An Efficient Method for Constructing Personal DataSpace

Yukun Li, Xiaofeng Meng, Yubo Kou
School of Information
Renmin University of China
Beijing, China
{liyukun,xfmeng}@ruc.edu.cn，yubokou@gmail.com

Abstract—**With increment of personal data amount, how to efficiently manage Personal DataSpace(PDS) becomes a serious problem and a hot research topic. In PDS, users often expect to relocate some known items. Therefore how to efficiently identify these known items from a great number of objects becomes an important research issue. To the best of our knowledge, there is no existing works tackling this problem. In this paper, we propose a concept KnownBy to model the basic relationship between data items and users, and take it as the baseline for deciding if an item belongs to PDS. By analyzing user access logs, we mine some rules for identifying known items, and propose efficient methods for marking known items from personal desktop resources. Our experiments validate the effectiveness and efficiency of our methods, which can be integrated into personal information management systems, such as desktop search tools, and etc.**

Keywords-Personal DataSpace; Rules; Construction

I. INTRODUCTION

With increment of personal data amount, how to efficiently manage Personal DataSpace(PDS)[1][2] becomes a serious problem and a hot research topic. Identifying personal items is the first step of Personal DataSpace Management (PDSM). Different from web search, known-item query [3][4] is a popular type of data access in PDS, which means users know the expected items exist in PDS, and want to recall them for reuse. For example, users often expect to revisit a document of his desktop developed before, but can't remember exact information (file name, directory, , and etc.) on it. In this case, relocating expected items in the mass of disordered files becomes a challenge to users. Although people tend to place personal data items in special folders, it is uneasy for users to manually maintain the categories well. Therefore how to help users efficiently maintain the categories becomes a meaningful research issue, and its first step is to identify PDS boundary. This is the focus of this paper.

A. Related work

To fit the new characteristics of data, a new concept *dataspace* is proposed [1][2], and Personal DataSpace Management(PDSM) becomes a hot research topic. PDS is user-oriented and composes of all items related to the specific person [5]. As the first step of personal dataspace management, personal data integration is paid much attention [6] [7], and these works highlight the importance of the associations of data items to increase efficiency of user operation. These works only take data item associations into considerations, whereas neglect the relationships between data items and PDS owner.

Desktop search engine is a popular tool for searching personal data items, which support keyword-based search by full-text indexing. Because most desktop search tools do not distinct known items from desktop items when creating index, their performance are often very poor. Reference [8] proposes to rank personal data items by exploiting user behaviors. Reference [9] focuses on improving recall by specific ranking policies. Neither of the works mentioned above considers the relationship between persons and items.

Studies show that revisit is a popular type of data access in PDS, and the tools of PDSM used by persons need to accord to user memory rules [10]. The goal of our work is to discover an efficient method to mark PDS boundary so as to help users revisit expected items efficiently.

B. Contribution Summary

Our main contributions in this paper are summarized as bellow:

1) Define a concept KnownBy to model the basic relationship between data items and users, and define PDS boundary based on it.
2) Analyze factors contributing to KnownBy attribute, such as type, name, directory, and so on, and propose two methods for identifying PDS boundary based on KnownBy attribute: content-based algorithms(CA) and structure-based algorithms(CSA).
3) Implement a prototype system to validate the efficiency of our methods, and the results validate the effectiveness and efficiency of these methods.

The rest of this paper is organized as follows: In Section 2, we overview the concepts of personal dataspace. In Section 3, we introduce the rules mined by experiments for identifying PDS boundary. In section 4, we introduce our algorithms for building PDS. Section 5 is evaluation, Section 6 concludes this paper.

II. Personal DataSpace Overview

A. *Personal DataSpace*

PDS is composed of a set of data items related to a specific user. There are three basic elements of PDS: owner, data set and services [11]. PDS owner is a specific entity, which are both administrator and end user. Dataset of PDS is a large set of data items related to the owner. In PDSMS, services are designed to help users manage personal data items, such as index, storage, query, and etc. PDS is made up of data items related to PDS owner, How to define "related to" is a key problem. Here we give some definitions.

Definition 2.1: Personal Data Item(PDI) . A PDI is a data item with relation to PDS owner. For example, a personal document or an email is a personal data item.

Definition 2.2: Person-Item relationship(PIR). Let P' is a user and I' is an item, PIR(P', I') denotes a relationship between P' and I'.

There are many such relationships (senderOf, developedBy, and so on). These relationships play an important role in helping users relocate expected items and are helpful for improving efficiency of PDS queries. We define a basic such relationship below.

Definition 2.3: KnownBy. It is a relationship between an item and a user, we denote it as KnownBy (I, U), which means "I is an item known by person U.

Definition 2.4: Personal dataspace. Personal DataSpace (PDS) is a set of data items known by a user, we denote it as 2-tuple (U',D'), where U' is a particular user and D' is an item set known by U'.

B. *Solution Framework*

The problem tackled by this work is formulated as below: Let P_0 be a person, and S' $=\{a_i\}$ is a set of data items of desktop of P_0, our work is to identify the items of S' which are known by the desktop owner.

Fig.1 shows the framework of our solution, where the input is a set of personal desktop items, and the output is the initial PDS. Our method is based on mining some features of known items by analyzing data items accessed recently. We discover some rules for identifying KnownBy, furthermore propose several methods to identify "KnownBy" items based on these rules.

III. Mining Rules

In this section we introduce the algorithms to mine the rules for identifying "KnownBy" items.

A. *Preliminary*

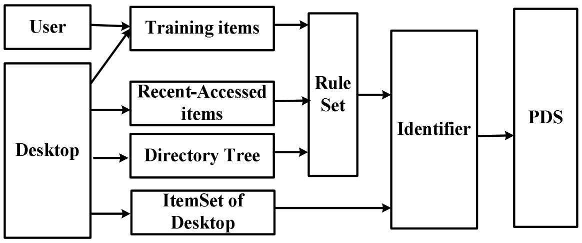


Figure 1. Solution framework for building PDS

We take association rules method for mining rules. Let I $= \{i_1, i_2, ..., i_n\}$ be a set of data items selected for training, A $= \{A_1, A_2, ..., A_m\}$ be a set of attributes, $V_{ij}=\{v_{i1}, v_{i2}, ..., V_{ik}\}$are values of attribute A_i, and U_0 is the user. An association rule is an implication of the form A B, where A is called the antecedent of the rule, and is formulated as $x.A_i = V_{ij}$, , and B is called the consequent of the rule, which means x.KnownBy = True, where x presents a data item. In our method, we consider three attributes, it means A = {type, directory, name}. Let V_{type} = {doc, ppt, pdf, ...}, the rule *(x.type = doc)* $\Rightarrow$ *(x.KnownBy = true)* means if the type of a given item x is doc, then x is an item known by the user U_0.

In general, the correctness of a rule R is a probability value, and it means each rule has a confidence value. To describe it, we introduce two functions: supp(A) and conf(R).

1) supp(A) : It means the number of items which satisfy A, where A is a condition expression, such as *type = 'doc'*, and so on.

2) conf(R): It means the confidence of rule R. Let R means $A \Rightarrow B$, *conf(R)* equals to $\text{supp}(A \wedge B)/\text{supp}(A)$,where $A \wedge B$ means both condition A and B are satisfied.

For example, let S_i be the item set for training and it consists of n(n = 200) items, in which there are 50 doc items, and 30 of them are known by users, then conf((x.type = doc) $\Rightarrow$ (x.KnownBy = true)) = 0.6. It means if a given item x is a doc file, the possibility it is known by the user is 0.6.

B. *Type-based Rules*

Type is a popular attribute related to user preferences. For example, a researcher popularly visits papers of pdf type. On the other hand, a programmer may tend to access java or cpp documents. Fig.2(a) shows the results of type-based statistics. It illustrates confidence of type-based rules. Let t_i be a type, we can get a set of rules r_i, where r_i means the rule ($x.type=t_i$) $\Rightarrow$ (x.KnownBy=Yes), conf(r_i) means the confidence of r_i. For example, if r_0 denote the rule (x.type=doc) $\Rightarrow$ (x.KnownBy = true) and conf(r_0) =0.91. It means if x is a doc item, the possibility x has been known by PDS owner is 0.91.

C. *Directory-based Rules*

Empirically, users prefer to classify personal data items with directory structure.

Definition 3.1: Access Ratio of Directory (ARoD). ARoD means the attention degree of a directory by a user. let D_0 be a directory of desktop, M be the number of items of D_0 known by u, and N be the number of total items in D_0, we denote $ARoD(D_0) = M/N$.

We analyzed the data item set marked by the 20 participants. The result is shown as Fig. 2(b), where X axis means the directories referred by the 16 participants, 750 directories are referred, and Y axis is the ARoD of the according directories. We can see, to most directories(More than 80%), ARoD equals 1, and about 15% directories' ARoD equals 0, and there are only less than 5% directories' which ARod are between 0 and 1. The phenomenon validates a rule: to a specific directory, a user has either accessed most items of it, or few items of it.

D. *Name-based Rules*

People have some common habits on naming personal data items. We studied personal data set of different people, and got the following observation.

Observation 3.1: Most Chinese users prefer to name a personal data item with a string including Chinese words.

We validate this observation by experiments. We compare two parameters SC and SCC for each user, where SC means the number of personal items named with Chinese words and SCC means the number of items which is named with Chinese words and is labeled with Yes for KnownBy. Fig.2(c) shows the results, we can see most items named with Chinese words are labeled known items. The ratio (SCC/SC) is about 93%. Therefore the conf of the rule (x.ChineseName=true)$\Rightarrow$ (x. KnownBy = Yes) is 0.93.

E. *Neighbored-Directory-based rules*

Empirically, the items in neighbored directories have similar KnownBy value. This observation means that the ARoD of a given directory is related to the ARoD of its neighbored directories. Let T_d be the tree of directory of desktop, D_1 and D_2 be two directories of T_d, $ARoD(D_1)$ is related to $ARod(D_2)$, and the weight of the relationship is based on the distance between D_1 and D_2. To validate it, we analyze the results of the experiments.

We let each participant label about 100 randomly-selected items with Yes(known) or No(unknown). In our experiments, we select all directories D, in each directory of D all selected items are marked with Yes. Then we can get another directory set D_P, which is made up of the parent directory of each directory of D. Let N_{DP} present the total number of items in D_P, and N_{DPC} mean the number of items of D_P which is marked with Yes. Therefore we can compute the access ratio (N_{DPC} / N_{DP}) based on its neighbored directories. Fig.2 (d) shows the average known-ratio is 83.4%, it means if all items in a directory are "known", the ratio the items in its parent directory are known is 83.4%.

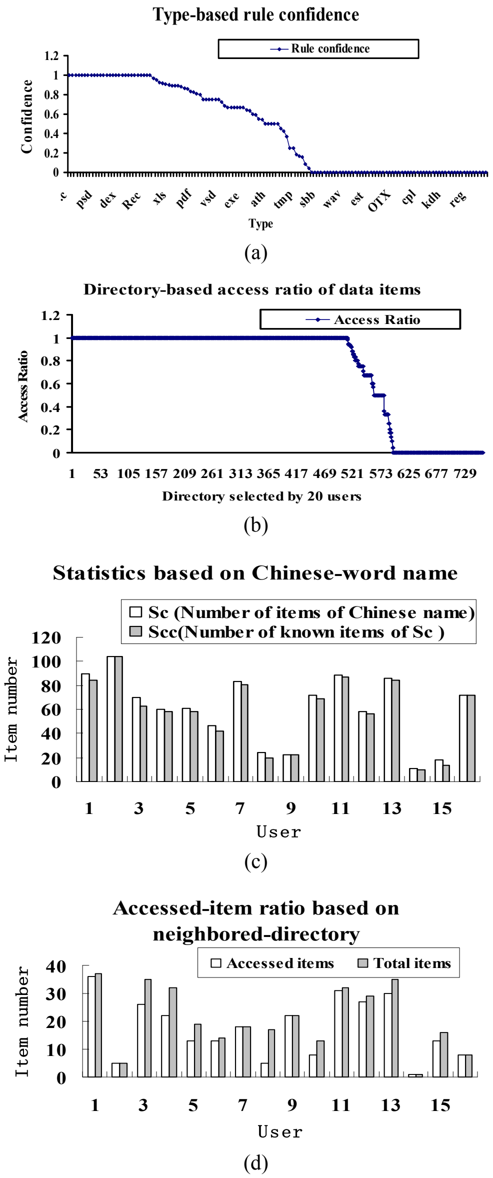


Figure 2. Illustration of rules based on single attributes

IV. ALGORITHMS

In section 3, we analyze the rules for identifying KnownBy feature of personal desk items. These rules can be divided into two classes: (1) content-based rules and (2) structure-based rules. Therefore we get two methods for identifying personal items:(1) Content-based Algorithm (CA), and (2) Content+ Structure based Algorithm(CSA).

A. *Content-based Algorithm*

Table I shows the specification of symbols used in CA algorithm. The input of this algorithm is a user U_0, an item

TABLE Ⅰ SPECIFICATION OF SYMBOLS

Symbol	Specification
S_{desk}	A set of items of personal desk
S_{acc}	A set of items accessed in recent period
x	A personal data item
$P_{type}(x)$	Type-based score of x.KnownBy
$P_{token}(x)$	Token-based score of x.KnownBy
$P_{CHI}(x)$	Name-based score of x.KnownBy
$P^{0}_{CHI}(x)$	Average name-based score
$P_{total}(x)$	Total score of x.KnownBy

set of desktop S_{desk}, and an item set accessed by U_0 in recent period S_{acc}. The output is PDS of U_0. In this algorithm, we considered three factors related to item content: type, semantic tokens and name. We firstly mine type set S_{type} and token set S_{token} preferred by U_0, then compute the type-based score($P_{type}(x)$), token-based score($P_{token}(x)$) and name-based score($P_{CHI}(x)$). Finally we compute the integrated score for deciding KnownBy attribute of the given item.

The process of the algorithm is described as below. We firstly summarize the type set preferred by U_0 by analyzing the S_{acc}, and present it as $S_{type}=\{\alpha_i\}$, where α_i is described as a 2-tuple (T_i, W_i), where T_i denotes type instance, and W_i means preference weight of U_0 to T_i. For example, ('doc',0.9) presents the preference degree of U_0 to doc type is 0.9. Let M_i be the number of the items of S_{acc} which type is T_i, N_i is the number of the items of S_{desk} which type is T_i, we compute $W_i = M_i/N_i$. $\forall x \in S_{desk}$, if $\exists t_i \in S_{type} \wedge t_i.type = x.type$, then $P_{type}(x) = t_i.weight$, else $P_{type}(x) = 0$.

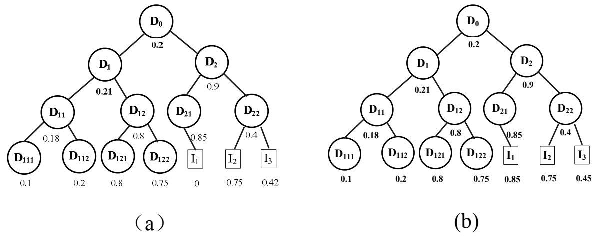
We get the token sets S_{token} by extracting tokens from the items of S_{acc}, and $S_{token} = \{e_i\}$, where e_i presents a token preferred by U_0. To improve the efficiency of algorithm, we only consider the name string of desktop files, it is because people prefer to name a personal item with some semantic tokens for recalling them easily in the future. Given an item x, we can get its tokens set T_x, by computing the similarity between T_x and S_{token}, we can compute $P_{token}(x)$ with the following formula:

$$P_{token}(x) = | T_x \cap S_{token} | / | T_x | \quad (1)$$

The finial content factor considered in this algorithm is Chinese words of name. We take P^{0}_{CHI} present the average confidence of the rule. According to experiments, $P^{0}_{CHI} \approx 0.93$. Based on these methods, given an item, we can compute its P_{type}, P_{token} and P_{CHI}. We take the following method to integrate the scores into a total score for deciding KnownBy attribute of the given item.

$$P_{total}(x) = \max\{P_{token}(x), P_{type}(x), P_{CHI}(x)\} \quad (2)$$

We illustrate the algorithm by an example. Let S_{type} = {(doc, 0.5), (ppt, 0.8)}, S_{token} = {dataspace, PIM, Survey}, and P_{Chi}= 0.91. If "D:\PDS\A survey on Dataspace Management.doc" is a given item x_0, then type(x_0) = 0.5, here T_{x0} ={survey, dataspace, management}, $S_{toke^{n}} \cap T_{x0}$ ={dataspace, survey}, $|T_{x0}|$= 3, so $P_{token}(x_0)$ = 2/3 = 0.67, $P_{Chi}(x_0)$ = 0, then $P_{total}(x_0)$ = max{0.5,0.67,0} = 0.67.



（a） (b)

Figure 3. Compute KnownBy based on directory structure

B. *Structure-based Algorithm*

According to our experiments, personal data items often distribute in neighbored directories of personal desktop. If directory structure is taken as a feature for computing KnownBy attribute, we can get a high recall. Therefore we propose an algorithm to compute KnownBy attribute based on content and structure (CSA algorithm), which including two steps. The first is to identify personal data items with a high precision by considering multiple content factors, including type and name features. Based on the results of the first step, we compute the preference degree of each directory of personal desktop, then we compute the KnownBy attribute for each item based on the preliminary results and the weighted directory structure. To identify the KnownBy feature, we add two attributes: NeighborWeight (NW) and PathWeight(PW), where NW means the weight based on the KnownBy attribute of the items in the same directory, PW means the weight based on the KnownBy features of each node of its path.

Definition 4.1: Interest Degree of Directory(IDD). Let D_0 be a given directory, and $\{I_i\}$ be the set of items of D_0, and W (I_i) be the weight computed for KnownBy feature of I_i with the content-based method。

We compute IDD for D_0 by the following formula.

$$IDD(D_0) = \frac{1}{n}\sum_{i=1}^{n} W(I_i) \quad （3$$

Based on the formula above, we compute IDD for each directory of personal desktop. Fig.3(a) shows the IDD value of each directory node in a directory tree example, and illustrates three items of directory D_{21} and D_{22}, and the KnownBy features of D_{21} and D_{22} are shown, which are computed by CA algorithm.

Definition 4.2: Logical distance of directory(LDD). Let Td= (D, E) be the directory tree of S_{desk} , where D is the set of directories of S_{desk}, E represents subfolder relationship. $\forall d_i, d_j \in Td.D$, LDD ($d_i$, d_j) equals to number of edges in the path from di to dj. For example, in figure 3(b), there are two edges between D_1 and D_2, then LDD(D_1, D_2) = 2; there are three edges between D_0 and D_{122}, then LDD(D_0, D_{122}) = 3.

Definition 4.3: Structure-based Weight(SW): Let I_0 be any item of desktop S_{desk}, x be an item and D_x be the directory of x, L_{path} = $\backslash D_1\backslash D_2 \backslash ...\backslash D_n$ be the path from root node of T_d to x, where D_i means a directory, n is the length of L_{path}. we compute SW as below:

$$SW(x) = \max\{IDD(D_i)/(LDD(D_x, D_i)+1)\} \tag{4}$$

Based on the weighted tree shown in Fig.3(b) and the preliminary KnownBy value of items, we can refine them with following formula: Ac=max{A_i, A_d, A_p}, where A_i means the initial KnownBy value by content-based method, Ad means the value of the IDD value of its directory, and Ap is path related weight for KnownBy. For example, D:\D_3\ D_2 \ D_1 \ x is an item of personal desktop, we can get LDD(x, D_1) = 1, LDD(x, D_2) = 2, LDD(x, D_3) = 3, if IDD(D_1) = 0.5, IDD(D_2) = 0.2, IDD(D_3) = 0.9, then A_p(x) = max{0.5/2, 0.2/3, 0.9/4} = {0.25, 0.07, 0.225}= 0.25.

Fig.3(b) illustrates the KnownBy value of I_2 and I_3, which are computed based on the directory structure and preliminary value. By the method, we can refine the results of the content-based algorithm.

V. EXPERIMENTS AND EVALUATION

In this section, we introduce the experimental dataset, design, and results.

A. *Data set and experimental design*

Currently there is no public data set for personal data management. To evaluate the effectiveness of our method on identifying boundary of personal dataspace, we develop a prototype system and select 15 users to run it in their desktops,and their PDS based on desktop are automatically built with our system. In our experiments, we take their desktop to simulate the public data space which includes his personal data items, and try to mark PDS boundary. Our experiments show that every participant has a recently-accessed file set in desktop, it shows our assumption on taking recent-accessed file set as training sample for mining rules to identify PDS boundary is reasonable. Table II shows the data set collected by us. The meaning of each column is specified as bellow.

S_{desk}: Number of files on personal desktop.

S_{rece} : Number of files in recent folder of desktop.

S_{sele} : Number of files randomly selected for evaluation.

S_{type} : Number of file types users selecte for evaluation.

S_{dir} : Number of directories users select for evaluation.

S_{acc}: Number of files marked as Known by participants

To make the experimental items more representative, we ask each user randomly select about 200 items from S_{desk} as testing sample, and each user is asked to select items referring more types and more directories as possible. In addition, we also ask each user to make S_{sele} include both known items and unknown items. In our work, We propose different methods to mark PDS boundary, like content-based method and directory structure based method. We design experiments for the different algorithms to test their efficiency.

TABLE II SPECIFICATION OF DATA SET

User	**S_{desk}**	**S_{rece}**	**S_{sele}**	**S_{type}**	**S_{dir}**	**S_{acc}**
U ser1	667123	156	186	35	56	166
U ser2	215678	369	184	32	93	131
U ser3	289712	135	200	34	71	147
U ser4	614569	301	133	36	84	97
U ser5	427172	183	180	36	30	154
U ser6	610529	994	173	23	29	114
U ser7	540981	154	204	40	79	135
U ser8	493622	24	215	5	3	24
U ser9	553612	162	185	34	57	111
U ser10	719880	220	201	35	46	186
U ser11	459053	261	198	33	37	141
U ser12	356783	154	204	40	79	135
U ser13	388980	268	101	24	12	83
U ser14	718906	158	179	39	30	45
U ser15	614569	190	387	36	45	190

B. *Evaluation Measures*

We take precision and recall as evaluation measures of our methods. In our experiments, we ask each participant to select a number of items S_{Sele} from his desktop randomly and label each of them with known or unknown, Based on which, we compute the precision and recall for each user based on different algorithms T_i. The formulas for

$$\Pr ecision(U_i, T_j) = \frac{|\{x \mid x \in S^i_{Acc}(U_i) \wedge x \in S^i_{Acc}(T_j)\}|}{|\{x \mid x \in S^i_{Acc}(U_i)\}|} \tag{5}$$

$$\mathrm{Re}\,call(U_i, T_j) = \frac{|\{x \mid x \in S^i_{Acc}(U_i) \wedge x \in S^i_{Acc}(T_j)\}|}{|\{x \mid x \in S^i_{Acc}(T_j)\}|} \tag{6}$$

computing the recall and precision are shown as blow:

Where U_i represents a user, T_j means a specific algorithm, $S^i_{Acc}(U_i)$ be the item set which is labeled known by U_i, and $S^i_{Acc}(T_j)$ means the item set which is labeled known with method T_j. In our experiments, we compute precision and recall for each participant with different algorithms. Furthermore, we can compute average recall ratio and precision ratio for each method T_j. Therefore we

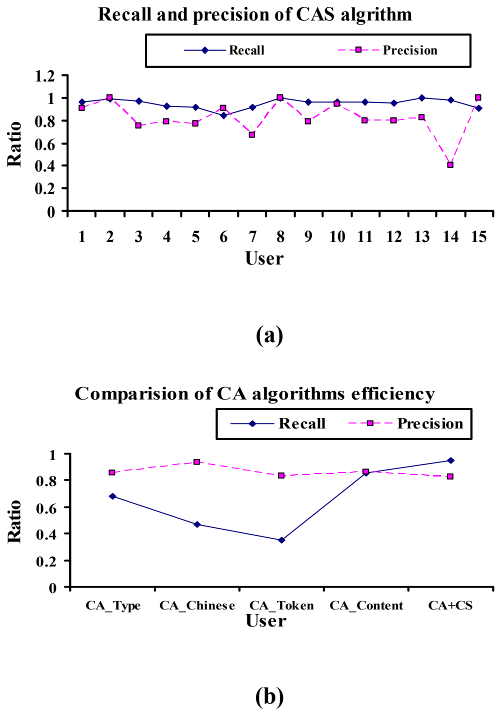


Figure 4. Experimental results

can analyze the efficiency of different methods for identifying PDS boundary.

C. *Experimental results*

We know the features influencing algorithms for identifying PDS boundary includes type, directory, name, structure of directory, and etc. To compare their different contribution to identifying PDS boundary, we evaluate the two methods by experiments: Content-based Algorithm (CA) and Structure based Algorithm(CSA). Based on the features of item content, we considered the following method. (1)CA_Type. It only considers the types interested by users. (2) CA_Chinese. It only considers if its name includes Chinese words. (3) CA_Token. It only considers the tokens interested by user. (4) CA_Content. It considers multiple content factors. (5) CA+CS(CSA). It considers both content factors and directory structure factors.

Fig.4 shows the results of our experiments. Fig.4(a) illustrates the recall and precision of CSA method of each participant. Fig.4(b) compares the recall and precision of each methods (CA and CSA). From the experiments we can see (1) CA_content has a good recall and precision, which are both 0.85, and (2) CSA shows a better efficiency than CA. The recall of CSA can reach 0.95, which is a great approving to CA method, but there is a little decline of precision, the average precision of CSA method is 0.83. Therefore CSA has a good efficiency totally.

VI. CONCLUSIONS

In this paper we propose an effective and efficient method to build initial PDS. The results show the combined method of CA and CS has a better efficiency. We have integrated the algorithms into our PDS prototype system OrientSpace, and they can be integrated into other PDS tools, such as desktop search and so on. In the future, we will integrate more personal data sources in our methods to construct more complete personal dataspace.

ACKNOWLEDGMENT

This research was partially supported by the grants from the Natural Science Foundation of China under grant number 60833005, 60573091; National High-Tech Research and Development Plan of China (No: 2007AA01Z155, 2009AA 011904)

REFERENCES

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

[2] A Halevy, M. Franklin, and D Maier, “Principles of Dataspace Systems”, in *Proc PODS*, 2006, pp. 1–9.

[3] P. Ogilvie and J. Callan, “Combining document representations for known-item search”, in *Proc ACM SIGIR*, 2003, pp. 143–150.

[4] C. Macdonald and I Ounis, “Combining fields in known-item email search”, in *Proc ACM SIGIR*, 2006, pp. 675–676.

[5] W. Jones and H Bruce, “A Report on the NSF-Sponsored Workshop on Personal Information Management”, PIM workshop2005.Available:http://pim.ischool.washington.edu

[6] X. Dong and A Halevy, “A Platform for Personal Information Management and Integration”, in *Proc ACM CIDR*, 2005, pp. 119–130.

[7] 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”, in Proc VLDB, 2007: 663-674.

[8] P.-A. Chirita, W. Nejdl, “Analyzing User Behavior to Rank Desktop Items”, in Proc SPIRE, 2006, pp. 86–97.

[9] C. Peery, W.Wang, A. Marian, T. D. Nguyen, “Multi-Dimensional Search for Personal Information Management Systems”. in Proc EDBT, 2008, pp. 464–475.

[10]T. Jaime, “How People Recall, Recognize, and Reuse Search Results”, ACM Transactions on Information Systems, Vol. 27, No. 1, Article 4, Publication date: December 2008.

[11]Y. Li, X. Meng, “Research on personal dataspace management”, in SIGMOD PhD Workshop(IDAR), 2008.