IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

GOOGLE SEARCH ENGINE PREDICTION

NAZEERA A MADABHAVI
Assistant Professor,
Dept of Computer Science & Engineering
SECAB Institute of Engineering & Technology
Vijayapura, Karnataka, India.

MISBA BANU JAKATI
UG Scholar,
Dept of Computer Science & Engineering
SECAB Institute of Engineering & Technology
Vijayapura, Karnataka, India.

ANTHYAKSHARI BALABATTI
UG Scholar,
Dept of Computer Science & Engineering
SECAB Institute of Engineering & Technology
Vijayapura, Karnataka, India.

PRIYANKA BIRADAR
UG Scholar,
Dept of Computer Science & Engineering
SECAB Institute of Engineering & Technology
Vijayapura, Karnataka, India.

Abstract: In this day and age of abundant information, search engines are essential for finding pertinent stuff. As the top search engine, Google works hard to improve user experience by correctly interpreting and displaying search results that are relevant to the user's purpose. A summary of the developments in Google's search engine prediction methods is provided in this document. We explore the fundamental approaches used to efficiently predict user queries, such as artificial intelligence techniques, natural language processing strategies, and user behavior analysis. We also talk about the difficulties in correctly interpreting user intent, including managing unclear inquiries, comprehending context, and adjusting to changing search trends. Furthermore, we talk about potential future research fields that aim to increase prediction accuracy and improve consumers' overall search experience. By realising.

Index Terms: Document clustering, natural language processing and machine learning, personalized topic. search, user log.

I. INTRODUCTION

Including Long Short-Term Memory (LSTM) algorithms has revolutionized Google's search engine, which is renowned for its intricate architecture. Recurrent neural networks (RNNs) such as Long Short-Term Memory (LSTM) have garnered accolades for processing sequential input with exceptional memory retention. The project intends to improve user experience and provide more precise and tailored results by revolutionizing the understanding of search queries and ranking results. Given the volume of information available, knowing how to effectively search for pertinent information and react to user queries promptly becomes crucial. Using search engines to assist individuals in finding the data they need quickly is an efficient method.

A. RESEARCH QUESTIONS FOR THIS STUDY

Nevertheless, search engines encounter two issues: (a) an excessive amount of material is returned [2], and (b) users frequently type brief queries [3], [4]. The first issue is that, for typical user searches, there are millions to hundreds of thousands of results that the search engine
will return. [5]. The user usually isn't most of the returning results piqued my interest. Because there are so many of them. The percentage of users viewing the first ten search results, or the search results page's first page. is 58%, and the page views of the first three pages reach 86%, based on the pertinent statistical data [6]. Stated differently, the top three search results pages are all that the majority of people are interested in. The queries submitted by the users pose the second issue.

B. RESEARCH IDEAS FOR THIS STUDY

In this work, we provide a solution to these two problems. To address the issue of the system processing data, the problem of returning too much data set will initially focus on the front search results. Statistics show that most users (about 86%) are only interested in the first three search results pages [6]. This indicates that most users scan through the first 30 search results only. Focusing on these small datasets with high search scores allows the system to respond to user requests faster.

C. RESEARCH OBJECTIVES FOR THIS STUDY

Traditionally, user log files—such as clicking and browsing histories—have been used to generate search results for specific users in personalized search services like Google History and My Yahoo. There are two main problems with this strategy: Two things: (1) personal seclusion; and (2) ample storage. Regarding the first issue, all of the user's personal browsing history is stored in the user log file because the personalized search needs users to check in to their account. The incorporation of LSTM algorithms into Google's search engine represents a significant breakthrough in natural language processing
and information retrieval. Google hopes to revolutionize the search experience by utilizing deep learning, giving people access to
information with never-before-seen efficiency and precision. As Google embraces developing AI and improves its algorithms

II. RESEARCH METHOD:

In this section, we go over the design techniques and mathematical models that we employed in this section's experimental system. The research flow for this project is shown in Figure 2. The user starts by asking a question of our system. Relevant search engine snippets are displayed in the "Query Process" column according to the user's query. Subsequently, the gathered snippets are cleaned up by the "SERP Process" using several NLP approaches. Additionally, it reorganizes all of the gathered snippets using an evaluation algorithm that is dependent on the user's surfing habits. Next, the "Topic Process" generates subjects linked to the question by first using the N-gram statistical language model. The procedure then groups related subjects into a topic tree for the user using the notion of sets in mathematics.

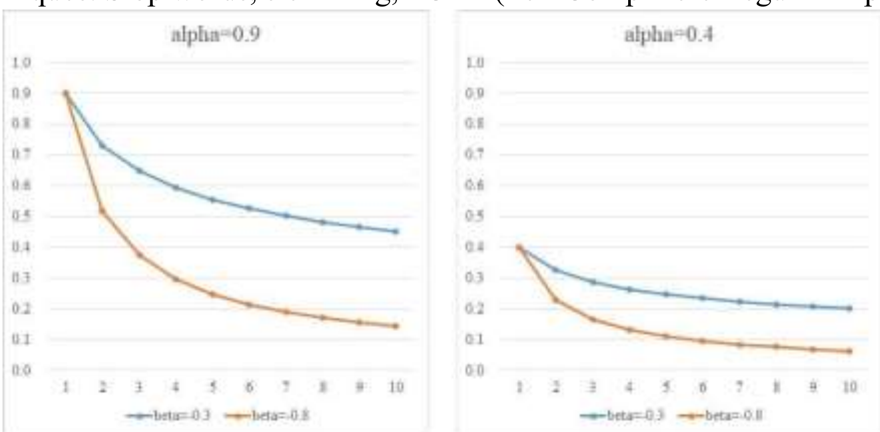
The meta-search technique [90] for general search engine post-processing acts as the basis for our investigation. The process primarily requests general search engines on behalf of users and compiles the results that these engines provide [91]. This approach offers the following benefits: (a) broadening the search's scope [92], (b) reducing enterprise build expenses [93], and (c) balancing the views of various search engine results [94].

Table1.A comparison of different types of recommendation systems

Characteristic	Result	Practice	Advantage	Disadvantage
Keyword suggestion	Related keywords	Proximity search	1. Easy to implement 2. Hot trend keywords	1. Suggested keywords lack semantics 2. A huge storage space
	Researcher			
	Renjie et al. [73]	They proposed a video clustering method based on the referrer video graph to obtain relevant suggested keywords.		
	Zhou et al. [74]	They proposed a method based on generative neural network to generate diversified and domain-consistent suggested keywords.		
Question answering	Result	Practice	Advantage	Disadvantage
	A clear answer	NLP	1. A precise answer 2. High accuracy for some areas	1. Difficult to apply to open domain questions 2. Open domain questions are less accurate
	Researcher	Idea		
	Brill et al. [84]	They proposed a system that uses different N-gram techniques to check the dependency of data redundancy.		
	Abacha and Zweigenbaum [76]	They provided more expressive, standard formal language and make corpus and question annotations shareable		
	Bhandwaldar and Zadrozny [85]	They answered facts and lists questions based on the method of named entities and the summary of the constructed paragraph size based on the lexical chain.		
Topic searching	Result	Practice	Advantage	Disadvantage
	Related search topics	Document clustering	1. It reduces user judgment time 2. Generate topic quickly	1. It cannot cover all the topics of the page 2. Different methods are difficult to compare
	Researcher	Idea		
	Scaiella et al. [38]	They used a graphical concept to analyze the relationship between topics and documents.		
	Gamare and Patil [55]	They used HAC and link-based algorithms to cluster the documents		

A. SERP PROCESS

The outcome of the search produced using the search engine is known as the snippet. It mostly consists of the title, URL, and snippet of text connected to the search engine-processed query [17], [58], [95], and [96]. The three benefits listed below are connected to search engine generated snippets [17], [95]: the essential content of the webpage, (b) saving the user time from having to view the entire website, and (c) the snippet frequently processes faster than the entire page. We must convert the transforming chaotic documents into organized ones for further processing because the search snippets that were returned were in an unstructured manner [97]. In this study, we perform this conversion using the following NLP techniques: Stop words, stemming, PCRE (Perl Compatible Regular Expressions), and



(a) PE distribution when alpha = 0.9 (b) PE distribution when alpha = 0.4

figure1. Correspondingpedistributionatdifferentαe

In general, Typically, a a list of the search results is displayed, and the roster is arranged as decreasingly relevant. In other words, the user's inclination towards the prior search list's results is declining. The trend distributions of user browsing by PE are displayed in Figure 3. PE largely defines its distribution based on two parameters: αe and βe. The user's initial perception of item list e is indicated by parameter αe, whereas their preference decline for the object list e is indicated by parameter βe. The key difference between the two sub-figures in Figure 1 is that subfigure (a) has a rather big αe value. This indicates that the user in subfigure (a) has a more favourable initial impression of the object than the user in subfigure (b).

III. EXPERIMENTAL ANALYSIS AND DISCUSSION

The first thing we cover in this part is how to set the appropriate topic process parameters. The performance and cost of the clustering results produced by the topic process are then examined. Next, we quantify the performance difference between personalized search and typical search using standard measures. Lastly, we go over this study's benefits.

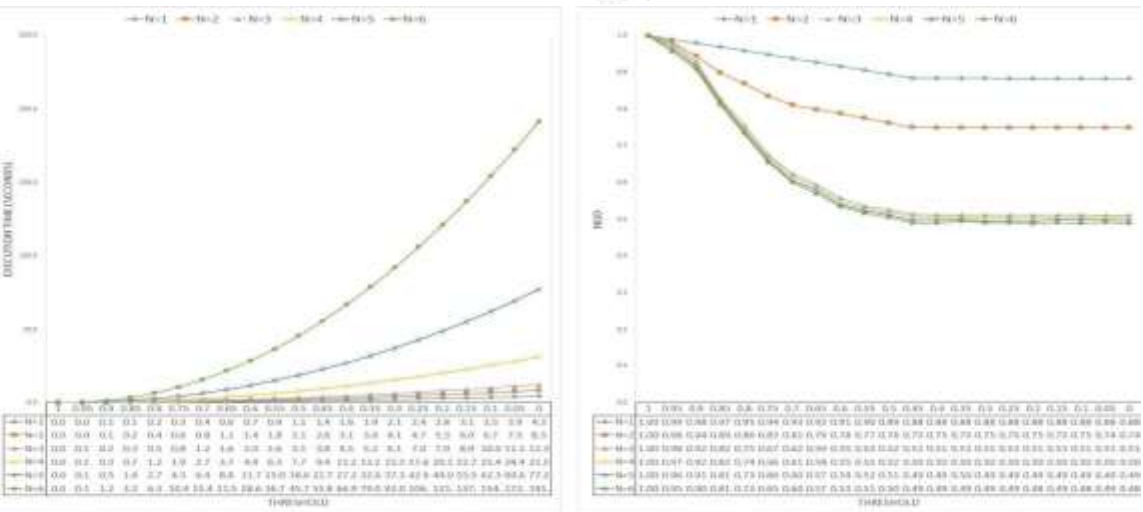
A. DISCUSSION ON RELATED PARAMETERS

Here N and the threshold, the topic's two parameters process, are discussed. Please refer to Section III.B.1 for the parameters' description. We may infer from Table 5's results that N and the threshold need to be appropriately chosen to establish a cost-performance balance. We first go over the training corpus and test data set that were utilized in this trial. We then go over the measure of performance that was applied in this trial. In the end, we contrast N's both cost and performance with the threshold.

1) THE TEST DATASET AND TRAINING CORPUS

An evaluation set of data for one thousand real searches that users would have entered into Google between March 13, 2013, and November 8, 2016, was used in the study. Due to the large number of test data sets, we advise Readers who are interested can view it at http://hlcs.sytes.net/pwsc/1000.pdf. The data set selected will affect the system's performance. The evaluation data set of 1000 queries used in this study is made up of the top 1000 real user queries from Google Trends over 1336 days. The purpose of this study's data set is to assist us in assessing and understanding the real search needs of the majority of users over an extended evaluation period.

2) PERFORMANCE AND COST COMPARISON FOR NAND THRESHOLD



(a) Execution time for different N and threshold values (b) NGD for different N and threshold values

FIGURE 2. Execution time and NGD for different N and threshold values

Figures 2-(a) and 2-(b) correspondingly display the NGD distribution of distinct N as well as the cutoff points, as well as the duration of the execution Figure: For a thousand test queries, the average execution time and average NGD are
represented by each dot. To locate the NGD and average execution time for all, we first determine the query Q's top 10 topics. To find the final average NGD and average execution time. (i.e., each dot in Figures 2-(a) and 2-(b)), we next average the Performance does not improve all that much of all 1000 inquiries. We discovered, by examining Figure 2-(a), that the time of execution is significantly shorter (averaging between 1.5 and 4.2 seconds) when N is less than 3. The user should be able to complete the inquiry within this reasonable wait time. Conversely, the execution time increases significantly (The range of the average execution time is 10.7 to 64.1 seconds) when N is larger than 3. People are unable to wait so long for outcomes. In the event that Ni is less than 3., NGD performance is noticeably reduced (the average NGD ranges from 0.92 to 0.79), as shown in Figure 2-(b). On the other hand, NGD is substantially better (average NGD ranges from 0.59 to 0.57) if N is more than or equal to 3. Additionally, as the chart demonstrates, NGD Performance does not improve appreciably when N exceeds 3. That is to say, the total 8 1 number of sequences g generated will be very huge when we use N greater than 3 to execute the cluster. Along with a notable increase in cost (execution time), this will also result in a negligible improvement in overall performance (NGD). Consequently, N was fixed to 3 in the study. Searching once more for Figure 2-(b) shows that the performance of NGD scarcely rises when the cutoff value is less than.
0.45, regardless of N. Consequently, in this investigation, we put the threshold at 0.45. Figures 2-(a) and 2-(b) illustrate how we may reply to the user's query demand at an average of 3.8 seconds and provide an average NGD of 0.46 for a threshold equal to 0.45 and a N equal to 3. This indicates that within a reasonable timeframe for users, we can produce quality clustering subjects.

B. COMPARISON OF PERFORMANCE AND COST OF CLUSTERING RESULTS

A contrast between various snippet-oriented clustering systems is shown in Table 6. The second section explains the idea of distinct systems. C. The various techniques for creating atopic zones are as follows: This study is based on the N-gram language model and the various operations on the set; Carrot2 and WS Care are based on the STC algorithm, and Tag My Search is the topic graph.Carrot2 and Tag My Search are displayed in a flat structure based on the clustering results, but the WSC and this study are portrayed in a hierarchical structure.Carrot2 and WSC do not offer any binary operations for personal search comparison, while Tag My Search can only do the OR operation on several themes. This study can perform various binary operations for various topics to satisfy users' varying search needs.

TABLE2. Compares on of snippet-oriented systems.

System / Characteristic	Carrot2	WSC	TagMySearch	This study
Method used	STC-based	STC-based	Topic graph	N-gram and set
Flat or Hierarchical	Flat	Hierarchical	Flat	Hierarchical
Provide personalization				
Average NGD				
Average execution time (second)	5.746	5.124	7.432	3.873

TABLE3. The execution time for different numbers of snippets.

#Snippets	10000*	20000	30000	40000	50000	60000	70000	80000	90000	100000
Time	2.73	5.24	8.10	10.92	13.74	15.72	19.33	22.21	23.63	26.73

* For each query, we use the parameter num in Google search to set the maximum number of returned snippets to 100, that is, every 10000 snippets are the search results returned by 100 queries.

TagMy-Search is the least effective (highest NGD) according to Table 2's results. Subject Matter For this reason, it selects multiple of the word patterns that occur most frequently as potential themes by using the frequent item sets idea. It's simple, therefore, to neglect potentially significant topics or choose ones that don't have distinct effects. We employ the set of mathematical formulas idea to integrate disparate themes with identical meanings but distinct combinations into a single subject, which makes our study preferable to STC-based solutions. To clarify, sequences τ1 (X, Y) and τ2 (Y, X) will be combined into sequence after undergoing a set operation in mathematics when they satisfy the requirements in Section III.B.1. By now, every snippet from τ1 and τ2 will be included in the snippet of the combined sequence τ. Also, we employ a multithreaded approach to gather the search engine result result snippets, which makes our study faster than previous systems at execution time. The amount of time required to construct the parent-child hierarchy tree is then covered. Using an Intel Core 2 Duo T9600 processor and 2GB of RAM, we construct an orderly tree in this project. When utilizing our hierarchical tree-generating method, Table 3 compares the execution times needed for varying numbers of snippets. We discovered that the overall time required to create the hierarchical trees is 2.73 and 5.24 seconds, respectively, when the quantity of excerpts is 10,000 and 20,000. These findings are shown in the table. It was also discovered that a total of 26.73 seconds is needed when the number of snippets is 100,000. In other words, each snippet takes roughly 270 microseconds on average to create the tree with layers. This tendency may also be seen in the other excerpt numbers in the table. This indicates that there is a direct correlation between the quantity of small samples. The results of the experiment demonstrate that when the system selects more snippets, binary encoding takes longer to compute. We create an appropriate topic tree by choosing the search engine snippet results that the majority of users may find most palatable so that the system can answer the user's query needs in real-time. Relevant statistics results [6] indicate that 86% of users look through the search engine results page in fewer than thirty snippets. It might, however, produce fewer subjects if the system only chooses search results with 30 or fewer snippets because fewer snippets are chosen. To attain a balance between the calculation time and the number of topics created, our system processes approximately 100 to 150 snippets for every user query. Because the met-a-search method is utilized in the query process to gather the number of snippets from multiple search engines (100 for Google, 50 for Yahoo, and 50 for Bing), the quantity of excerpts returned by each query is not fixed.

C. THE ADVANTAGES OF THIS STUDY

We initially describe the study's contribution before elaborating on how it differs from earlier research when discussing the study's benefits. Lastly, we go over how the binary operations in this study differ from those in the conventional search engine.

1) DISCUSSION ON THE CONTRIBUTION OF THIS STUDY

2) This work has three key contributions, which are as follows: it swiftly generates a hierarchical tree, establishes a multi-cluster relationship, and performs various binary operations.

3) Easily construct a hierarchical tree: The study conducted in Section IV.B showed a straight line connecting the quantity of **4)** snippets and execution time. As was already noted in Section IV.B., timing works best. **5)** Perform a many type of binary

6) operations quickly: We provide several binary operations based on the concept of binary encoding. By employing binary operations among several topics, we can quickly and effectively produce search results tailored to each user. In theory, our binary encoding method can be used to quickly perform any possible binary operation.

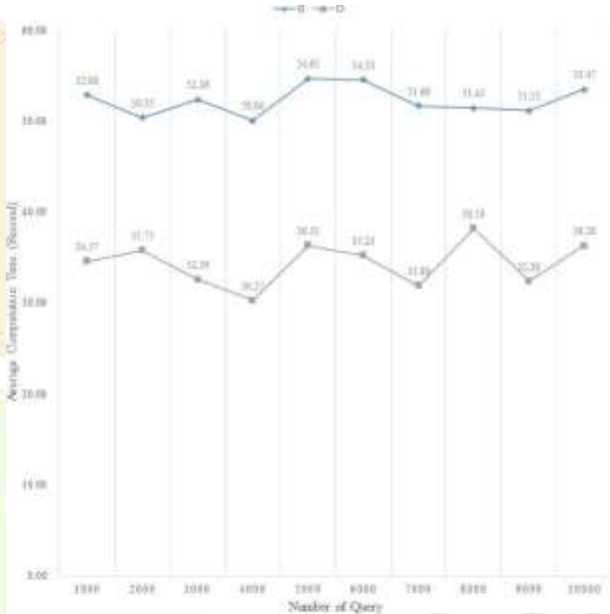
DISCUSSION ON THE DIFFERENCE BETWEEN THIS STUDY AND EXISTING RESEARCH PERSONALIZED SEARCH

Table 1 demonstrates the distinction between this research and the body of current research in personalized look-for. Given that both manage customized searches, Google History and My Yahoo handle user log files; nonetheless, they all have significant storage and privacy problems. However, concerns about personal privacy and a lot of storage space can be avoided by Tag My Search, WSC, Carrot2, and this study since they all achieve personalized search without storing user log files. Tag My Lookup and Carrot2 exclusively offer flat topic displays They oppose hierarchical topics. presentations when it comes to customizable subjects. Conversely, though, hierarchical topic

presentation is supported by this study and WSC. Relevant research demonstrates that unclear or poor-quality inquiries benefit greatly from hierarchical subject presentation. Based on the experimental outcomes, we discovered that Tag My Search performs the poorest in terms of computation time and non-storage technique. Conversely, however, this study's computing time just requires linear time and optimal performance. Except for this study's customizable search options and Tag My Search, the customization choices do not accommodate other systems. Comparing the customization possibilities offered by this study covers all possible binary operations for these two systems. While Tag My Search just offers the OR option.

7) DISCUSSION ON THE DIFFERENCE BETWEEN THIS STUDY AND THE TRADITIONAL SEARCH ENGINEINBINARYOPERATIONS

For binary operations, the following are the differences between this study and an extensive search engine such as Google or Bing: Collaborating with the data set: The main search engine prioritizes user queries' pertinent websites or their binary operations. With this method, the user must specify query terms for the appropriate binary operations. Since the average query length entered by the user is short, such work is challenging for the user [6]. This study concentrates on the binary operations of associated subjects within the query. The benefit of these four approaches is that our system offers extremely relevant topics to queries automatically. Users' thought times



for relevant questions or topics can be significantly shortened as a result. Processing speed: The primary reason for the general search engine's delay is that, when a query involves many binary operations, it must re-index the database to produce appropriate search results. The computer power of the device has a direct effect on the re-indexing time. Rebuilding the hierarchical topic tree while employing binary operations takes up the most of the study's duration. Building the hierarchical tree in this study takes between 0.0270 and 0.0405 seconds per query, according to the experimental data in Section IV.B. Afterwards, we experimented to examine the time difference between rebuilding the hierarchical topic tree for this study and re-indexing the database for the general search engine. This experiment used a single computer (Intel Core 2 Duo T9600 with 2GB RAM) to run two separate search mechanisms. Page Rank, which is widely recognized in the search engine industry, is the primary re-indexing algorithm used by the general search engine [115]. The line with the letter G in Figure 7 represents a comparison of the computation times for various search engines. As per the findings of this experiment, the average computation time of four studies is superior to that of the general search engine. This experiment demonstrates how the hierarchical topic tree approach suggested in this paper can outperform the generic search engine in replying to the user's query criteria. Conserving space: The general search engine must keep the related index files and cached pages for the gathered Web pages. The number of cached pages and the number of index fields are the two factors that determine the index file's size. However, as there is no indexing action, this study does not require the storage of any index files. Because of the distinctions between points 2 and 3, the general corporation lacks the financial resources necessary to develop a matching search system. But thanks to this study's design processes, all we need to set up a functional search system is a general-purpose personal computer.

8) DISCUSSION ON THE VARIATIONS AMID THIS RESEARCH AND THE COST-EFFECTIVE METHOD

This investigation and the cost-effective G A method vary in three key ways [58]: Table 9 displays the solution object, implementation detail, and computation time. Resolve the object: This research aims to create a personalized search engine capable of produce results tailored to the specific requirements of each user. These customized search results are produced by executing a binary operation by search engine snippets that relate to the user-selected themes. Consequently, without altering any search engine snippets, tailored search results are essentially just rearranged search engine snippets. The intention of the economical GA approach is to create a search result for page clipping that aggregates relevant paragraphs from several sites. These clipping results are produced by running GA across every page that corresponds to the user-selected themes. Consequently, without incorporating any snippets, the clipping results are essentially generated by clipping and synthesizing some portions of the entire page.

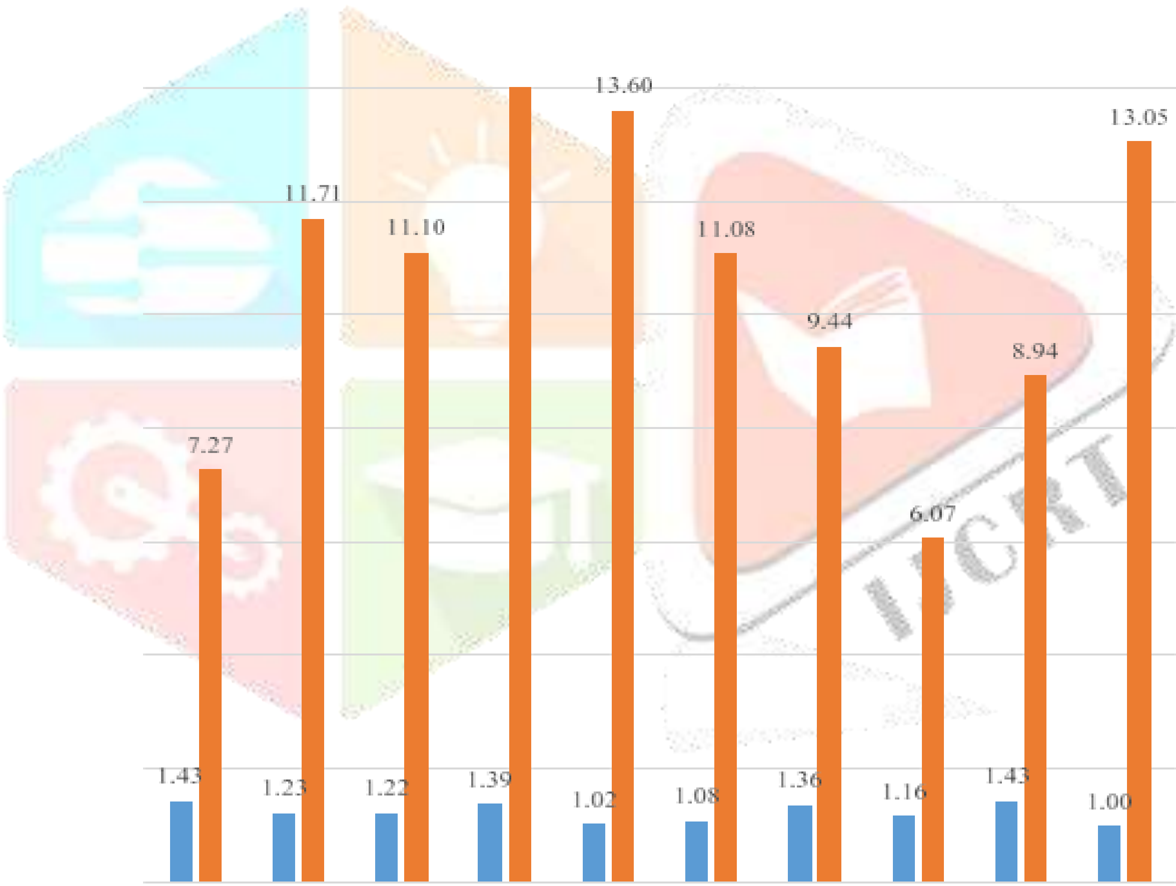


FIGURE8. Comparison of response time between this cost-effective method

TABLE 9.A comparison between this cost-effective method

Point of difference		This study	Cost-effective GA
Solve object		Personalized search	Page clippings
Implementation detail	Topic generation	Mathematical set	Hash
	Collected data	More snippets and no whole page	Fewer snippets and some whole pages
	Ranking function	UBF	MRR
	Noise processing	More	Less
Computation time		Faster	Slower

This study also combines two topics with somewhat identical snippets into a lengthier topic name but with distinct variations on the term. For instance, if the subjects XY and XYZ have essentially identical snippets, we combine them into one topic and simply display the lengthier topic name XYZ. That example, a longer term has a fuller description than a shorter name, thus we utilize it to describe the issue. The data that is gathered and handled differently by the two approaches accounts for the second distinction. The primary data for this study that we gather and analyse are the search engine result snippets. We can save a lot of time and space by using snippets rather than entire pages as post-processing objects since the search engine summarizes the relevant text of the page in the snippet. The economical GA approach initially creates themes with a limited number of snippets. To obtain the results of clipping synthesis, it then runs a GA on the entire page content of connected pages to synthesize various paragraphs on various pages. GA demands a lot of computing time, which means it needs a lot of storage space because each generation's computation results need to be kept for the following one[116]. Longer computing times and storage space are therefore needed for this strategy. Computation time: Next, we experimented to compare how quickly the two approaches responded. In this experiment, two methods are run independently to ascertain the true response time and 1000 queries are used in the test datasets to evaluate the data. The identical computer (Intel Core 2 Duo T9600 with 2GB RAM) is used for both approaches. Using Apache ab [119], a tool for assessing Web server performance, we run a script with all test requests to vary response times for various approaches. A comparison of the two reaction time strategies is shown in Figure 8. The average response time for this study and the cost-effective GA approach for every 100 queries are 1.23 and 10.63 seconds, respectively, based on the results shown in the figure. To obtain a cost-effective solution for clipping, the cost-effective GA method must execute a specific amount of generational GAs. Every generation of GA needs a topper to execute processes related to selection, crossover, and mutation on all chosen page content. The number of pages chosen and the size of each page affect how long each generation of GA takes to execute. The number of generations performed, the number of pages chosen, and the size of each page are therefore the three factors that affect the reaction time of the cost-effective GA method. The number of executions and the size of each snippet in this study are significantly smaller than the number of generations and the page size in the cost-effective GA method, although the number of snippets selected in this study is larger than the number of pages selected by the cost-effective GA method. Based on the reasons for the number of executions.

IV. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

We've suggested a personalized topic search system in this paper. The benefits of the system are as follows. It displays query-related topics first in the form of a hierarchical tree. Giving query-related topics in this format is especially helpful for users who frequently ask shorter, more confusing queries. Second, it can prevent issues with individual seclusion and big storage capacity that frequently arise in personalized searches because it doesn't keep track of user browsing or clicking history. This benefit is that it lowers the company's investment costs in addition to preventing potential privacy breaches that could happen with internet businesses. Thirdly, it can create relationships between many clusters, meaning that the child topics it develops can appear in multiple parent topics. Because a topic can have multiple meanings, multi-cluster relationships can aid users in understanding the meaning of various topics simultaneously. Fourth, It can generate personalized search results and quickly construct hierarchical trees. This guarantees that our framework can react to user queries as fast as other search engines on the internet. The next two tasks are directions for future research. First, we now employ the relevant NLP processing primarily for English papers; nevertheless, there are differences in the processing of NLP for documents written in other languages. Consequently, to assess the content of documents written in different languages, we will search for NLP processing appropriate for that language. Second, even though we employed a lot of datasets for trials that were similar within the trial part, our test dataset was unable to include widely used terms. produced anytime because of the expanding Internet. As a result, we will keep gathering common keywords for the test information set, run associated tests, and modify the system's pertinent parameters considering the experimental findings.

REFERENCES

[1] IDC. (2014). *The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things*. Accessed: Jan. 10, 2019.[Online]. Available: https://goo.gl/GbmFKN

[2] Y.Li,C.Luo,andS.M.Chung,"Aparalleltextdocumentclusteringalgorithmbasedonneighbors,"*ClusterComput.*,vol.18,no.2,pp.933– 948,Jun. 2015.

[3] D.E.RoseandC.Stevens,"V-Twin:Alightweightengineforinteractiveuse,"in*Proc.5thTextRetr.Conf.(TREC)*,Gaithersburg,MD,USA,1996,pp. 1–12.

[4] F. Wiesman, H. J. van den Herik, and A. Hasman, "Information retrieval by meta browsing,"*J.Amer.Soc.Inf.Sci.Technol.*,vol.55,no.7,pp. 565–578, May 2004.

[5] A. K. Sharma, N. Aggarwal, N. Duhan, and R. Gupta, "Web searchresultoptimizationbyminingthesearchenginequerylogs,"in*Proc.Int.Conf.MethodsModelsComput.Sci.(ICM2CS)*,Dec.2010,pp. 39–45.

[6] B. J. Jansen, A. Spink, and T. Saracevic, "Real life, real users, and real needs: A study and analysis of user queries on the Web," *Inf. Process.Manage.*, vol. 36, no. 2,pp. 207–227, Mar. 2000.

[7] C. Carpineto, S. Osinski, G. Romano, and D. Weiss, "A survey of Webclusteringengines,"*ACMComput.Surveys*,vol.41,no.3,pp. 1– 38,Jul. 2009.

[8] W.K.T.M.Gunarathne,C.Chootong,W.Sommool,A.Ochirbat,Y.-C. Chen, S. Reisman, and T. K. Shih, "Web-based learning objectsearchenginesolutiontogetherwithdatavisualization:ThecaseofMERLOT II," in *Proc. IEEE 42nd Annu. Compute. Softw. Appl. Conf.(COMPSAC)*, Jul. 2018,pp. 1026–1031.

[9] J. Wang, J. Z. Huang, J. Guo, and Y. Lan, "Query ranking model for search engine query recommendation,"*Int. J. Mach. Learn.Cybern.*,vol. 8, no. 3,pp. 1019–1038, Jun.2017.

[10] M. R. Henzinger, "Link analysis in Web information retrieval," *IEEEData Eng. Bull.*,vol. 23,no. 3,pp. 3–8,Sep. 2000.

[11] C. Palihawadana and G. Poravi, "A comparative study of link analysis algorithms," in *Proc. 8th Int. Conf. Intell. Syst., Model. Simul. (ISMS)*,May 2018, pp. 100–104.

[12] Y. Li, "Toward a qualitative search engine," *IEEE Internet Comput.*,vol. 2, no.4, pp. 24–29,4th Quart., 1998.

[13] K. Sugihara and J. Nanzan, "Using complex numbers in website ranking calculations: A non-ad hoc alternative to Google's PageRank," *J. Softw.*,vol. 14, pp. 58–64, Feb. 2019.

[14] M.SperettaandS.Gauch,"Personalizedsearchbasedonusersearchhistories,"in*Proc.IEEE/WIC/ACMInt.Conf.WebIntell.(WI)*, Compiegne,France, 2005, pp. 622–628.

[15] S. Dey and S. Abraham, "User interface for a search engine: A customized and multi-domain approach," *Int. J. Comput. Inf. Syst. Ind.Manage. Appl.*, vol.4, pp.169–179, Dec. 2012.

[16] H.Zamani,M.Bendersky,X.Wang,andM.Zhang,"Situationalcontextforrankinginpersonalsearch,"in*Proc.26thInt.Conf.WorldWideWeb*, Apr. 2017, pp. 1531–1540.

[17] P. Ferragina and A. Gulli, "A personalized search engine based on Websnippethierarchicalclustering,"*Softw.,Pract.Exper.*,vol.38,no.2,pp. 189–225, Feb. 2008.

[18] R. Brenneke, T. Mandl, and C. Womser-hacker, "The development and application of an evaluation methodology for person search engines," in*Proc. 1st Eur. Workshop Hum.-Comput. Interact. Inf. Retr.*, Newcastle,U.K., 2011, pp. 42–45.

[19] C.N.Pushpa,N.K.V.Kumar,T.Shivaprakash,J.Thriveni,S.H.Manjula, K.R.Venugopal,andL.M.Patnaik,"ImprovingtheprecisionandrecallofWebpeoplesearchusinghashtableclustering,"in*Proc.5thInt.Conf.Inf. Process.*, Bengaluru,India, 2011,pp. 155–160.

[20] D.Verma,K.Minocha,andB.Kochar,"Amulti-agent-based personalized search engine with topical crawling capabilities," *IUP J. Comput. Sci.*,vol. 8, pp. 20–33, 2014.

[21] N. Saxena, S. Agarwal, and V. Katiyar, "Personalized Web search using user identity," *Int. J. Comput. Appl.*, vol. 147, no. 12, pp. 14– 17, 2016.

[22] F.Akhlaghian,B.Arzanian,andP.Moradi,"A personalizedsearchengineusingontologybasedfuzzyconceptnetworks,"in*Proc.Int.Conf.Data Storage DataEng.*, Feb. 2010,pp. 137– 141.

[23]Z. Jiang and X. Deng, ''A personalized search engine model based onRSS user's interest,'' in *Proc. 2nd Int. Conf. Future Comput. Commun.*,2010, pp. V2-196–V2-199.

[24]K. W.-T. Leung, D. L. Lee, W. Ng, and H. Y. Fung, ''A framework forpersonalizingWebsearchwithconceptbaseduserprofiles,''*ACMTrans.Internet Technol.*, vol. 11, no. 4, pp. 1–29, Mar. 2012.

[25]V. Mishra, P. Arya, and M. Dixit, ''Improving mobile search throughlocationbasedcontextandpersonalization,''in*Proc.Int.Conf.Commun.Syst.Netw. Technol.*, May 2012, pp. 392–396.

[26]A. Singh and B. Alhadidi, ''Knowledge orientedpersonalized searchengine:AsteptowardswisdomWeb,''*Int.J.Comput.Appl.*,vol.76,no.8,pp. 1–9, 2013.

[27]N. Ramesh and J. Andrews, ''Personalized search engine using social networking activity,'' *Indian J. Sci. Technol.*, vol. 8, no. 4, pp. 301–306,2015.

[28]S.A.Tabrizi,A.Shakery,H.Zamani,andM.A.Tavallaei,''PER-SON: Personalized information retrieval evaluation based on citationnetworks,''*Inf.Process.Manage.*,vol.54,no.4,pp. 630–656,Jul.2018.

[29]X.Wang,N.Golbandi,M.Bendersky,D.Metzler,andM.Najork,''Positionbiasestimationforunbiasedlearningt orankinpersonalsearch,''in*P roc.11thACMInt.Conf.WebSearchDataMining(WSDM)*,Marina DelRey, CA,USA, 2018,pp. 610–618.

[30]Y.Nam,D.Shin,andD.Shin,''Personalsearchsystembasedonandroidusing lifelog and machine learning,'' *Pers. Ubiquitous Comput.*, vol. 22,no. 1, pp. 201–218, Feb. 2018.