Evaluating Personal Information Management Using an Activity Logs Enriched Desktop Dataset

Sergey Chernov, Gianluca Demartini, Eelco Herder, Michał Kopycki, Wolfgang Nejdl

L3S Research Center, Leibniz Universität Hannover
Appelstr. 9a, 30167 Hannover
Germany
{chernov,demartini,herder,kopycki,nejdl}@L3S.de

ABSTRACT

The effective evaluation of Personal Information Management is a crucial problem for the research community. While evaluation methodologies for retrieval on the Web and in digital libraries are well-developed, the experiments with the advanced desktop tools are neither repeatable nor comparable. As privacy concerns do not allow to copy and distribute personal data outside the research lab, we suggest to overcome this problem by creation of desktop datasets within different research groups using a single methodology and a common set of tools. A dataset can include not only a static snapshot of the desktop documents, but also the logs of user activity on the desktop within several last months. We present the structure of the required dataset, a set of implemented tools and a sample dataset collected within the L3S Research Center.

ACM Classification Keywords

H.3.3 Information Storage and Retrieval: Information Search and Retrieval

General Terms

Experimentation, Design

Author Keywords

Evaluation, Desktop Dataset, User Activity Data

INTRODUCTION

The volume of data stored on a single hard drive and the amount of interactions with files and applications greatly increased in last years. Many Desktop search tools and systems for Personal Information Management (PIM) were released recently by main search engine vendors. The variety of PIM project calls for evaluation and comparison of proposed algorithms. As functionality of many PIM systems stems from area of information retrieval one can consider existing sound experimental methodologies, e.g. Cranfield methodology [8] or a method for evaluation of interactive systems [2]. The mainstream evaluation methodologies require an appropriate common test collection that is accepted by the community [16]. However, no such dataset is publicly available and testing algorithms on artificial datasets can be misleading. Without a reliable dataset, it is difficult to make a choice between any ranking algorithms, and results from different research groups become non-repeatable and incomparable.

Currently existing datasets came either from traditional digital libraries or from the Web data. The Desktop files are different from Web pages, since they usually do not contain explicit hyperlinks between documents. On the other hand, a lot of work in PIM is related to personalization and the collections from digital libraries cannot provide personalized user profiles. We also observe that the volume of unstructured information is gradually moving toward semi-structured representation, partially thanks to metadata annotation capabilities developed in state-of-art PIM systems [12]. For example, the address book contains different metadata fields for personal contacts, while the email messages can be searched by date, sender or title. This information should be present in the dataset too. Morover, the information need of the user searching her Desktop has a different focus than that on the Web. For example, people often seek for a previously known item on a Desktop, which makes the historical data rather important. These Desktop-specific features do not allow re-using existing datasets for PIM evaluation.

Highly personalized systems are designed using the information about the current Desktop content, but also take into account the current user's activities. It is very likely that users will highly benefit of "a system having knowledge of their specific tasks" [3]. A standard evaluation setup must incorporate and provide activity logs as well as data and metadata of the desktop items. As many desktop resources are accessed within some given activity context, one must be able to reconstruct these contexts in order to exploit them for information retrieval tasks, for example, using metadata annotations, file access timestamps, information about co-active items, etc. For such a reason we need to include in a Desktop evaluation collection history files (logs) of the activities performed by the Desktop user. A dataset satisfying these requirements will allow all the Desktop systems that make use of such information to be consistently evaluated and compared against each other.

[1] Aduna Aperture.
`http://www.aduna-software.com/technologies/aperture/`

[2] Beagle++ Project.
`http://beagle2.kbs.uni-hannover.de/`

The high privacy level of user files and data heterogeneity across multiple desktops makes it challenging to create a customized dataset for the PC Desktop environment - a *Desktop Dataset* (DD). We should address it already on the stage of data gathering. While some people are willing to share information with their close friends and colleagues, they do not want to disclose it to outsiders. In this case, there is a way to keep information available only for a small number of people within a single research group.

In this paper we present an approach we envision for generating such a DD. Our dataset includes *activity logs*, containing the history of each file or email. This DD provides a basis for designing and evaluating special-purpose retrieval algorithms for different Desktop search tasks. It extends our earlier work started with [4] towards a common DD based on real users' desktop information. After comparing our approach to similar ones, we present a possible DD design and ways for collecting the personal information. We describe a private test collection made of desktop data of 14 users. We also outline the discussion points for the future work.

RELATED WORK

The PIM field was recently developed within the information retrieval, database management, human-computer interaction and semantic Web communities. A number of interesting papers used Desktop data and/or activity logs for experimental evaluation. For example, in [15], authors used indexed Desktop resources (i.e., files, etc.) from 15 Microsoft employees of various professions with about 80 queries selected from their previous searches. In [13] Google search sessions of 10 computer science researchers have been logged for 6 months to gather a set of realistic search queries. Similarly, several papers from Yahoo [12], Microsoft [1] and Google [17] presented approaches to mining their search engine logs for personalization. In other papers [5] [6] the temporary experimental settings were used, which made these experiments neither repeatable nor comparable. We aim to provide a common Desktop specific dataset to this research community.

One open problem in the field of IR evaluation is to understand if the "queries in a test collection form an unbiased sample of a real search workload" [14]. A test collection that contains user query logs as well, like the one we propose here, can help in pushing forward this field of research. A different approach to evaluate PIM systems is the one adopted in the NEPOMUK project[3] where user scenarios, designed observing activities of real users, are the base for the creation of artificial data which are used for the evaluation of the PIM tools developed within the project. We believe that using artificial data is not sufficient in order to guarantee significant evaluation results. We hope that the our proposed approach will help this and other projects in the evaluation of their systems.

The good overview of the recent work in PIM evaluation and a new proposal for task-oriented evaluation is presented in [10]. Currently, we do not annotate the data with task descriptions as suggested, but it might be an interesting future extension. An evaluation dataset that needs to face privacy issues is the one provided by the MIREX initiative: a standardized dataset and evaluation framework to evaluate Music Information Retrieval systems and algorithms. The MIREX data sets cannot be redistributed due to copyright restrictions and then the organizers provide a service which allows "remote execution of black-box algorithms submitted by participants, and provides participants with real-time progress reports, debugging information, and evaluation results" [11]. The most related dataset creation effort is the TREC-2005/2007 Enterprise Track [4]. Enterprise search considers a user who searches the data of an organization in order to complete some task. The most relevant analogy between the Enterprise search and Desktop search is the variety of items of which the collection is composed (for example, in the TREC-2006 Enterprise Track collection e-mails, cvs logs, Web pages, wiki pages, and personal home pages are available). The most prominent difference between the two collections is the presence of *personal documents* and especially *activity logs* (e.g., resource read/write time stamps, etc.) within the DD.

DATASET DESIGN

Type of Information to Store

The data for aech DD can be collected among the participating users within a research groups. Several file formats should be stored: TXT, HTML, PDF, DOC, XLS, PPT, MP3 (tags only), JPG, GIF, and BMP. Each group locally collects several Desktop dumps, making use of logging tools for a number of applications like Acrobat Reader, MS Office family products, Internet Explorer, Mozilla Firefox and Thunderbird. We distinguish between *permanent information* which can be obtained during the one-pass indexing, and a *timeline information*, which has to be continuously logged. The desired permanent and timeline information is listed in Table 1. The part of this information which is already captured by our tools is described in details in the Section "Logging Tools".

Information Processing Tasks

One of the current issues is a consensus in the community on what set of tasks to be evaluated. Among possible information retrieval tasks we envision Ad Hoc retrieval, Folder Retrieval (i.e., ranking personal folders), and Known-Item Retrieval. The discussion is also open for Context Related Items Retrieval, both using example items or keyword queries, Information Filtering, Email Management and related tasks. It is also interesting what kind of advanced search criteria users need. As a starting point, we show some examples of simple search tasks.

Ad Hoc Retrieval Task

Ad hoc search is the classic type of text retrieval when the user believes that relevant information exists somewhere. Several documents can contain pieces of necessary data, but the user might not remember whether or where it has been

[3] http://nepomuk.semanticdesktop.org

[4] http://www.ins.cwi.nl/projects/trec-ent/

Permanent Metadata Information (indexing)	**Applied to**
URL	stored HTML files
Song Metadata tags*	MP3
Saved picture's URL and saving time*	Graphic files
Path Annotation$^+$	All Files
Scientific Publications$^+$	PDF Files
Publication Bibliography Data$^+$	BibTeX Files
Web Cache$^+$	Web History
Emails and attachments$^+$	emails
Timeline information (logging)	
Time of being in focus	All applications
Time of being opened	All applications and files
Path of the file being edited	MS Office files and PDF
Being printed	Thunderbird, Firefox
Text selections from the clipboard*	Text pieces within a file
Time of Conversation with Someone (Chat client)	Skype, MSN Messenger
Browsers actions: bookmark, clicked link, typed URL	Web Pages
Bookmarking Actions (creations, modifications, deletions)	Firefox
Google Web Search queries	Firefox, IE
IP address*	User's Desktop
Metadata of emails being in focus	Thunderbird, Outlook
Adding/editing an entry in calendar and tasks*	Outlook

Table 1. Permanent and Timeline Logged Information provided by indexing and logging operations. We denote with $*$ the not yet implemented features. We denote with $+$ the features provided by the Beagle++ indexing system as example.

stored, and might not be not sure which keywords are best to find them.

Known-Item Retrieval Task

Targeted or known-item search task is the most common for the Desktop environment. Here the user wants to find a specific document on the Desktop, but does not know where it is stored or what is its exact title. This document can be an email or a working paper. The task considers that the user has some knowledge about the context in which the document has been used before. Possible additional query fields are time period, location, and a topical description of the task in which scope the document had been used.

Folder Retrieval Task

Many users have their personal items topically organized in folders. At some point, they may search not for a specific document, but for a group of documents in order to use it later as a whole - browse them manually, reorganize or send to a colleague. The retrieval system should be able to estimate the relevance of folders and sub-folders using simple keyword queries.

Queries

As we aim at real world tasks and data, we want to reuse real queries from Desktop users. As every Desktop is a unique set of information, its user should be directly involved in both query development and relevance assessment. Therefore, Desktop contributors should be ready to give 10 queries selected from their everyday tasks. Their participation in relevance assessment solves the problem of subjective query evaluation, since users know best their information needs.

In this setting each query is designed for the collection of a single user. However, some more general scenarios can be designed as well, such as finding relevant documents in every considered Desktop. One could envisage the test collection as partitioned in sub-collections that represent single Desktops with their own queries and relevance assessments. This solution would be closely related to the MrX collection used in the TREC SPAM Track, which is formed by a set of emails of an unknown person.

The query can have the following format:

- <num> KIS01 < /num>
- <query> Eleonet project deliverable June< /query>
- <metadataquery> date:June topic:Eleonet project type: deliverable < /metadataquery>
- <taskdescription>I am combining a new deliverable for the Eleonet project.< /taskdescription>
- <narrative> I am looking for the Eleonet project deliverable, I remember that the main contribution to this document has been done in June. < /narrative>

We included the <metadataquery> field, to enable the specification of semi-structured parameters like metadata field names, in order to narrow down the query. The set of possible metadata fields would be defined after collecting the Desktop data.

The Desktop contributors must be able to assess pooled documents 6 months after they contributed their Desktop. Eeach query is supplemented with the description of context (e.g., clicked/opened documents in the respective query session), to allow users to provide relevance judgments according to the actual context of the query. As users know their documents very well, the assessment phase should go faster than usual TREC assessments. For the task of known-item search, the assessments are quite easy, since only one (at most several duplicates) document is considered relevant. For the adhoc search task we expect users to spend about 3-4 hours to do relevance assessment per query.

The Goal: Standard Evaluation Approaches

With a DD, built in the way here described, it is possible to perform IR evaluation experiments. Researchers can build and use their own DDs, which are not publicly redistributed. As they all are similarly structured, the evaluation results - although they stem from different real data - are comparable and, as we define it, "*soft-repeatable*".

Even when semantic information (e.g., RDF annotations, Activities, etc.) is integrated as part of a search system, the traditional measures from information retrieval theory can and should still be applied when evaluating system performance. This allows the use of the same set of metrics in the evaluation of Desktop IR systems, to make the results comparable among different systems.

LOGGING TOOLS

Implicit Feedback Approach

In our proposal for collecting usage data, we decided to use Implicit Feedback. This approach was exploited in [7] and proved to be a representative indication of user interests. We acquire activity data automatically by using logging software, which does not require explicit user input. User interaction with the Desktop is being monitored without interrupting her workflow. The lack of direct user input is compensated by the amount and granularity of the automatically acquired data.

User Activity

User actions are articulated through the interaction with different applications. In Windows XP, this interaction is expressed by handling windows, which are the visual representation of an application. For example, the window that is currently in focus, is the window that the user is currently looking at (presumably working with). By observing user's actions on windows, we examine the actual activity that the user is performing on the Desktop. However, one window can act on several resources (for example, all emails in one instance of a rich email client or several Web pages viewed in an Internet browser). In these cases, we extend the logging activity to monitor interaction with these resources. The main advantage of this approach is that the context of accessing the resource or application is being logged. This information could be used to extract missing links between Desktop objects.

Implementation

Our Logging Framework is presented in Figure 1. As we wanted to keep the logging process as generic as possible, we have developed a system-wide logging utility, the *User Activity Logger*. Although this approach gives an overview of the entire interaction between the user and the Desktop, the acquired information presents only basic description of user activity. The in-depth information is gathered by extensions to the applications that we want to log. Such an extension, which is part of the application itself, has direct access to resources involved in user activities. The description of the resource enriches the description of an activity - and the other way round: the resource description is enriched by the actions that the user is performing on it. For example, the User Activity Logger receives a notification that Outlook 2003 is currently being used and the Outlook 2003 plug-in retrieves detailed information about emails being currently processed by the user. Another example: the Firefox plug-in indicates that since 5 minutes the user was looking at a particular Web page; however, based on data from the User Activity Logger, we know that the system is actually in idle time.

This architecture is highly extensible. One can download our framework and write a customized plug-in to explore the user activity of interest. To this end, we opened the development to those willing to participate via a *SourceForge* project [5].

Our main contribution to logging utilities is the User Activity Logger. Once installed, it uses Windows Hooks to intercept every "activate", "create" and "destroy" window notification (pop-up windows, invisible windows and dialog boxes are considered irrelevant and filtered out). For each notification, a generic activity description is being extracted. For some of the applications, the Logger acquires additional information that describes the resource displayed in the window. For example, for Word text editor or Adobe Acrobat Reader, the file path of the currently viewed file is stored; for Internet Explorer, the URL of the Web page currently viewed; for Outlook Express, the currently selected email message. Table 2 describes the information being logged by the User Activity Logger. Currently, the Windows XP version of the logger prototype is available for download at the Personal Activity Track Web page [6].

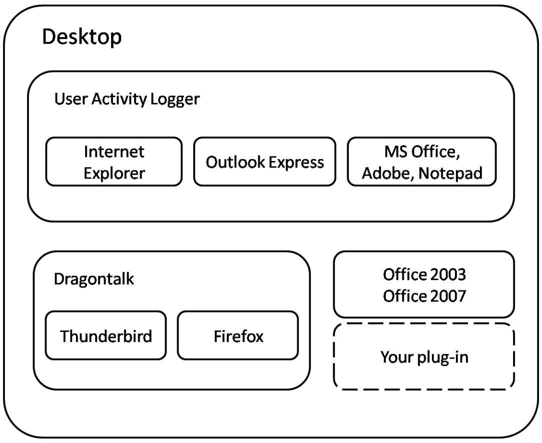


Figure 1. Logging Framework

Generic information	Applied to
Operation type (created, activated, destroyed)	All applications
Timestamp	All applications
Unique window handle	All applications
Application exe name	All applications
Window caption	All applications
Resource specific Information	
File path to resource being viewed	MS Office products, Adobe Acrobat Reader, Notepad
URL	Internet Explorer
Sender, recipients, received date, sent date	Outlook Express

Table 2. Generic and resource specific data collected by the logger

Collecting detailed resource information from User Activity Logger level is possible for a limited number of applications. For other relevant applications, we developed or adapted existing plug-ins. The plug-ins store resource and activity information every time a notification has been triggered by the user. We have implemented such plug-ins for Outlook 2003 and Outlook 2007. By using Visual Studio Tools for Office technology [7], which allows to write extensions for MS Office Family products, we were able to collect in-depth email usage data. Data collected by Outlook plug-ins is described in Table 3.

[5] `http://sourceforge.net/projects/activity-logger/`

[6] `http://pas.kbs.uni-hannover.de/`

[7] `http://msdn2.microsoft.com/en-us/office/aa905543.aspx`

Data description	Applied to
Operation type	Outlook, Thunderbird
Timestamp	Outlook, Thunderbird
Unique email ID	Outlook, Thunderbird
Path to the email in the email folder hierarchy	Outlook, Thunderbird
Subject	Outlook, Thunderbird
Sender (name and email adress)	Outlook, Thunderbird
Recipients (name and email adress)	Outlook, Thunderbird
Cc recipients (name and email adress)	Thunderbird
Bcc recipients (name and email adress)	Thunderbird
Address book entry	Thunderbird

Table 3. Email data collected by the Outlook 2003 and 2007 and Thunderbird plug-ins

For applications from the Mozilla family, we have used an already existing solution and adapted it to our requirements. Dragontalk project [8] provides extensions to the Thunderbird rich email client and Firefox Internet browser. The extensions allow monitoring of user interaction with both applications. Our adaptation of Dragontalk included changing the outputting method, extending the functionality by supporting new notifications, and adding methods to preserve user privacy. See Table 3 for a description of the data collected from Thunderbird.

Information Representation and Storage

Table 4 presents the full list of notifications that are currently supported by the framework. For each notification, additional data from Table 3 and 2 is extracted and stored.

Supported user actions	Supported Applications
General	
Window actions (create, activate, destroy)	All applications
Documents	
Document actions (open, activate, close)	MS Office, Adobe Acrobat Reader, text file editors like Notepad, TextPad, Notepad++, etc.
Web	
Navigate to URL (click, type in)	Internet Explorer, Firefox
Tab (create, change, close)	Internet Explorer, Firefox
Bookmark (create, modify, delete)	Firefox
Forward, backward, reload, home	Firefox
Print page	Firefox
Submit Web form	Firefox
Submit Google Web search query	Internet Explorer, Firefox
Email	
Email actions (select, sent)	Outlook 2003, Outlook 2007, Outlook Express, Thunderbird
Email actions (receive, reply, forward, delete, move, print)	Thunderbird
Address book entry (create, modify, delete)	Thunderbird
Email Folder (create, modify, delete)	Thunderbird
Instant Messengers	
Conversation (start, activate, finish)	Skype, MSN Messenger
System state	
Idle time (start, end)	System event
Hibernation (start, end)	System event
Framework state	
Logger actions (activate, deactivate)	User Activity Logger

Table 4. Types of notification supported by the Logging Framework

Collected data is stored in a simple human-readable format in text files located directly on user's computer. As different parts of the Logging Framework focus on user interaction with different resources, the format and granularity of output data differ as well. For example, a single notification intercepted by the User Activity Logger (e.g. Firefox window activated), may imply several notifications from the Dragontalk Firefox logger (switching between Web pages without leaving the Firefox window). For this reason, we decided to keep a separate log file for each component of the framework. As a result, in the current implementation, the user can have up to four log files (User Activity Logger, Thunderbird, Firefox, and Outlook 2003 and 2007). However, the simplicity of the format allows to parse it to any other format. In the scope of cooperation with the NEPOMUK project [9] we translated our output format into NEPOMUK Ontologies [10] by using a readable RDF syntax, called Notation3 [11].

Privacy Issues

Obviously, each logging utility introduces some privacy issues. The collected data is very sensitive and exposes user interaction with the whole desktop. Our main consideration was to protect the data from unauthorized access. Because all the data is stored directly on the user's computer in plain text files in human-readable format, it is up to the user to decide to whom and in what form the data should be released.

In the Logging Framework we preserve the user's privacy by offering means to stop or pause the logging process. The user can pause the process or simply shut down the logging utility via a user-friendly menu (Figure 2).

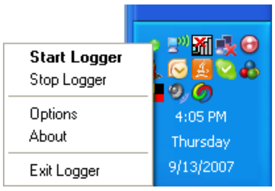


Figure 2. Tray icon and menus provide control over the logging process

However, the goal of monitoring the user activity is to collect as much data as possible. Therefore, we introduced other means that only restrict the logging range without terminating the process itself. Figure 3 presents two dialog boxes that allow the user to specify which Web domains should be excluded from the logging process. Once specified, the utilities will ignore any notifications involving these resources.

Future Directions

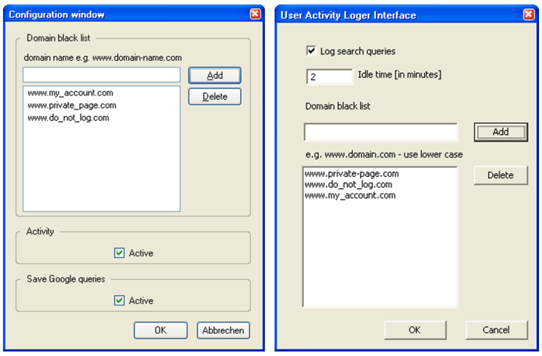
The framework's architecture is extensible, which means that one only needs to concentrate on developing new plug-ins to gather more precise information about user actions. Currently, the prototype of MS Office plug-in is in a testing phase. The plug-in extends the notifications involving file resources accessed via MS Office applications.

[8] http://dragontalk.opendfki.de/

[9] http://nepomuk.semanticdesktop.org

[10] http://www.semanticdesktop.org/ontologies/

[11] http://www.w3.org/DesignIssues/Notation3



Configuration window

Domain black list

domain name e.g. www.domain-name.com

www.my_account.com
www.private_page.com
www.do_not_log.com

Activity

☑ Active

Save Google queries

☑ Active

User Activity Loger Interface

☑ Log search queries

2 Idle time [in minutes]

Domain black list

e.g. www.domain.com - use lower case

www.private-page.com
www.do_not_log.com
www.my_account.com

Figure 3. Menus restricting the range of the logging process by specifying pages that should be excluded from the logging process. Firefox (left) and Internet Explorer (right)

As the User Activity Logger covers the whole desktop, it is directly bounded to the system architecture. As an implication, it is not portable between operating systems. Addressing larger groups of users requires porting the User Activity Logger to other platforms like Windows Vista or Linux distributions.

We also plan further extensive cooperation with the NEPOMUK project to exploit the capabilities of implicit feedback as user interest indication.

OUR EXPERIENCE: GATHERING DATA FROM USERS

In this section, we describe the approach taken by our group in order to build a personal information search test collection. For evaluating the retrieval effectiveness of a personal information retrieval system, a test collection that accurately represents the desktop characteristics is needed. However, given highly personal data that users usually have on their desktops, currently there are no desktop data collections publicly available. So we created for experimental purposes our internal desktop data collection.

The collection that we created - and which are currently using for evaluation experiments - is composed of data gathered from the PCs of 14 different users. The participant pool consists of PhD students, PostDocs and Professors in our research group. The data has been collected from the desktop contents present on the users' PCs in November 2006. For this reason the data and the activity logs collected are mainly referred to the year 2006.

Each data provider is allowed to use the entire collection for research experiments. We observed that only a subset of providers are actually experimenters but, in any case, all the providers must sign a written agreement as they gain the access to the collection.

Privacy Preservation

In order to face the privacy issues related to providing our personal data to other people, a written agreement has been signed by each of the 14 providers of data, metadata and activities. The document is written with implication that every data contributor is also a possible experimenter. The text is reported in the following:

> **L3S Desktop Data Collection**
> **Privacy Guarantees**
>
> - I will not redistribute the data you provided me to people outside L3S. Anybody from L3S whom I give access to the data will be required to sign this privacy statement.
> - The data you provided me will be automatically processed. I will not look at it manually (e.g. reading the emails from a specific person). During the experiment, if I want to look at one specific data item or a group of files/data items, I will ask permission to the owner of the data to look at it. In this context, if I discover possibly sensitive data items, I will remove them from the collection.
> - Permissions of all files and directories will be set such that only the *l3s-experiments-group* and the super-user has access to these files, and that all those will be required to sign this privacy statement.

Currently Available Data

The desktop items that we gathered from our 14 colleagues, include emails (sent and received), publications (saved from email attachments, saved from the Web, authored / co-authored), address books and calendar appointments. A distribution of the desktop items collected from each user can be seen in table 5:

User#	Emails	Publications	Addressbooks	Calendars
1	109	0	1	0
2	12456	0	0	0
3	4532	1054	1	1
4	834	237	0	0
5	3890	261	1	0
6	2013	112	0	0
7	218	28	0	0
8	222	95	1	0
9	0	274	1	1
10	1035	31	1	0
11	1116	157	1	0
12	1767	2799	0	0
13	1168	686	0	0
14	49	452	0	0
Total	29409	6186	7	2
Avg	2101	442	0.5	0.1

Table 5. Resource distribution over the users.

A total number of 48,068 desktop items (some of the users provided a dump of their desktop data, including all kinds of documents, not just emails, publications, address books or calendars) has been collected, representing 8.1GB of data. On average, each user provided 3,433 items.

In order to emulate a standard test collection, all participants provided a set of queries that reflects typical activities they would perform on their DDs. In addition, each user was invited to contribute their activity logs, related to the period until the point at which the data were provided.

All participants defined their *own* queries, related to their activities, and performed search over the reduced images of their desktops, as mentioned above.

The queries sets are composed as follows. Each user has been asked to provide two clear keyword queries (single or multiple keywords), two ambiguous keyword queries (single or multiple keywords), two only-metadata queries (e.g. "from:smith"), and two metadata and keyword queries (e.g. "information retrieval author:smith"). In total, 88 queries were collected from 11 users. The average query length was 1.77 keywords for the clear queries, 1.27 for the ambiguous queries, and 1.65 for the metadata queries. As expected, the ambiguous queries are shorter than the clear queries, which are in 73% of the case composed of a single term. These results are comparable to the average of 1.7 keywords, as reported in other larger scale studies (see for example [9]).

In order to collect also some ground truth data, we asked the data providers to manually assess the relevance of some search results. For every query and every system (we used 3 different ranking algorithms), each participant rated the top 5 output results on a Likert scale (from 0 to 4, with 4 being very relevant for the query and 0 without any connection to the query).

FUTURE WORK AND CONCLUSIONS

There are several important questions that are not solved yet and that require an additional discussion within the community. In this concluding section, we list some of the issues that we consider most important.

- **Data and Privacy**. It is difficult to select appropriate data to build a testbed collection for experiments with personalization. There are several issues to be investigated, including: (1) privacy implications and data anonymization, (2) storage and accessibility of test data, (3) information sources (here, one of our major interests goes toward analyzing and discussing the logging of personal activities). The discussion should also consider the personal data privacy problem both at the stage of data gathering and the stage of document relevance assessment. What makes a good collection and what is the best way to interact with it? How should the collection be composed? Which information to include in the personal application activity logs? How to manage the privacy issues for the sharing the data?
- **Loggers and Test Applications**. This aspect is more focused on how we can collect necessary data and what kind of technical infrastructure should be implemented for PIM evaluation. Among other questions, we investigate which logging tools are already available, how they can be reused for PIM evaluation and which experimental setup from existing evaluation initiatives can be adopted.
- **Measurement and Relevance Assessments**. Finally, a query format and the relevance metrics should be discussed. While there are already a plethora of metrics, do we need more novel measures or can we adopt existing ones? We should agree on how relevance assessments should be performed. It would be interesting to formalize the user benefit from the PIM systems usage.

The creation of a testbed for experiments with personalized search is more challenging task than creating a Web search or XML retrieval dataset, as it is highly complicated by privacy concerns. This paper describes the ongoing work toward a common DD based on users' desktop information. Here we presented a possible DD design and means for collecting the personal information. Further, we outlined the discussion points for the future work and discussion within the IR community.

Our main goal is the promotion of the usage and development of tools (e.g. the Activity Logger) that can help the PIM research community to create a standardized approach to the evaluation of PIM systems.

ACKNOWLEDGMENTS

We would like to acknowledge the large number of people involved in the development of the tools and the contribution to the collections: Pavel Serdyukov, Paul-Alexandru Chirita, Julien Gaugaz, Sven Schwarz, Leo Sauermann, Enrico Minack, Raluca Paiu, and Stefania Costache. We also want to thank all those who contributed their desktop data. This work was partially supported by the Nepomuk project, funded by the European Commission under the 6th Framework Programme (IST Contract No. 027705).

REFERENCES

1. E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, New York, NY, USA, 2006. ACM Press.
2. P. Borland and P. Ingwersen. The development of a method for the evaluation of interactive information retrieval systems. *Journal of Documentation*, 53(3):225–250, 1997.
3. T. Catarci, B. Habegger, and A. Poggi. Intelligent user task oriented systems. In *Proceedings of the Workshop on Personal Information Management held at the 29th ACM International SIGIR Conf. on Research and Development in Information Retrieval*, 2006.
4. S. Chernov, P. Serdyukov, P.-A. Chirita, G. Demartini, and W. Nejdl. Building a desktop search test-bed. In *ECIR '07: Proceedings of the 29th European Conference on Information Retrieval*, pages 686–690, 2007.
5. P.-A. Chirita, S. Costache, W. Nejdl, and R. Paiu. Beagle^{++}: Semantically enhanced searching and ranking on the desktop. In *ESWC*, pages 348–362, 2006.

6. P. A. Chirita, J. Gaugaz, S. Costache, and W. Nejdl. Desktop context detection using implicit feedback. In *In Proceedings of the Workshop on Personal Information Management held at the 29th ACM International SIGIR Conf. on Research and Development in Information Retrieval*. ACM Press, 2006.

7. M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, pages 33–40, New York, NY, USA, 2001. ACM.

8. C. Cleverdon. The cranfield tests on index language devices. In *Readings in information retrieval*, pages 47–59, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

9. S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff i've seen: A system for personal information retrieval and re-use. In *SIGIR*, 2003.

10. D. Elsweiler and I. Ruthven. Towards task-based personal information management evaluations. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 23–30, New York, NY, USA, 2007. ACM Press.

11. M. C. Jones, M. Bay, J. S. Downie, and A. F. Ehmann. A "do-it-yourself" evaluation service for music information retrieval systems. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, page 913, 2007.

12. R. Kraft, C. C. Chang, F. Maghoul, and R. Kumar. Searching with context. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 477–486, New York, NY, USA, 2006. ACM Press.

13. F. Qiu and J. Cho. Automatic identification of user interest for personalized search. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 727–736, New York, NY, USA, 2006. ACM Press.

14. T. Rowlands, D. Hawking, and R. Sankaranarayana. Workload sampling for enterprise search evaluation. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 887–888, 2007.

15. J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456, New York, NY, USA, 2005. ACM Press.

16. E. Voorhees. The philosophy of information retrieval evaluation. In *Proc. of the 2nd Workshop of the Cross-Language Evaluation Forum (CLEF)*, 2001.

17. B. Yang and G. Jeh. Retroactive answering of search queries. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 457–466, New York, NY, USA, 2006. ACM Press.