieved in response to user queries. Results will be re-ranked with respect to detected search engine expertness, preferring for particular category results retrieved by search engines with higher expertness. User profiles will be employed to be a basis for personalized query optimization, performed on the top of retrieved and according to SEEP re-ranked result set. Neither the SEEPs nor IUPs are needed to be specified explicitly. On the contrary, meta-information provided by ordinary user tasks performed during web search activities and hence almost omnipresent (and in majority of search environments omitted), will be extracted and reasonably exploited to discover SEEPs and UIPs. Hereby created and maintained search engine expertness profiles and user profiles do not require any additional feedback from the users, although an explicit feedback should be considered as a mean to speed up the expertness and user learning process and make the profiles more literal.

A search environment implementation, built according to proposed architectural concepts, will offer the advantages of expertness and user aware advanced search techniques on the top of contemporary systems allowing the exploitation of their today's information retrieval capabilities for more sophisticated search improvement. Our future work should lead to prototype implementation of proposed architecture and experimental confirmation of proposed concepts. Major attention will be paid to the particular methods for creating click-through based implicit SEEPs and IUPs.

5. References

[1] A. Gulli and A. Signorini, "The indexable web is more than 11.5 billion pages," in *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, (New York, NY, USA), pp. 902–903, ACM Press, 2005.

[2] R. M. Losee, "When information retrieval measures agree about the relative quality of document rankings," *Journal of the American Society of Information Science*, vol. 51, no. 9, pp. pp. 834–840, 2000.

[3] M. Gordon and P. Pathak, "Finding information on the world wide web: the retrieval effectiveness of search engines," *Inf. Process. Manage.*, vol. 35, no. 2, pp. 141–180, 1999.

[4] D. H. Kraft, F. E. Petry, B. P. Buckles, and T. Sadasivan, "Genetic Algorithms for Query Optimization in Information Retrieval: Relevance Feedback," in *Genetic Algorithms and Fuzzy Logic Systems* (E. Sanchez, T. Shibata, and L. Zadeh, eds.), (Singapore), World Scientific, 1997.

[5] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998.

[6] B. Amento, L. Terveen, and W. Hill, "Does authority mean quality? predicting expert quality ratings of web documents," in *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 296–303, ACM Press, 2000.

[7] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, ACM, 2002.

[8] G.-R. Xue, H.-J. Zeng, Z. Chen, Y. Yu, W.-Y. Ma, W. Xi, and W. Fan, "Optimizing web search using web click-through data," in *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, (New York, NY, USA), pp. 118–126, ACM Press, 2004.

[9] M. Gomez and J. C. Abasolo, "Improving meta-search by using queryweighting and numerical aggregation operators," in *Proceedings of the Information Processing and Management of Uncertainty conference, IPMU 2002*, 2002.

[10] R. R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking," in *Readings in Fuzzy Sets for Intelligent Systems* (D. Dubois, H. Prade, and R. R. Yager, eds.), pp. 80–87, San Mateo, CA: Kaufmann, 1993.

[11] D. Zhang and Y. Dong, "An efficient algorithm to rank Web resources," *Computer Networks (Amsterdam, Netherlands: 1999)*, vol. 33, no. 1–6, pp. 449–455, 2000.

[12] P.-H. Wang, J.-Y. Wang, and H.-M. Lee, "Queryfind: Search ranking based on users' feedback and expert's agreement," in *EEE '04: Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)*, (Washington, DC, USA), pp. 299–304, IEEE Computer Society, 2004.

[13] A. H. Keyhanipour, B. Moshiri, M. Piroozmand, and C. Lucas, "Webfusion: Fundamentals and principals of a novel meta search engine," in *Proceedings of International Joint Conference on Neural Networks*, 2006.

[14] S. Gauch and G. Wang, "Information Fusion with ProFusion," in *Proceedings of WebNet '96: The First World Conference of the Web Society*, (San Francisco, CA), pp. 174–179, Oct. 1996.

[15] O. Cordón, F. de Moya, and C. Zarco, "Fuzzy logic and multiobjective evolutionary algorithms as soft computing tools for persistent query learning in text retrieval environments," in *IEEE International Conference on Fuzzy Systems 2004*, (Budapest, Hungary), pp. 571–576, 2004.

[16] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1996.

[17] V. Ramos and A. Abraham, "Evolving a stigmergic self-organized data-mining," *CoRR*, vol. cs.AI/0403001, 2004.

[18] D. Húsek, V. Snásel, R. Neruda, S. S. J. Owais, and P. Krömer, "Boolean queries optimization by genetic programming," *WSEAS Transactions on Information Science and Applications*, vol. 3, no. 1, pp. 15–20, 2006.

[19] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. pp. 513–523, 1988.