Building a Desktop Search Test-bed

Sergey Chernov[1], Pavel Serdyukov[2],
Paul-Alexandru Chirita[1], Gianluca Demartini[1], and Wolfgang Nejdl[1]

[1] L3S / University of Hannover, Appelstr. 9a D-30167 Hannover, Germany
[2] Twente Database Group, MISSING 7500 AE Enschede, The Netherlands
{chernov, chirita, demartini, nejdl}@l3s.de,
serdyukovpv@cs.utwente.nl

Abstract. In the last years several top-quality papers utilized temporary Desktop data and/or browsing activity logs for experimental evaluation. Building a common testbed for the Personal Information Management community is thus becoming an indispensable task. In this paper we present a possible dataset design and discuss the means to create it.

1 Introduction

In the last years several top-quality papers utilized Desktop data and / or activity logs for experimental evaluation. For example, in [4], the 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 [3] 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 [2], Microsoft [1] and Google [5] presented approaches to mining their search engine logs for personalization. We want to provide a common public Desktop specific dataset for this research community.

The most related dataset creation effort is the TREC-2006 Enterprise Track [3]. Enterprise search considers a user who searches the data of an organisation in order to complete some task. The most relevant analogy between the Enterprise search and Desktop Search is the variety of items which compose the collection (e.g., in the TREC-2006 Enterprise Track collection e-mails, cvs logs, web pages, wiki pages, and personal home pages are available). The biggest 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 Desktop dataset.

In this paper we present an approach we envision for generating such a Desktop dataset. We plan our new dataset to include *activity logs* containing the history of each file, email or clipboard usage. This dataset will bring a basis for designing and evaluating of special-purpose retrieval algorithms for different Desktop search tasks.

2 Dataset Design

File Formats and Metadata. The data for the Desktop dataset will be collected among the participanting research groups. We are going to store several file formats: TXT,

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

Application	File Format
Acrobat Reader	PDF files
MS Word	DOC, TXT, RTF
MS Excel	XSL
MS Powerpoint	PPT
MS Explorer	HTML
MS Outlook	PST
Mozilla Firefox	HTML
Mozilla Thunderbird	MSF and empty extension, mbox format

Table 1. Logged Application and File Formats

Permanent Information	**Applied to**
URL	HTML
Author	All files
Recipients	Email messages
Metadata tags	MP3, WMA
Has/is attachment	Emails and attachments
Saved picture's URL and saving time	Graphic files
Timeline information	
Time of being in focus	All files
Time of being opened	All files
Being edited	All files
History of moving/renaming	All files
Request type: bookmark, clicked link, typed URL	HTML
Adding/editing an entry in calendar and tasks	Outlook Journal
Being printed	All files
Search queries in Google/MSN Search/Yahoo!/etc.	Explorer/Firefox search fields
Clicked links	URL
Text selections from the clipboard	Text pieces within a file and the filename
Bookmarking time	Explorer/Firefox bookmarks
Chat client properties	Status, contact's statuses, sent filenames and links
Running applications	Task queue
IP address	User's address and addresses user connects to
Email status	Change between received/read

Table 2. Timeline and Permanent Logged Information

HTML, PDF, DOC, XLS, PPT, MP3 (tags only), JPG, GIF, and BMP. Then, each group willing to test its system would submit 1-2 Desktop dumps, using logging tools for a number of applications listed in the Table 1.

The set of logged applications can be extended in the future. Loggers save the information which we describe in Table 2.

Data Gathering. As the privacy issue is very important here, we propose two options for possible information gathering.

1. Optimistic approach. We assume there are volunteers ready to contribute some of their personal information to the community, given that this information would be redistributed only among a restricted group of participants. As a test case, we gave two laptops to students for half a year. They were able to use them for free, but the condition was that all the information on these laptops will be available for future research. They were also warned not to store highly private information like passwords or credit card numbers. As this approach worked well, we expect that all participating groups will find similar reasonable incentives to attract more volunteers.

2. Pessimistic approach. While some people are ready to share information with their close friends and colleagues, they do not like to disclose it to outsiders. For this

case, there is a way to keep information available only for a small number of people: Personal data is collected from participating groups by some coordinators and preprocessed into the publicly available uniform XML format. Every group can adapt its search prototypes to this format and submit binary files to the coordinators. Runs are then produced locally by a coordinator and results are sent back to the participants. This way, only trusted coordinators have access to the actual documents, while it is possible for all participants to evaluate their results. Similar schemes has been tested in TREC Spam Track, and it might be a necessary solution for future TREC tracks as well, whenever they involve highly private data (i.e. medical, legal, etc.).

3 Relevance Assessments and Evaluation

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

In this setting queries are designed for the collection of a single user, but some more general scenarios can be designed as well, for example finding relevant documents in every considered Desktop. It is thus possible to see the test collection as partitioned in sub-collections that represent single Desktops with their own queries and relevance assessments. This solution would be very 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:

- <top>
- <num> KIS01 < /num>
- <query> Eleonet project deliverable June< /query>
- <metadataquery>date:June topic:Eleonet project type:deliverable< /metadataquery>
- <taskdescription>I am combining 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>
- < /top>

We include the <metadataquery> field so that one could specify 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 the Desktop. Moreover, each query will be supplemented with the description of context (e.g., clicked / opened documents in the respective query session), so that users could 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 normal 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.

4 Proposed Tasks

1. AdHoc Retrieval Task. Ad hoc search is the classic type of text retrieval when the user believes she has relevant information somewhere. Several documents can contain pieces of necessary data, but she does not remember whether or where she stored them, and she is not sure which keywords are best to find them.

2. 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, a working paper, etc. 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, topical description of the task in which scope the document had been used, etc.

3. Folder Retrieval Task. It is very popular among users to have their personal items topically organized in folders. Later 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.

5 Conclusion

Building a Desktop IR testbed seems to be more challenging than creating a Web Search or an XML Retrieval dataset. In this paper we presented the concrete parameters for defining the features of such a Desktop Dataset and discussed the possible means for creating it, as well as utilizing it for algorithm assessments.

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. 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.
3. 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.
4. 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.
5. 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.