ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

Online Generalization on User Profiles to Protect the Personal Privacy without Compromising the Search Quality

AR. Arunachalam, Sundararajan.M, Arulselvi S

Assistant Professor, Dept. of CSE, Bharath University, Chennai, Tamil Nadu, India

Director, Research Center for Computing and Communication, Bharath University, Chennai, Tamil Nadu, India

Co-Director, Research Center for Computing and Communication, Bharath University, Tamil Nadu, India

ABSTRACT: Personalized web search (PWS) has demonstrated its effectiveness in improving the quality of various search services on the Internet. The surveys show that users' reluctance to disclose their private information during search has become a major barrier for the wide proliferation of PWS. To protect user privacy in profile-based PWS, we have to consider two contradicting effects during the search process. On the one hand, to attempt to improve the search quality with the personalization utility of the user profile. On the other hand, to need to hide the privacy contents existing in the user profile to place the privacy risk under control. A few surveys , suggest that people are willing to compromise privacy if the personalization by supplying user profile yields better search quality. Here we use a more efficient index structure, the Generalized Inverted IndeX (Ginix), which merges consecutive IDs in inverted lists into intervals to save storage space. With this index structure, more efficient algorithms can be devised to perform basic keyword search operations. Thus, user privacy can be protected without compromising the personalized search quality. Now, there is a tradeoff between the search quality and the level of privacy protection achieved from generalization.

KEYWORDS: Ginix, Keyword search, Privacy protection, personalized web search, profile.

I. INTRODUCTION

The web search engine has long become the most Important portal for ordinary people looking for useful information on the web. However, users might experience failure when search engines return irrelevant results that do not meet their real intentions. Such irrelevance is largely due to the **enormous variety of users' contexts and** backgrounds, as well as the ambiguity of texts. Personalized web search (PWS) is a general category of search techniques aiming at providing better search results, which are tailored for individual user needs. As the expense, user information has to be collected and analyzed to figure out the user intention behind the issued query.

With the huge amount of new information, keyword search is critical for users to access text datasets. These datasets include textual documents (web pages), XML documents, and relational tables (which can also be regarded as sets of documents). Users use keyword search to retrieve documents by simply typing in keywords as queries. Current keyword search systems usually use an inverted index, a data structure that maps each word in the dataset to a list of IDs of documents in which the word appears to efficiently retrieve documents.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

The inverted index for a document collection consists of a set of so-called inverted lists, known as posting lists. Each inverted list corresponds to a word, which stores all the IDs of documents where this word appears in ascending order. In practice, real world datasets are so large that keyword search systems usually use various compression techniques to reduce the space cost of storing inverted indexes. Compression of inverted index not only reduces the space cost, but also leads to less disk I/O time during query processing. As a result compression techniques have been extensively studied in recent years. Since IDs in inverted lists are sorted in ascending order, many existing techniques, such as Variable-Byte Encoding (VBE) and PForDelta, store the differences between IDs, called d-gaps, and then use various techniques to encode these d- gaps using shorter binary representations. To address this problem, this paper presents the Generalized INverted IndeX (Ginix), which is an extension of the traditional inverted index (denoted byInvIndex), to support keyword search. Ginix encodes consecutive IDs in each inverted list of InvIndex into intervals, and adopts efficient algorithms to support keyword search using these interval lists.

Ginix dramatically reduces the size of the inverted index, while supporting keyword search without list decompression. Ginix is also compatible with existing d-gap-based compression techniques. As a result, the index size can be further compressed using these methods. Technique of document reordering, which is to reorder the documents in a dataset and reassign IDs to them according to the new order to make the index achieve better performance, is also used in this paper.

The contributions of this paper are:

- This paper presents an index structure for keyword search, Ginix, which converts inverted lists into interval lists to save storage space.

- Extensive experiments that evaluate the performance of Ginix are conducted. Results show that Ginix not only reduces the index size but also improves the search performance on real datasets.

II. MOTIVATIONS

To protect user privacy in profile-based PWS, researchers have to consider two contradicting effects during the search process. On the one hand, they attempt to improve the search quality with the personalization utility of the user profile. On the other hand, they need to hide the privacy contents existing in the user profile to place the privacy risk under control.
A few previous studies suggest that people are willing to compromise privacy if the personalization by supplying user profile to the search engine yields better search quality. In an ideal case, significant gain can be obtained by personalization at the expense of only a small (and less-sensitive) portion of the user profile, namely a generalized profile. Thus, user privacy can be protected without compromising the personalized search quality. In general, there is a tradeoff between the search quality and the level of privacy protection achieved from generalization.

Unfortunately, the previous works of privacy preserving PWS are far from optimal. The problems with the existing methods are explained in the following observations:

1. The existing profile-based PWS do not support runtime profiling. A user profile is typically generalized for only once offline, and used to personalize all queries from a same user **indiscriminatingly. Such "one profile fits all" strategy certainly** has drawbacks given the variety of queries. One evidence reported is that profile-based personalization may not even help to improve the search quality for some ad hoc queries, though exposing user profile to **a server has put the user's privacy at** risk. A better approach is to make an online decision on

a. whether to personalize the query (by exposing the profile) and

b. what to expose in the user profile at runtime. To the best of our knowledge, no previous work has supported such feature.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

2. The existing methods do not take into account the customization of privacy requirements. This probably makes some user privacy to be overprotected while others insufficiently protected. For example, all the sensitive topics are detected using an absolute metric called surprisal based on the information theory, assuming that the interests with less user document support are more sensitive. However, this assumption can be

doubted with a simple counterexample: If a user has a large **number of documents about "sex," the** surprisal of this topic **may lead to a conclusion that "sex" is** very general and not

sensitive, despite the truth which is opposite. Unfortunately, few prior work can effectively address individual privacy needs during the generalization.

3. Many personalization techniques require iterative user interactions when creating personalized search results.

They usually refine the search results with some metrics which require multiple user interactions, such as rank scoring , average rank , and so on. This paradigm is, however, infeasible for runtime profiling, as it will not only pose too much risk of privacy breach, but also demand prohibitive processing time for profiling. Thus, we need predictive metrics to measure the search quality and breach risk after personalization, without incurring iterative user interaction.

III. BASIC CONCEPTS OF GINIX

Let D={d1, d2, ……, d_N } be a set of documents. Each document in D includes a set of words, and the set of all distinct words in D is denoted by W . In the inverted index of D, each word w € W has an inverted list, denoted by I_W, which is an ordered list of IDs of documents that contain the word with all lists (ID lists and interval lists) sorted in ascending order. For example, Table below shows a collection of titles of 7 papers and gives its inverted index. The inverted index of this sample dataset consists of 18 inverted lists, each of which corresponds to a word. This example shows the lists of 4 most frequent words,

Table
sample dataset of 7 paper titles

Dataset content

ID	Content
1	Keyword querying and ranking in databases
2	Keyword searching and browsing in databases
3	Keyword search in relational databases
4	Efficient fuzzy type-ahead search
5	Navigation system for product search
6	Keyword search on spatial databases
7	Searching for hidden-web databases

Ginix

Word Intervals	
Keyword	1,2,3,6
……..	

InvIndex

Word IDs	
Keyword	[1,3],[6,6]
……..	

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

Databases	[1,3],[6,7]	Databases	1,2,3,6,7
Searching	[2,2],[7,7]	Searching	2,7
Search	[3,6]	Search	3,4,5,6
……		……	

i.e., "keyword", "databases", "searching", and "search" (word stemming is not considered). Lists in inverted indexes can be very long for large datasets, and many existing approaches have paid much attention to how to compress them. An important observation is that there are many consecutive IDs on the inverted lists. The size of the whole inverted index can be reduced by merging these groups of consecutive IDs into intervals, since each interval, denoted by r, can be represented by only two numbers. For example, the ID list of databases in the sample dataset can be converted into an interval list represented by {[1, 3], [6, 7]}. We call the new index structure in which all the ID lists of a standard inverted index are converted into interval lists (called equivalent interval lists) the generalized inverted index (Ginix). Table above shows the generalized inverted index for the sample dataset. Ginix is more appropriate for those datasets whose documents are short or structured because relational tables usually have some attribute values that are shared by many records. As a result, inverted lists contain many consecutive IDs and the size of Ginix will be much smaller than a traditional inverted index. In addition, in such datasets, other information in the inverted lists such as the frequency information and position information do not significantly impact either the query processing or result ranking. Thus, this paper only considers structured or short documents and does not consider the frequency and position information.

A straightforward way to store an interval in Ginix is to explicitly store both its lower and upper bounds, as is illustrated in Table above. However, if an interval [l, u] is a single-element interval, i.e., l = u, two integers are still needed to represent the interval. Thus if there are many single-element intervals in the interval list, the space cost will be expensive. The extra overhead for storing the interval lists is reduced by splitting each original interval list into 3 ID lists with one for single element intervals and the other two for the lower and upper bounds of multi-element intervals. These three lists are denoted as S, L, and U. For example, the interval list {[1, 1],[3, 3], [6, 7], [9, 9],[12, 15]} can be split into 3 ID lists with S={1, 3, 9} L= {6, 12}, and

U={7, 15}. This reduces the number of integers from 10 to 7. Efficient sequential/sorted access is a basic requirement of keyword search based on the interval lists. Two position indicators, p and q, are used here to indicate the current positions in S and L/U. At the beginning, p and q are all set to 0, indicating that they are all pointing to the first elements in S and L/U. The current interval is found by comparing the two elements

Sp and Lq. If Sp is smaller, we return the single element interval .Sp, Sp• and increment p by 1, if Lq is smaller, return the multi-element interval [Lq,Uq]• and increment q by 1. Given an ID list S containing n IDs and its equivalent interval list R, the three lists, R.S, S.L, and S.R, used to store R will contain no more than n integers in total. This property of interval lists means that Ginix can be regarded as a compression technique, which is orthogonal to d-gap-based techniques. Moreover, d- gap-based compression algorithms, such as VBE and PForDelta, can still be applied to Ginix, since all the lists in Ginix are ordered lists of IDs.

IV. SEARCH ALGORITHMS

A keyword search system usually supports union and the intersection operations on inverted lists. The union operation is a core operation to support OR query semantics in which every document that contains at least one of the query keywords is returned as a result. The intersection operation is used to support AND query semantics, in which only those documents that contain all the query keywords are returned.

Traditional search algorithms are all based on ID lists. Specifically, a traditional keyword search system first retrieves the compressed inverted list for each keyword from the disk, then decompresses these lists into ID lists, and then calculates the intersections or unions of these lists in main memory. This method introduces extra computational costs

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

for decompression, and ID list based search methods can be very expensive because ID lists are usually very long.

Union operation
As in set theory, the union of a set of ID lists, denoted by
S= { S1, S2 ,…..,Sn}, is another ID list, in which each ID is contained in at least one IDlist in S. Thus the union of a set of interval lists can bedefined as follows:
Union of Interval Lists - Given a set of interval lists,

R ={R1,R2,…,Rn}, and theirequivalent ID lists, S

={S1, S2, .., Sn}, the union of R is the equivalent interval list
of $U^{n}_{k=1}S_{k.}$.

For example, consider the following three interval lists: {(2, 7),(11, 13)},{(5, 7),(12, 14)} and{(1, 3),(6, 7)},{(9,9), (12,15)}. Their equivalent ID lists are {2, 3, 4, 5, 6, 7, 11, 12, 13}, {5, 6, 7, 12, 13, 14}, and {1, 2, 3, 6, 7, 9, 12, 13, 14, 15}. The union of these three ID lists is {1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15}; thus, the union of the three interval lists is the equivalent interval list of this ID list, i.e., {[1, 7], [9,

9],[11,15]}.

In this algorithm, the interval lists are first converted into ID lists with the union calculated using the well known multi-way merge algorithm and the result then converted back into an interval list. This method is called the NAÏVE UNION algorithm. Since the goal is to design an algorithm for calculating the union of interval lists without list conversion, this method will be used as a baseline for comparison.

Intersection of Interval Lists

Given a set of interval lists, R = {R1,R2**,…**Rn}, and their equivalent ID lists, S = {S1, S2, **…** ,Sn}, the intersection of R is the equivalent interval list of $N_{k=1}{}^{n}S_{k.}$.

Consider the three interval lists that we have used previously: {[2, 7],[11,13]}, {[5, 7],[12,14]},• and {[1,3],[6,7],[9, 9],[12,15]}. Their equivalent ID lists are {1,2, 3, 4, 5, 6, 7, 11,12,13}, {5, 6, 7, 12, 13, 14}, and {1,2, 3, 6, 7, 9, 12, 13, 14, 15},respectively. The intersection list of these ID lists is {6,7,12 ,13}, thus the intersection of the interval lists is the equivalent interval list of this ID list, i.e.,{[6,7],[12,13]}.

V. CONCLUSIONS

In this paper we perform online generalization on user profiles to protect the personal privacy without compromising the search quality. We proposed two greedy algorithms, namely GreedyDP and GreedyIL, for the online generalization. We also describe a generalized inverted index for keyword search in text databases. Ginix has an effective index structure and efficient algorithms to support keyword search.Experiments show that Ginix not only requires smaller storage size than the traditional inverted index, but also has a higher keyword search speed. Moreover, Ginix is compatible with existing d- gap-based list compression techniques and can improve their performance

REFERENCES

[1] May Wang and Benjamin Yen, "Web Structure Reorganization to Improve Web Navigation Efficiency", 11th Pacific-Asia Conference on Information System.
[2] Sathish Kumar M., Karrunakaran C.M., Vikram M., "Process facilitated enhancement of lipase production from germinated maize oil in Bacillus spp. using various feeding strategies", Australian Journal of Basic and Applied Sciences, ISSN : 1991-8178, 4(10) (2010) pp. 4958-4961.
[3] J. Teevan, S.T. Dumais, and E. Horvitz, "Personalizing Search via Automated Analysis of Interests and Activities," Proc. 28th

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

Ann.Int'l ACM SIGIR Conf. Research and Development in Informatio Retrieval (SIGIR), pp. 449-456, 2005.
[4] Kaliyamurthie K.P., Parameswari D., Udayakumar R., "QOS aware privacy preserving location monitoring in wireless sensor network", Indian Journal of Science and Technology, ISSN : 0974-6846, 6(S5) (2013) pp.4648-4652.
[5] M. Spertta and S. Gach, "Personalizing Search Based on User Search Histories," Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence (WI), 2005.
[6] Sharmila D., Muthusamy P., "Removal of heavy metal from industrial effluent using bio adsorbents (Camellia sinensis)", Journal of Chemical and Pharmaceutical Research, ISSN : 0975 – 7384, 5(2) (2013) pp.10-13.
[7] K. Sugiyama, K. Hatano, and M. Yoshikawa, "Adaptive Web Search Based on User Profile Constructed without any Effort from Users," Proc. 13th Int'l Conf. World Wide Web (WWW), 2004.
[8] Udayakumar R., Khanaa V., Saravanan T., Saritha G., "Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction", Middle - East Journal of Scientific Research, ISSN : 1990-9233, 16(12) (2013) pp.1781-1785.
[9] X. Shen, B. Tan, and C. Zhai, "Implicit User Modeling for Personalized Search," Proc. 14th ACM Int'l Conf. Information and Knowledge Management (CIKM), 2005.
[10] Kalaiselvi V.S., Prabhu K., Ramesh M., Venkatesan V., "The association of serum osteocalcin with the bone mineral density in post menopausal women",Journal of Clinical and Diagnostic Research, ISSN : 0973 - 709X, 7(5) (2013) pp.814-816.
[11] X. Shen, B. Tan, and C. Zhai, "Context-Sensitive Information Retrieval Using Implicit Feedback," Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development Information Retrieval (SIGIR), 2005
[12] J. Pitkow, H. Schu¨ tze, T. Cass, R. Cooley, D.
Turnbull, A. Edmonds, E. Adar, and T. Breuel, "Personalized Search," Comm. ACM, vol. 45, no. 9, pp. 50-55, 2002.
[13] Y. Xu, K. Wang, B. Zhang, and Z. Chen, "Privacy-Enhancing Personalized Web Search," Proc. 16th Int'l Conf. World Wide Web (WWW), pp. 591-600, 2007.
[14] F. Scholer, H. E. Williams, J. Yiannis, and J. Zobel, Compression of inverted indexes for fast query evaluation, in Proc. of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tammpere, Finland, 2002, pp. 222-229.
[15] M. Zukowski, S. Hman, N. Nes, and P. A. Boncz,Superscalar RAM-CPU cache compression, in Proc. of the 22nd International Conference on Data Engineering, Atlanta, Georgia, USA, 2006, pp. 59.
[16] J.Arul Hency Sheela, GC-MS Studies of the Plant Clematis Gouriana , International Journal of Innovative Research in Science, Engineering and Technology , ISSN: 2319-8753 ,pp 13514-13519 ,Vol. 3, Issue 6, June 2014.
[17] Jemima Daniel,Usage of Language,International Journal of Innovative Research in Science, Engineering and Technology,ISSN: 2319-8753, pp 7073-7075,Vol. 2, Issue 12, December 2013.
[18] Jemima Daniel,The Duality of Human Nature,International Journal of Innovative Research in Science, Engineering and Technology,ISSN: 2319-8753,pp 511-512,Vol. 2, Issue 2, February 2013.
[19] Jemima Daniel,Impact of E-Mail communication,International Journal of Innovative Research in Science, Engineering and Technology,ISSN: 2319-8753,pp 527-528,Vol. 2, Issue 2, February 2013.
[20] Jemima Daniel,The Enchanting World In Karnas`S Plays,International Journal of Innovative Research in Science, Engineering and Technology,ISSN: 2319-8753,pp 757-758,Vol. 2, Issue 3, March 2013.
[21] Jemima Daniel,Myth in Indian English Dramas,International Journal of Innovative Research in Science, Engineering and Technology,ISSN: 2319-8753,pp 1551-1555,Vol. 2, Issue 5, May 2013.