Supporting Context-based Query in Personal DataSpace

Yukun Li
School of Information
Renmin University of China
Beijing, China
liyukun@ruc.edu.cn

Xiaofeng Meng
School of Information
Renmin University of China
Beijing, China
xfmeng@ruc.edu.cn

ABSTRACT

Many users need to refer to content in existing files (pictures, tables, emails, web pages and etc.) when they write documents(programs, presentations, proposals and etc.), and often need to revisit these referenced files for review, revision or reconfirmation. Therefore it is meaningful to discover an approach to help users revisit these references effectively. Traditional approaches(file explorer, desktop search, and etc.) fail to work in this case. In this paper, we propose an efficient solution for this problem. We firstly define a new personal data relationship: Context-based Reference(CR), which is generated by user behaviors. We also propose efficient methods to identify CR relationship and present a new type of query based on it: Context-based Query(C-Query), which helps users efficiently revisit personal documents based on CR relationship. Our experiments validate the effectiveness and efficiency of our methods.

Categories and Subject Descriptors

H.2.m [**Database Management**]: Miscellaneous

General Terms

Algorithms, Human Factors, Performance

Keywords

Context, Query, Personal DataSpace

1. INTRODUCTION

With development of information technology, more and more personal data items are collected, and Personal Information Management (PIM) [1] becomes a critical problem and a promising research area. Studies show that many personal data accesses($> 58\%$) are "revisit" [3, 4, 5], and "meaningful" data relationships(senderOf, authorOf, publishedIn and etc.) can help users relocate expected items more effectively [2]. However, there are two basic questions need to be answered: (1)what relationships are "meaningful" and (2) how to identify these "meaningful" relationships.

Since the aim of identifying data relationships is to improve effectiveness of data query, the definition of "meaningful" depends on user query requirements. When users produce personal documents(programs, presentations, proposals, and etc.), they often refer to some contents in existing files. In addition, when a user accesses one of her documents or redo a task, she often needs to revisit its references for revision or reconfirmation. Therefore "referenceOf" is one of the "meaningful" relationships of personal data.

The popular tools used by persons to revisit expected documents are folder explorer and desktop search. Folder explorer demands users remember path and name of the expected files. If a user can only remember fuzzy information, she has to try possible paths many times. Therefore folder explorer can not work well in this case. Desktop search demands users remember keywords included by the expected files, which does not work well when a user can not remember exact keywords.

There are also some works on Personal DataSpace(PDS) model [6], personal data integration [7, 8], index [9] and query [11]. But all these works focus on improving efficiency of personal data operation by identifying objective associations of items(senderOf, authorOf, and etc.). Our work is different, we focus on proposing a new personal data relationship based on user behaviors and a new type of query. Our main contributions can be summarized as below.

- Propose a new semantic relationship between personal data items: *Context-Based Reference(CR)*, which is generated by user behaviors. We also propose an effective method for identifying CR relationship.
- Propose a new type of query in PDS: *Context-based Query(C-Query)*, which is based on CR relationship. We give a solution framework of C-Query and propose an efficient approach for C-Query processing.

The rest of this paper is organized as follows: In Section 2, we give a solution framework. In Section 3, we describe the algorithms for identifying CR relationship. In Section 4, we introduce the approach for C-Query processing. Section 5 evaluates our measures and section 6 concludes this paper.

2. SOLUTION FRAMEWORK

As shown in Figure 1. Our solution framework includes four parts: CR Database(CDB), Context-based Reference Relation(CRR) Identifier, C-Query Engine and Query Interface.

CIKM'09, November 2–6, 2009, Hong Kong, China.

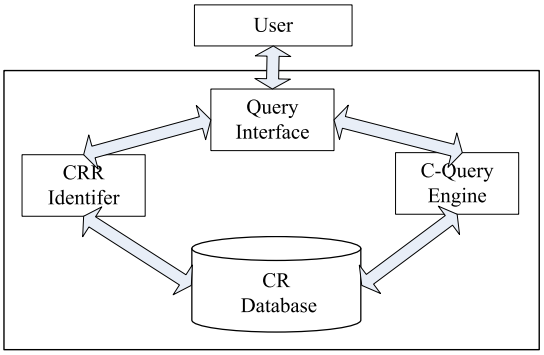


Figure 1: C-Query Implementation Framework

- **CR Database:** It is the data structure for describing personal data items and data relationships. We take several relation tables to store CR relationship, because relation tables are simply implemented and work at a high performance.
- **CRR identifier:** It monitors user operations. When a user conducts an operation, it captures user operation and update CR database in time.
- **C-Query Engine:** It handles user query and produces results. When a user submits a C-Query requirement, it produces result based on the CR database and user input.
- **Query Interface** It is utilized to handle user input and display query results. To help users relocate expected items quickly, it allows users to refine query results easily.

In this framework, CRR identifier is the basis of our solution, and is also a big challenge, because (1) there is no explicit information for mining this relationship and (2) the approach for identifying CRR shouldn't increase users' burden. Studies [10] show that the two items accessed continuously often has associations. Inspired from this idea, we propose a method for identifying CR relationship based on user behaviors. To make it more clear, we first introduce the following concepts.

Definition 2.1 (Personal Data Item). *A personal data item(PDI) is the basic element of personal data, and is the smallest unit of personal data operation(read, modify, delete, and etc.).*

There are multiple data relationships among personal data items, such as senderOf, authorOf, referenceOf, and so on. Based on personal data items and their relationships, we propose a new concept: Personal DataSpace.

Definition 2.2 (Personal DataSpace). *A Personal dataspace $\mathcal{D}$ is described as a 2-tuple ($\mathcal{N}$, $\mathcal{R}$), where $\mathcal{N}$ is a set of personal data items and $\mathcal{R}$ is a set of personal data relationships.*

Definition 2.3 (Context-based Reference Relation). *We denote it as $R^{CR}(I_1, I_2, U)$, where I_1 and I_2 are two personal data items, and U is a user, which means there is a reference relationship between I_1 and I_2, which is generated by activities of U.*

3. CR-RELATIONSHIP IDENTIFYING

In this section we first overview CR relationship, then present algorithms for identifying CR-Relationship.

3.1 Overview CR Relationship

To tackle the problem of absence of public personal data set, we implement a prototype to capture user access behaviors(operations on desktop, email box and web pages). We run it in personal computers of five persons of our group, and obtain a data set, which includes the access logs of the five users in two months. Based on analyzing these user access logs we propose a new concept: *Time Sequential List.*

Definition 3.1 (Time Sequential List). *A Time Sequential List(TSL) is an item list ordered by access sequence. We say it $(I_1, I_2, ..., I_n)$, where I_i is a personal data item, and $\forall i$, if $1 \leq i \leq n-1$, $I_i <> I_{i+1}$.*

In a TSL, there is no two sequent items mapping to a same item. Figure 2 shows an example of time sequential list. By analyzing the access logs of the five users, we discover three types of CR relationship: Sequence Adjacent Relation(SAR), Sequential Inclusive Relation(SIR) and Lineage Relation(LR). To make them clear, we define them as below.

Definition 3.2 (Sequencial Adjacent Relation). *We denote it as binary relation. Let I_i and I_j are two items of PDS, if I_i and I_j appear in TSL sequentially, $(I_i, I_j) \in R^{SAR}$.*

We define SAR as a symmetrical relation, it means if $(A, B) \in R^{SAR}, (B, A) \in R^{SAR}$. Take the access list shown in figure 2 for example, A and C are accessed frequently, then $(A, C) \in R^{SAR}$ and $(C, A) \in R^{SAR}$.

Definition 3.3 (Item Sequential Loop). *Let L' be a TSL and $L' = (X_1, X_2, ..., X_n)$. If $L'' = (X_i, X_{i+1}, ..., X_j)$ is a sub list of L', $j - i \geq 2$ and $X_i.item = X_j.item$. We call L'' an item sequential loop(ISL). We call $X_i.item$ the master item, and call the items of $\{X_{i+1}, ..., X_{j-1}\}$ slave items.*

In the example shown by figure 2, there are following ISLs (A,B,A), (A,C,D,A), (B,E,F,G,B), and so on.

Definition 3.4 (Sequential Inclusive Relation). *A sequential inclusive relation R^{SIR} is a binary relation. Let L' be an item sequential loop, X' is the master item, and $Y_1, Y_2, .., Y_m$ are the slave items of it, $\{(X', Y_i) | 1 \leq i \leq m\} \subseteq R^{SIR}$.*

In the example shown by figure 2, (A, C, D, A) is a ISL, then $\{(A, C), (A, D)\} \in R^{SIR}$.

Definition 3.5 (Lineage Relation). *We denote it as $R^{LR}(I_1, I_2)$, where I_1 and I_2 are two items of PDS, $R^{LR}(I_1, I_2)$ denotes I_1 and I_2 are two versions of a same personal data item. We define LR as symmetrical relation.*

Figure 2 shows an example of user access sequential list, where A is the early version of H, therefore $(A, H) \in R^{LR}$ and $(H, A) \in R^{LR}$. LR is also generated by user behaviors(copy to, save as, and etc.).

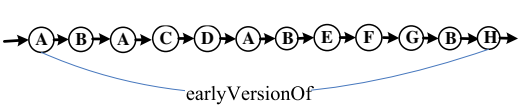


Figure 2: An example of time sequential list

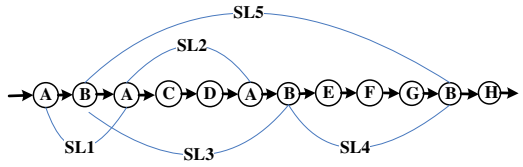


Figure 3: Sequential loop examples

3.2 Identify Sequential Adjacent Relation

We take a triple set $TS = \{(x_i, x_j, w)\}$ to specify SAR relationship, where x_i and x_j are two items and w is the weight of $R^{SAR}(x_i, x_j)$. We define w as the times the two items orderly appear in ISL. Based on the access sequence list, we can construct TS easily. Its input is an item access list $(X_1, X_2, ..., X_n)$, and its output is a triple set T'. In the list each X_i represents an operation, and $X_i.I$ represents the item referenced by X_i. Firstly, we scan the items of the access list one by one, if $(X_i.I, X_{i+1}.I) \in T'$, $W(X_i.I, X_{i+1}.I)$ is added by 1, otherwise we insert a new tuple $(X_i.I, X_{i+1}.I)$, and set its weight as 1. It is an incremental process to build TS. Every time when a new operation is monitored, TS will be updated at once.

3.3 Identify Sequential Inclusive Relation

Based on Item Sequence Loop(ISL), we can derive Sequence Inclusive Relation(SIR). In the definition of ISL, we do not set limitation for the length of ISL, obviously it results in low precision. Therefore we propose a new concept.

DEFINITION 3.6 (MINIMUM SEQUENTIAL LOOP). *Let L' be a SL, and $\nexists L''$, L'' be a SL and $L'' \subset L'$, we call L' a minimum sequential loop(MSL).*

For example, as shown in figure 3, SL_2 is included in SL_3, and SL_3 is included in SL_5, thus SL_3 and SL_5 are not MSL. Because there are not SLs in SL_1, SL_2 and SL_4, they are MSLs.

Our method for identifying SIR is based on MSL. Let $(X_1, X_2, ..., X_n)$ be a time sequence list, and $(I_1, I_2, ..., I_n)$ is the corresponding item list. Assume a new operation X_{n+1} is monitored, and its item is I_{n+1}, we scan backwards to find a MSL mastered by the new item I_{n+1}. When we find the nearest X_i, where $Xi.I = X_{n+1}.I$, and there does not exist a SL in the list $(X_i, X_{i+1}, ..., X_{n+1})$, it means we find a MSL $(X_i, X_{i+1}, ..., X_n, X_{n+1})$, and we can identify the following inclusive relations: $\{(I_{n+1}, I_{i+1}), (I_{n+1}, I_{i+2}), ..., (I_{n+1}, I_n)\}$. The same as SAR, We take a triple set to describe SIR.

3.4 Identify Lineage Relation

As the naive method, we can identify it by monitoring the special operations of users(copy to, save as, and etc.). Because this method depends on monitoring specific applications, it is a challenging problem to monitor all possible applications. By analyzing user access logs we find there is a high similarity between the names of two personal files which are two different versions of one document. And users prefer to name different versions of a document with similar strings, and tend to distinct them with prefix or postfix. Based on the observation, we present an edit distance-based approach to decide LR by computing the name similarity of two files. For the reason of space limitation, we do not introduce it in details here.

4. QUERY PROCESSING

After identifying CR relationship, we can revisit personal data items based on it. In this section, we introduce the processing of C-Query. In our method, we take three adjacency matrixes M^{SAR}, M^{SIR} and M^{LR} to specify the three relationships SAR, SIR and LR. Therefore we can compute the query results based on the following formula:

$$M^R = M^I \times M^{LR} \times (M^{SAR} + M^{SIR})$$

Here M^I is an entry vector $(x_1, x_2, ..., x_n)$, if I_i belongs to the input items, $x_i = 1$, else $x_i = 0$. Based on the formula, we can get a result vector M^R. Let t be the threshold predefined, if $M^R(i) > t$ then I_i is one item of the query results. To get a high recall here we take a relax policy: if $M^R(i) > 0$, we think item I_i belongs to the result set. It means we take all "suspicious" items as results.

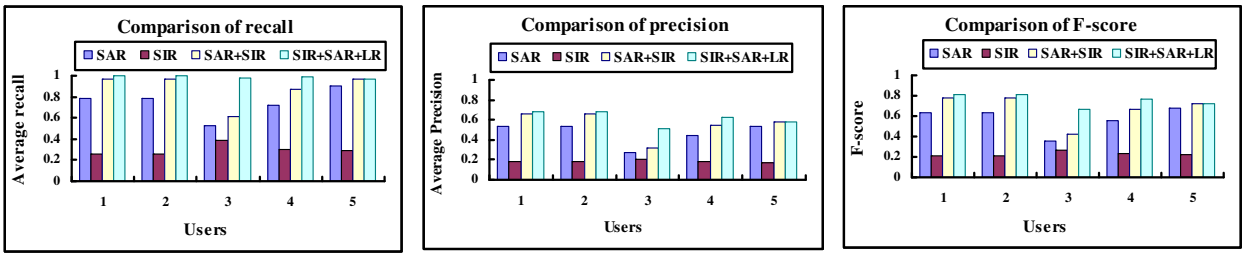
According to the characteristics of C-Query, we design a friendly and flexible interface. Users can input an existing item, or select it by exploring personal resources. It also provides multiple ways for users to refine query results, such as filter or sort the results based on type, access time, keywords, and etc. If the input is a multi-version item, it can display all versions of it and all references of each version.

5. EVALUATION

We selected 5 students(2 undergraduate students, 2 master students and 1 PhD student) as participants of our experiments. They run our prototype on their computers. By this way we collected a data set for experiments. We collected two-month access logs(include accesses to desktop files, emails and web pages) of each participant. Table 1 shows the specification of the data set. The meaning of each column is specified as follows.

- $\mathcal{S}_{logs}$ is the number of tuples of user access log file.
- $\mathcal{S}_{items}$ is the number of items which has been accessed by user U_i in the two months.
- $\mathcal{S}_{LR}$ is the number of elements in the set of Item Lineage Relationship(LR).
- $\mathcal{S}_{SAR}$ is the number of elements in the set of Sequencial Adjacent Relation(SAR).
- $\mathcal{S}_{SIR}$ is the number of elements in the set of Sequencial Inclusive Relationship(SIR).

Our aim is to help users revisit personal documents based on CR relationship. Therefore effectiveness and efficiency are the key factors of our approach. Let I' be the input item, based on the three CR relationships(SAR, SIR and LR), we derive several algorithms to identify CR relationship. (1) SIR. It takes the items with SIR to I' as results. (2) SAR. It takes the items with SAR to I' as results.



(a) (b) (c)

Figure 4: Evaluation of recall, precision and F-score

Table 1: Specification on dataset of experiments

$User$	$\mathcal{S}_{logs}$	$\mathcal{S}_{items}$	$\mathcal{S}_{LR}$	$\mathcal{S}_{SAR}$	$\mathcal{S}_{SIR}$
$User_1$	2314	613	22	1690	250
$User_2$	944	415	13	527	186
$User_3$	2012	792	27	2089	390
$User_4$	3052	915	19	2527	673
$User_5$	802	223	6	539	93

(3)SIR+SAR. It takes the items with SIR or SAR to I' as results. (4)SIR+SAR+LR. It takes the items with SIR or SAR or LR to I' as results.

For each user, we select ten "representative" documents from her access logs. "Representative" means that the selections should not only include "lightweight" personal documents, but also contain some "heavyweight" documents, which cost users more energy, such as paper drafts, presentation slides and etc. These documents often have more Context-based References. We deliver the selected documents to participants and ask them give "standard answer" to each document.

To test the effectiveness of our methods for identifying CR relationship, we take recall and precision to evaluate. We take each selected document as input, and assume that each user U_i submits 10 C-Queries. By comparing the results produced by our algorithms and the right answer given by users, we can compute the recall and precision of each algorithm. We take traditional F-measure method to compute F-score of each method and give evaluation of them.

Figure 4 shows the results of our experiments. Figure 4(a) compares recall of the four methods, Figure 4(b) compares their precision, and Figure 4(c) compare their F-score based on F-measures. The results show that the SIR+SAR+LR method has the best effectiveness. It shows that although we take a relax method, the average precision still reach 60%, and the average recall is more than 90%.

6. CONCLUSIONS

In this paper we propose a new semantic relationship *Context-based reference(CR)* and present a new type of query of PDS *Context-based Query(C-Query)*. We also propose an efficient method to identify CR relationship based on user operation logs, and present the processing method of C-Query. This is only a preliminary work on supporting context-based query in personal dataspace. In the future, we will try to improve the precision of identifying CR relationship by considering more user-related information, and will study the ranking approaches of C-Query results.

7. ACKNOWLEDGMENTS

This research was partially supported by the grants from the National High-Tech Research and Development Plan of China (No:2007AA01Z155, 2009AA011904); the Natural Science Foundation of China under grant number 60833005. And the authors would like thank Xin Dong, Xiaodong Zhou and Wei Lu for some discussions regarding this work.

8. REFERENCES

[1] W. Jones and H. Bruce. A Report on the NSF-Sponsored Workshop on Personal Information Management. PIM worshop 2005.

[2] M. J. Franklin, A. Y. Halevy, and D. Maier. From databases to dataspaces: A New Abstraction for Information Management. SIGMOD Record, 34(4):27-33, 2005.

[3] L. Catledge. and J. Pitkow. Characterizing browsing strategies in the World Wide Web. Computer Networks and ISDN Systems, 27(6), 1065-1073, 1995.

[4] B. McKenzie and A Cockburn. An empirical analysis of web page revisitation. In Proceedings of the 34th International Conference on System Science (HICSS34), 2001.

[5] L. Tauscher and S. Greenberg. How people revisit Web pages: Empirical findings and implications for the design of history systems. International Journal of Human Computer Studies, 47(1), 97-138,1997.

[6] J.-P. Dittrich and M.A.V. Salles. iDM: A Unified and Versatile Data Model for Personal Dataspace Management. VLDB 2006: 367-378

[7] X. Dong and A.Y. Halevy. A Platform for Personal Information Management and Integration. CIDR 2005:119-130.

[8] M.A.V. Salles, J.-P. Dittrich, S.K. Karakashian, O.R. Girard and L. Blunschi. iTrails: Pay-as-you-go Information Integration in Dataspaces, VLDB 2007: 663-674.

[9] X. Dong and A. Halevy. Indexing Dataspace. SIGMOD 2007: 43-54.

[10] P.-A. Chirita and W. Nejdl. Analyzing User Behavior to Rank Desktop Items. SPIRE 2006: 86-97.

[11] C. Peery, W. Wang, A. Marian and T. D. Nguyen. Multi-Dimensional Search for Personal Information Management Systems. EDBT 2008.