Knowledge-Based Systems 131 (2017) 1–9

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A PDF document re-finding system with a Q&A wizard interface

Gangli Liu[a,*], Baihui Jiang[b], Ling Feng[a]

[a] Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
[b] School of Law, Tsinghua University, Beijing, 100084, China

ARTICLE INFO

Article history:
Received 3 January 2017
Revised 24 May 2017
Accepted 29 May 2017
Available online 30 May 2017

Communicated by Prof. H. Fujita

Keywords:
Re-finding
Information retrieval
User interface

ABSTRACT

Re-finding electronic documents from personal computers is a frequent demand. For a simple re-finding task, people can use many methods to retrieve the target document, such as navigating directly to its folder, searching with desktop search engines, or checking the Recent Files List. When encountering difficult re-finding tasks, people usually cannot rely on attributes exploited by conventional re-finding methods, the re-finding would fail. We propose a new method to support difficult re-finding tasks, by collecting extra new attributes of a document, such as number of pages, number of images, reading frequency, and coverage percentage. If the document is quested later, the collected attributes and experiences are used to filter out it. A question and answer wizard interface is utilized to alleviate cognitive burden when recollecting. Finally, we develop a PDF document re-finding system to evaluate the effectiveness of the method.

1. Introduction

As computers have been an indispensable tool, re-finding a document which has been accessed previously becomes a common task [1]. Some of the re-finding tasks are effortless to accomplish, especially when the user is familiar with the target file or it has been accessed recently. For a simple re-finding task, a variety of methods can be selected to accomplish it. Such as navigating directly to the folder which contains the target document. If a user cannot recall a document's path but remembers its keywords or other meta-data attributes such as size, creation time, author etc., a desktop search engine can be used to retrieve it. If the document is still "warm", checking the "Recent file list" maintained by the operating system might be a better choice.

However, if a user's memory about the quested document is obscure; no effective attributes (i.e., path, keywords, meta-data etc.) are recollected to locate the document, it is a difficult re-finding task. Typically, none of the methods listed above can deal with these re-finding tasks because of poorly recollected memory. We argue that maybe the user can recall some other attributes which have not been exploited by existing file retrieval systems or methods. E.g., supposing a user is re-finding a document, he cannot remember its path and keywords, but clearly remembers it has hundreds of pages; it contains at least two images and no math equations; he only read the first chapter of it for less than an hour; he might have accessed it one or two times, but definitely less than five times; he has printed some pages of it. Apparently, these information fragments are detailed enough to locate the document without much effort. However, most of them are not collected and exploited by existing document re-finding systems. Studies of recollection for texts and stories indicate that people are not good at remembering precise details. Instead what tends to be remembered are high-level meanings or gists like information of the example [2].

If plenty of these gist attributes are recalled, even though none of them are detailed enough to locate the quested document separately, the combined group effort of them can be effective tools to achieve the re-finding task. If each of these high-level meaning information can exclude 1/3 of irrelevant documents, 16 attributes can filter a candidate set of 10,000 into about 15. Within 15 candidates, it will not be strenuous to find out the target with just scan and recognition. If the user can recall a more specific value about a particular attribute, fewer filters are required.

1.1. The three stages of memory

Human memory works in a very complex and elusive way [3,4], even today scientists still do not fully understand how humans remember or recall exactly . Most psychologists believe that the process of memory begins with encoding. It allows the perceived item of interest to be converted into a construct that can be stored within the brain, and then recalled later from short-term or long-term memory. There are two parts of the brain, the hippocampus

[*] Corresponding author.
E-mail address: gl-liu13@mails.tsinghua.edu.cn (G. Liu).

http://dx.doi.org/10.1016/j.knosys.2017.05.028
0950-7051/© 2017 Elsevier B.V. All rights reserved.

and the front cortex, are responsible for analyzing various sensory inputs and decide if they are worth remembering [5].

According to the multi-store model [6], people's memory can be stored in three stages: the sensory stage, the short-term stage, and ultimately, the long-term stage. Each stage of human memory functions like a filter which discards useless information. The sensory memory can only last a fraction of a second, the short-term memory can only hold about seven items for 20 to 30 s at a time. Only those information which is of interest to the subject can be gradually transferred into long-term memory.

Therefore, in a difficult re-finding scenario, only those attributes stored in long-term memory can be used to locate the target, since sensory memory and short-term memory can only be held for less than one minute.

1.2. Other possible attributes for re-finding

A document can be characterized by many attributes, such as file name, size, keywords, and authors etc. The user's experiences about a document can also be used to locate the target, such as printing experiences, reading it at an unusual place or time etc. For an attribute to be an effective filter, not only it should be recollected by the user, but also there exists a logging system which has recorded the attribute values for all the documents the user has accessed. Since the re-finding system needs to compare the value of the quested document with values of others. There is uncertainty which set of attributes will get into a user's long-term memory, and can be recollected when the re-finding occurs [7]. It depends on many circumstances, such as the user's interest, characteristics etc.

We propose to use a logging system to record other possible attributes when a person is processing a document, such as a printing experience, an unusual time or location, and some conclusive gists like cumulative processing time of the document etc.; preparing for that any set of attributes would be used as filters to locate the document later. The quantity of documents a user can access during a life span is limited, so the total recorded information will not be huge.

1.3. Alleviating cognitive burden

To recollect many kinds of information about a document would bring the user considerable cognitive burden. According to findings of cognitive psychology, when people are trying to recollect something, a great deal of their memory is *"available but not accessible"* [8,9]. For example, if a user is asked to recollect attributes of a document he has read, typically, little information can be recollected because a large portion of the information is *"available but not accessible"*. If some hints are provided, such as a list of questions about the document, more information can be recollected since it relieves the subject's cognitive burden. Therefore, a question and answer wizard interface is utilized to alleviate users' cognitive burden when the recollecting occurs.

A logging system is developed to record attributes of documents and the user's reading experiences about them. Some attributes cannot be extracted directly, such as how long and how much the user has read a document. Therefore, an analyzing system is used to calculate these values.

To further alleviate the user's cognitive burden, recommendations are provided for each question. A recommendation is generated by analyzing the distribution of attribute values. As mentioned above, in a difficult re-finding context, the user's memory about the target document is usually some high-level gists, so the recommendations are generalized to a gist about the document. For example, one of the questions asks the user:

"How many tables does the document contain?"

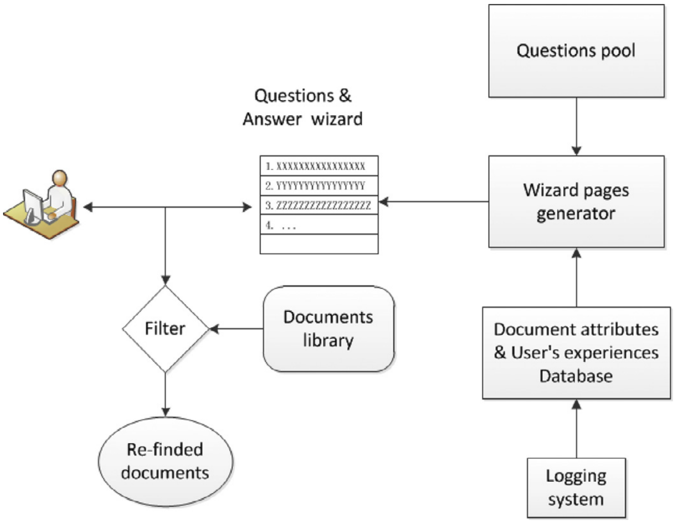


Fig. 1. The framework of the method.

Instead of recommending specific numbers like 2, 3 or 10, a generalized answer is recommended, such as:

"A. More than 3 B. Less than or equal to 3"

Fig. 1 shows the framework of the method for re-finding. In Section 3, a PDF (Portable Document Format) re-finding system is developed based on the method. Details of introducing new attributes and providing recommendations will be discussed there. A user study is carried out to evaluate the system in Section 4.

2. Related work

Personal Information Management (PIM) systems help people acquire, store, organize and re-find information [10]. Re-finding is a significant segment of PIM systems. It is a frequent demand in people's daily lives. McKenzie and Cockburn show 60–80% of web page visitations are re-access [11]. 40% of web searches are to re-find a piece of information [12]. Re-visitation of emails is less common compared to web pages and documents, Elsweiler et al. show that 6–15% of emails will be retrieved later [13].

There are two psychological processes involved in a re-finding process, recall-directed search and recognition-based scanning [14]. People pay careful attention when naming a file, primarily not for using the name to search, but for recognizing the file when it appears in a list of candidates [15]. In a desktop search process, file names are frequently used in the recall-directed search process too.

Bergman et al. note navigating through a file hierarchy is the most common method people employ to re-find their desktop documents [16]. However, if the target is an email or web-page, keyword searching is preferred [13,17]. For desktop re-finding, navigation has a significant advantage, it can feedback some cues while the user is navigating between file folders [18]. These cues can help retrieve more memories about the target file, such as file name, keywords, and their relationship with other files. This implies that re-finding tasks are not independent, a previous re-finding experience can affect later re-finding behaviors, even though they are for different targets. Navigation makes scan and recognition easier because files contained in the same folder usually share some common characteristics, such as belonging to the same project or being different versions of a same document. These advantages make navigation the most popular method for re-finding. When the user's memory about the target document is misty or erroneous, navigation is an unsatisfactory choice for re-finding; the user will get *'lost'* in the folder hierarchies, viewing same folders or even same files multiple times [13].

"Faceted search" [19–21] employs a similar navigation paradigm, which allows users to navigate based on file attribute values, such as file type and modification time. It has a similar advantage with navigation; the user may obtain some useful cues while navigating. However, if there are many sub-categories for each "facet", or the path to the target is deep, it will introduce enormous cognitive burden for the user to reach the target. A Q&A system alleviates the user's cognitive burden. The system recommends the sequence of the attributes and a concise categorization of the attribute values. The categorization is based on the user's personalized distribution of the attribute values. With the recommendations, the user can focus his energy on recollecting effective memories and making necessary inferences for locating the target.

Besides navigation, many re-finding systems and methods have been developed to help people re-find documents, web pages, or emails etc. Deng et al. devised a *'ReFinder'* system which utilizes the user's previous access context (such as location, access time, background computer procedures etc.) to re-find a web page [22]. It takes memory degradation into consideration and tackles it with a corresponding degrading context tree. Qvarfordt et al. designed a *'SearchPanel'* Chrome browser extension which records a user's process history information of the search result page (SERP), provides sense-making, navigation, and re-finding functions during a complex information demand [23]. Chau et al. developed a *'Feldspar'* system [24] which can retrieve personal files by associations related to other objects, such as email, person, folder or event etc. *"Stuff I've Seen"* is a system which helps people re-find their files, it integrates web pages, emails and documents into a centralized system, illustrates that time and associated people are important retrieval cues to assist re-finding [25]. A further study shows that segmenting the time-line by some landmark events about the user can improve re-finding efficiency [26].

A document can be in digital or printed form. Trullemans et al. devised a system called "PimVis", which can re-find both digital and paper documents in a cross-media information space; it facilitates a unified organization of digital and paper documents through the creation of bidirectional links between the digital and physical information space [27]. Sadeghi et al. propose a model to predict whether a web search activity is re-finding information, it also estimates the difficulty level of the re-finding task [28]. As Social Media (SM) has become a valuable information source, re-finding SM content is of value to users. Meier and Elsweiler conducted a naturalistic, log-based study of user interaction with Twitter, the result shows remembered people are used as a stepping stone to Tweets rather than searching for content directly [29].

A variety of attributes and strategies have been exploited to re-find information in all kinds of data sources. The system presented in this paper proposes a new strategy for re-finding. Instead of passively waiting the user to provide characteristics of the quested document, it actively asks the user for information that may help locate the document. In addition, several new characteristics are exploited to filter out the target document, such as "Coverage percentage", "Access frequency" and "cumulative process time". Due to the personalized characteristics of re-finding behavior and the interferences between re-finding tasks [18], it is almost impossible to make a rigorous and convincing comparison of performance between these systems and methods.

3. A PDF document re-finding system

A PDF document re-finding system is developed based on the framework of Fig. 1. PDF documents are very popular to knowledge workers. Re-finding PDF documents like papers or electronic books is a common activity for knowledge workers.

The re-finding system is comprised of two parts: the analyzing and logging subsystem and the Q&A wizard subsystem. The analyzing and logging subsystem records a document's various attributes and the user's reading experiences about it. The Q&A wizard subsystem provides an interface to fulfill the user's document re-finding demands.

3.1. The analyzing and logging subsystem

To implement the analyzing and logging subsystem, a plug-in procedure for the Adobe Acrobat Reader application is developed. At present the plug-in only supports Microsoft Windows operating system. When the user is reading PDF documents, the plug-in records a variety of attributes it collected into a MySQL database. Attributes being utilized as filters are classified into two categories: intrinsic document attributes and the user's experiences about the documents; in each category, they are subdivided into categorical and numeric according to the characteristics of their values. Attributes like file path and file name are also collected, but not used as filters. Because in a difficult re-finding scenario, the user is assumed to have forgotten these attributes.

3.1.1. Intrinsic attributes

Intrinsic attributes are attributes possessed intrinsically by a document, they will not change when the user accesses a document in read-only mode, such as file size, title, authors, number of pages, number of images etc. If a document has been edited by the user, some of intrinsic attributes may change. A revised document is deemed as a different version of the original document. Table 1 shows the intrinsic attributes being collected now, new items can be extended in future. Attributes tagged with an asterisk are new ones that have not been used by other re-finding systems.

"File size" and "File acquisition date" are extracted through APIs (Application Programming Interface) provided by the operating system. "File creation date", pages, and authors are obtained through APIs provided by Adobe. "FCD" and "FAD" only count the *Year* of the date, omitting the *Month* and *Day* information. If the author information cannot be extracted from the meta-data segment, document content between the title and abstract is analyzed to collect information about the authors. For some documents, no information about its authors can be detected; its "Number of authors" is recorded as "NULL". By checking existence of a line entitled "Bibliography" or "References", attribute values of Bibliography are obtained.

A simplified strategy is utilized to collect attributes like "Number of images", "Number of tables", and "Number of formulae"; they are collected by analyzing document content. Since if there are images, tables or formulae contained in a document, usually there are some particular phrases like "Fig. *** illustrates", "Table *** shows", or "Formula ***" referenced in the content of the document. By comparing and counting the number of distinct phrases of this kind, the values of these attributes are collected. Further research will attempt to locate images, tables, and formulae by differentiating distinct formats of these elements.

3.1.2. User experience attributes

User experience attributes are attributes which involve the user's processing history of a document. Table 2 shows user experience attributes being collected currently, new items can be added in future if available.

"Printing experience" and "Last access time" are obtained through APIs provided by Adobe. "Re-finding experience" is recorded by the system. "LAT" is classified into four categories: "Within a week", "Within a month", "Within a year", and "Older

Table 1
Intrinsic Document Attributes.

Attri. name	Abbr.	Notes	New
File size	FS		
File creation date	FCD	when did the authors create the document?	
File acquisition date	FAD	when did the user obtain the document?	
Pages		Number of pages the document contains	*
Number of authors	Authors		*
Number of images	Images		*
Number of tables	Tables		*
Number of formulae	Formulae		*
Bibliography	Bib	Does it contain a bibliography?	*

Table 2
The User Experience Attributes.

Attri. name	Abbr.	Notes	New
Printing experience	PE	Whether printed it previously	
Re-finding experience	RE	Whether retrieved it previously	
Unusual access time	UAT	Whether accessed it at an unusual time	*
Unusual access location	UAL	Whether accessed it at an unusual place	*
Access frequency	AF	How many times has the user accessed it?	*
Cumulative process time	CPT	How much time has been spent on it?	*
Coverage percentage	Coverage	How much of it has been read?	*
Last access time	LAT		

than a year". An "Unusual access time" is discovered by analyzing patterns of the user's reading time. Each day is roughly divided into four parts: morning (5 am to 12 pm), afternoon (12 pm to 5 pm), evening (5 pm to 9 pm) and night (9 pm to 4 am). For example, if the system discovers a user seldom reads documents at Saturday morning, if an event of Saturday morning reading occurs, it is categorized as an "Unusual access time". "Unusual access location" is calculated similarly. The reading location is discriminated by checking the IP address of the computer. If the access location is a city that is different from where the user lives, the location is categorized as an "Unusual access location".

3.1.3. Cumulative processing time

The user may have read a document many times. The cumulative processing time is calculated as the total time of all the reading experiences. It is common that a user is reading a document while doing other things, such as programming. Simply counting the interval between document close and open time will introduce some invalid processing time. Therefore, in the implementation, when the user starts up the Adobe Acrobat Reader application, the plug-in initiates a thread which periodically checks whether the Reader application is the foreground window; if the Reader application is not the foreground window, the timer stops.

It is possible that the user has left while the monitor continues showing a document. To detect this situation, the program periodically checks the user's mouse and keyboard input activities. If there are no inputs within a threshold (ten minutes), it is assumed that the user has left, then the timer stops.

If there are multiple copies or versions of the same document, and the user has spent some time on each of them, the cumulative processing time of the document cluster is characterized as the sum of each document's cumulative processing time. Other copies or versions of a PDF document are recognized by comparing the "DocumentID" and "InstanceID" attributes of the documents, these two attributes can be extracted through APIs provided by Adobe.

3.1.4. Document process coverage

The user might have a memory of how much he has read a document. For example, he does not have enough time to read a paper completely; he just reads the abstract of it. This information can be exploited to help re-find the document.

To calculate coverage percentage, reading time of each page is recorded. If the reading time is greater than a threshold (three minutes), the page is assumed having been covered. Coverage percentage of a document is calculated as the ratio of pages having been covered to the total pages of the document. Recording a user's reading time of each page can also help him quickly locate the desired information. Suppose the user has retrieved a document, by checking the recorded page processing time, the set of pages the user has read intensively can be retrieved efficiently

3.1.5. Access frequency

Usually the user cannot remember exactly how many times he has accessed a document. However, if the question is about a general impression, like *"less than 5 times"* or *"more than 5 times"* , typically it is not an arduous question to answer. Therefore, access frequency is recorded as a re-finding attribute.

If the user accidentally opened a document and found it was not the intended one, then closed it immediately. This access is not counted as a valid reading experience when calculating access frequency. A timer is set to record a document's open and close time, if their interval is less than a threshold (i.e., one minute), access frequency of the document is not updated. If a document has many copies or versions, their access frequencies are summed up.

3.2. The Q&A wizard subsystem

The Q&A wizard subsystem provides an interface to collect the user's memory pieces about the target document, and shows the retrieved results to the user at the end.

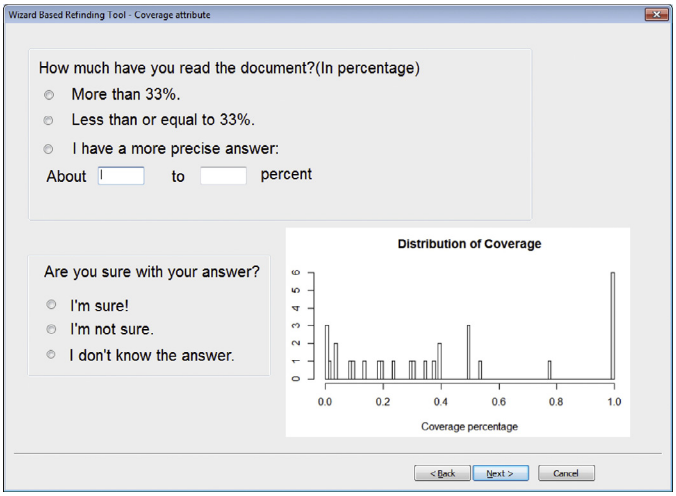
3.2.1. The user interface

At each step of the wizard, the user is asked a question about an attribute, such as *"How many pages does the document have?"*

To alleviate the user's cognitive burden, recommendations are provided for each question, such as:

"A. More than 10 pages B. Less than or equal to 10 pages"

Parameters of the recommendations are calculated by analyzing the distribution of the values, the mean value is utilized as the diving line currently. Other quantiles such as the median value can be alternatives. The analyzing and logging subsystem has recorded



Wizard Based Refinding Tool - Coverage attribute

How much have you read the document?(In percentage)

○ More than 33%.

○ Less than or equal to 33%.

○ I have a more precise answer:

About [ ] to [ ] percent

Are you sure with your answer?

○ I'm sure!

○ I'm not sure.

○ I don't know the answer.

Fig. 2. One of the pages of the wizard.

these values for each document. The user is provided an option to give a more precise answer than the recommendations. It is possible that the user has a clear memory about a particular attribute due to some special reasons. For example, the user noticed a document contains two images showing two terrifying monsters, which were very impressive. This experience might leave a solid memory for him. If the document is quested later, the question *"How many images does the document contain?"* will get a precise answer. With the accurate answer, a lot of ineligible documents can be excluded.

For a difficult re-finding task, it is common that the user is not sure about his answer to a question. To collect this information of confidence level, three options are provided to let the user describe how confident he is about the answer:

1.*"I'm sure!"* 2.*"I'm not sure."* 3.*"I don't know the answer."*

After the memory collecting phase, attributes on which the user has chosen the first option are used with priority; If these attributes are not enough to filter out the candidate set to less than a threshold (20 documents), attributes on which the user has chosen the second option will be used. If the third option is selected, that attribute is ignored during the re-finding process.

To further help the user make a correct decision, a picture illustrating the global distribution of the attribute values is provided as a reference. The values are fetched from the database and plotted by a plotting function. Fig. 2 shows an example page of the wizard during a re-finding process.

3.2.2. *Ranking the results*

Users tend to examine search results from top to bottom. Many algorithms have been developed to rank the results. Such as the well-known PageRank algorithm [30], which works by counting the number and quality of links to a web page. The series of the Okapi BM25 algorithms rank a set of documents by matching the query terms with the terms appearing in each document [31]. Since the PDF document re-finding system does not use links and keywords to re-find a document, algorithms like PageRank and BM25 are incapable for sorting the results. The attributes used in the system are primarily utilized to exclude irrelevant documents, not much information is provided to rank the results. The system sorts the results by comparing the user's *Familiarity Degree* on the target document and each candidate document. Because there is a tendency that the more time the user spends on a document, and nearer to last accessing of it, the more familiar the user would be to the document. On the other hand, there is also a tendency that the more familiar the user is on the target document, the more confident answers there will be, and the less time will be cost to answer a question.

Equation $X_i = C_i/E_i$ is used to estimate the user's "Familiarity Degree on a Candidate Document" (FDCD), C_i denotes the document's cumulative process time; E_i is time elapsed since last access of it. The user's "Familiarity Degree on the Target Document" (FDTD) is assessed by equation $Y_t = (2A_c + A_u)/T_a$. A_c is number of confident answers ; A_u is number of uncertain answers ; T_a denotes the average time of answering a question. There are other methods for calculating the familiarity degrees, such as substituting exponential decay for the reciprocal of the elapsed time. To balance the differences of different calculating methods, the familiarity degrees are firstly normalized in Algorithm 1, by subtracting the mean value and then divided by the standard deviation. Eventually the candidates are ranked by comparing their normalized familiarity degrees with the target. The *Familiarity Distance* is calculated as the absolute value of their difference, then the candidate documents are ranked by their *Familiarity Distance* from the target. The smaller the distance, the higher it is ranked. The mean value and standard deviation of FDTD are obtained by training the system with a group of artificially generated re-finding tasks. Since the system do not utilize keywords to re-find, the title or first sentence is used to hint the documents to be retrieved.

Algorithm 1 An algorithm to rank the candidate documents.

```
Input:
    A set of candidate documents $D_n$; each candidate document's FDCD $X_i$; the target document's FDTD $Y_t$; mean value and deviation of FDCD, $\mu_X$ and $\sigma_X$; mean value and deviation of FDTD, $\mu_Y$ and $\sigma_Y$.
Output:
    The sequence of candidate documents, $D_n$;
Normalize FDCD as: $F_i = \frac{X_i - \mu_X}{\sigma_X}$;
Normalize FDTD as: $F_t = \frac{Y_t - \mu_Y}{\sigma_Y}$;
for each document $i \in D_n$ do
    Calculate Familiarity Distance as: $d_i = abs(F_i - F_t)$;
end for
Sort $D_n$ according to $d_i$ in ascending order;
return $D_n$;
```

It is worthy mentioning that re-finding a personal document is quite different from searching the Internet. The search space is much smaller, usually thousands of documents; after filtering with the Q&A wizard, the result space is also smaller, typically tens or hundreds of documents. Under such a situation, scan and recognition usually play a more significant role than a ranking algorithm. To help the user locate the target more efficiently, thumbnails of the most intensively processed pages of a document can be showed in the results.

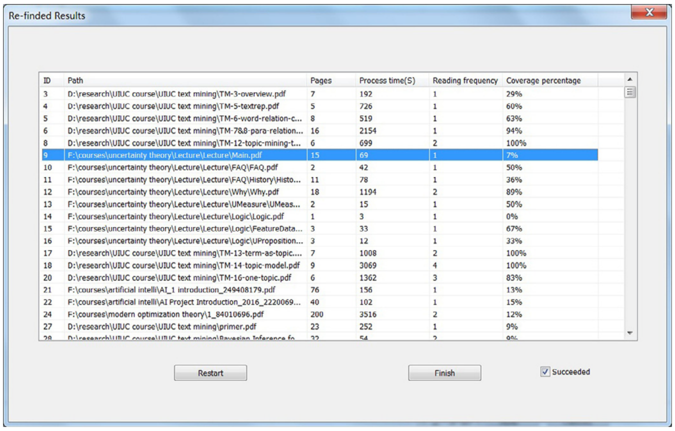
Fig. 3 shows the retrieved results. To keep the neatness of the interface, only six attributes are displayed at present; they are document ID, path, number of pages, cumulative process time (seconds), access frequency, and coverage percentage. Other attributes can be enclosed if necessary. The user can click the "Restart" button to start a new attempt, or the "Finish" button to end the re-finding session. The "Succeeded" check box is for counting success rate during the user study of Section 4.

4. User study

A user study is conducted to evaluate the performance of the PDF document re-finding system under difficult re-finding circumstances. We recruited 20 participants to test the system, 12 males and 8 females, aged from 22 to 28. To collect a relatively comprehensive document reading history, all the participants are selected

Table 3
Collected data during the first phase.

	Doc	FS	Pages	Authors	Images	Tables	Formu.	AF	CPT	Cover.
P1	747	615	58	2.6	8	7	12	2.1	56	28%
P2	1148	1426	22	4.1	13	3	5	1.4	49	34%
P3	1953	738	7	2.8	9	12	8	1.2	13	22%
P4	968	1053	48	3.7	12	2	2	1.3	26	29%
P5	1867	899	17	3.5	3	7	10	1.6	35	40%
P6	1947	849	55	3.1	14	13	1	1.5	8	57%
P7	801	1282	23	5.4	4	5	11	1.1	32	61%
P8	1690	794	56	2.9	11	8	1	1.7	18	36%
P9	786	820	6	4.3	12	14	14	2.8	35	45%
P10	1919	954	14	3.6	16	7	8	1.9	16	25%
P11	699	494	21	4.1	4	5	23	1.1	35	48%
P12	1876	876	38	3.7	6	4	6	3.4	25	41%
P13	1841	1165	43	3.2	7	2	9	2.2	34	39%
P14	631	912	39	3.9	2	10	3	1.9	29	34%
P15	876	629	21	4.5	10	8	12	1.6	50	75%
P16	1393	163	32	3.9	8	5	2	2.2	38	34%
P17	1941	1100	21	2.9	16	4	6	1.1	36	17%
P18	2820	825	32	2.8	3	3	13	1.3	3	4%
P19	1711	292	21	3.4	8	9	7	1.9	34	51%
P20	977	858	11	3.8	7	6	4	1.5	48	83%



ID	Path	Pages	Process time(S)	Reading frequency	Coverage percentage
3	D:\research\UIUC course\UIUC text mining\TM-3-overview.pdf	7	192	1	29%
4	D:\research\UIUC course\UIUC text mining\TM-5-textrep.pdf	5	726	1	60%
5	D:\research\UIUC course\UIUC text mining\TM-6-word-relation-c...	8	519	1	63%
6	D:\research\UIUC course\UIUC text mining\TM-7&8-para-relation...	16	2154	1	94%
8	D:\research\UIUC course\UIUC text mining\TM-12-topic-mining-t...	6	699	2	100%
9	F:\courses\uncertainty theory\Lecture\Lecture\Main.pdf	15	69	1	7%
10	F:\courses\uncertainty theory\Lecture\Lecture\FAQ\FAQ.pdf	2	42	1	50%
11	F:\courses\uncertainty theory\Lecture\Lecture\FAQ\History\Histo...	11	78	1	36%
12	F:\courses\uncertainty theory\Lecture\Lecture\Why\Why.pdf	18	1194	2	89%
13	F:\courses\uncertainty theory\Lecture\Lecture\UMeasure\UMeas...	2	15	1	50%
14	F:\courses\uncertainty theory\Lecture\Lecture\Logic\Logic.pdf	1	3	1	0%
15	F:\courses\uncertainty theory\Lecture\Lecture\Logic\FeatureData...	3	33	1	67%
16	F:\courses\uncertainty theory\Lecture\Lecture\Logic\UProposition...	3	12	1	33%
17	D:\research\UIUC course\UIUC text mining\TM-13-term-as-topic....	7	1008	2	100%
18	D:\research\UIUC course\UIUC text mining\TM-14-topic-model.pdf	9	3069	4	100%
20	D:\research\UIUC course\UIUC text mining\TM-16-one-topic.pdf	6	1362	3	83%
21	F:\courses\artificial intelli\AI_1 introduction_249408179.pdf	76	156	1	13%
22	F:\courses\artificial intelli\AI Project Introduction_2016_2220069...	40	102	1	15%
24	F:\courses\modern optimization theory\1_84010696.pdf	200	3516	2	12%
27	D:\research\UIUC course\UIUC text mining\primer.pdf	23	252	1	9%
28	D:\research\UIUC course\UIUC text mining\Bayesian Inference fo...	32	54	2	9%

Fig. 3. The retrieved results.

as junior postgraduate students from several adjacent campuses. As postgraduate students, they read PDF documents regularly; as junior students, there are less PDF reading experiences before the startup of the user study, therefore, less interference of previous reading activities is introduced than senior students. The participants are from different majors, including computer science(7), chemistry(4), medicine(3), architecture(3), and others(3).

4.1. Collecting attribute values

The user study was conducted in two phases. In the first phase, the plug-in was installed on the participants' computers, collected their document reading activities for about ten months. During the time, no operation was required for the participants, all information collecting work was done by the plug-in procedure automatically. Table 3 shows part of collected data during the first phase, except for the first and second columns, all other columns are average values of the attributes. The "Doc" column is the quantities of documents collected during the experiment, other attributes are described in Tables 1 and 2. The unit of "FS" is *KB*; the unit of "CPT" is *Minutes*. Fig. 4 shows distributions of three attributes ("Pages", "Images", and "CPT") . Outliers that exceed the maximum value of Y axis are not showed. In order to save space, distributions of other attributes are not presented.

4.2. A diary study

In the second phase, a diary study of document re-finding tasks was conducted for about one month. The participants are required to keep a diary of their PDF document re-finding activities during the time. To alleviate their burden of manually recording information and encourage their enthusiasm of participating, for easy re-finding tasks, they are only required to record its occurrence and time of duration.

4.2.1. Definition of difficult re-finding tasks

To evaluate the performance of the PDF document re-finding system under difficult re-finding circumstances, it is necessary to discriminate difficult re-finding tasks from normal ones. If a participant has a demand to re-find a document, he first starts a timer, and then chooses any of the three conventional methods mentioned in Section 1 (navigation, desktop search, checking the "Recent file list") to re-find it; if the document cannot be located within ten minutes, the task is defined as a difficult re-finding task. Once a difficult re-finding task is recognized, the participants are required to utilize our re-finding system to accomplish it. Multiple attempts are allowed during each re-finding session.

4.2.2. Results of the diary study

Table 4 shows the results of the one-month diary study of re-finding. "Tasks" denotes the number of re-finding tasks occurred during the time; "Diff." means the number of difficult tasks; "SR" stands for "Success Rate". If a participant can retrieve a document within 15 min with the system, it is defined as a successful re-finding. SR is the proportion of succeeded tasks, it can be seen 52% of difficult tasks can be attained with the system on average; "Round" denotes the average rounds attempted to fulfill the succeeded tasks; "Time" denotes the average time (minutes) spent on retrieving them. In addition, the participants are asked whether the Q&A wizard interface is helpful during their recollecting, 90% of them give an affirmative answer.

4.2.3. Evaluation of the attributes

It is useful to know which set of attributes are inclined to be recollected confidently, hence serving as good filtering attributes. To test the participants' memory about the attributes, their selections of confidence levels about the answers are counted and graded. Questions with confident answers implying good memory, are scored 2; questions ignored indicate obscure memory,

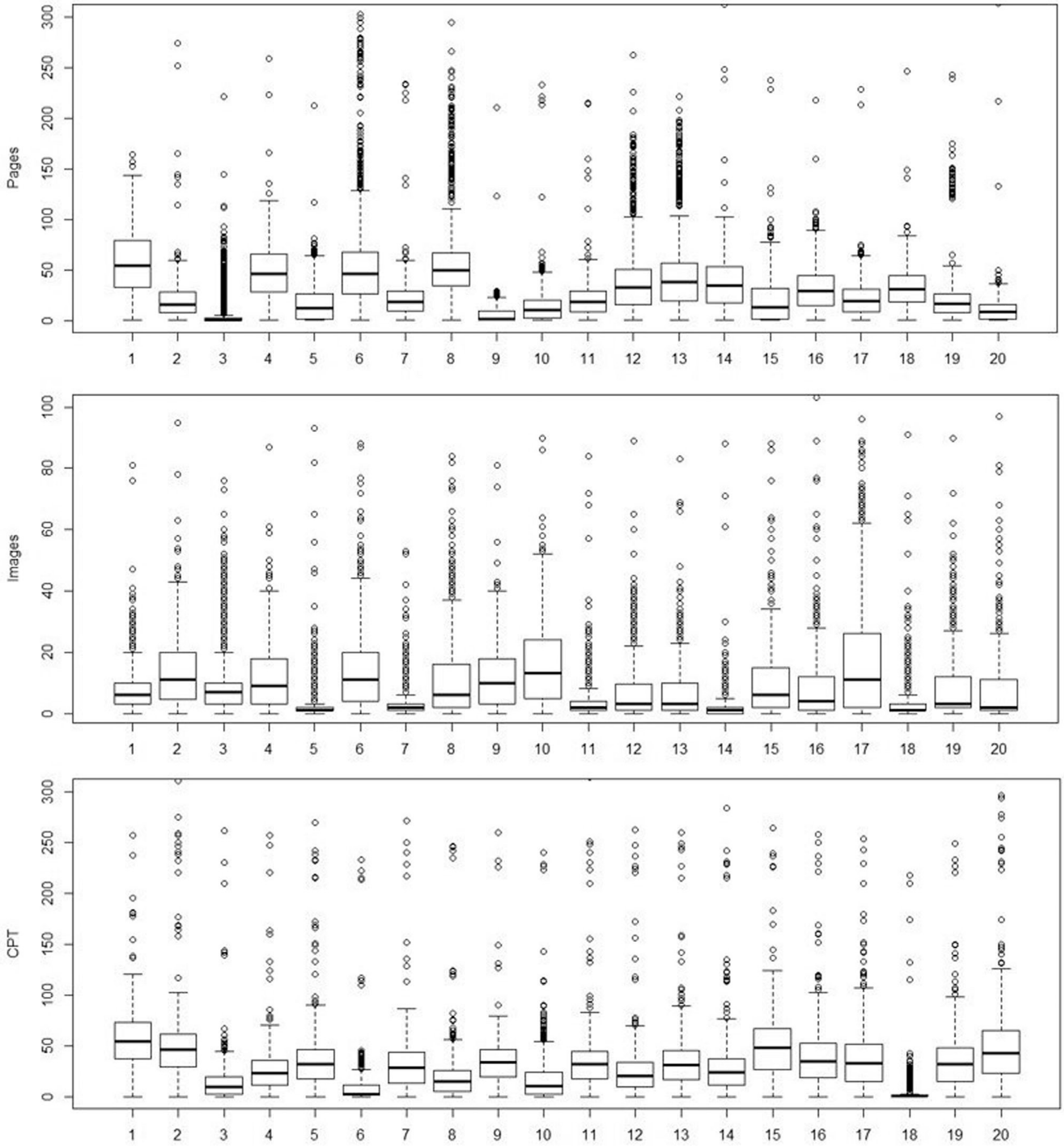


Fig. 4. Distributions of Pages, Images, and CPT.

Table 4
Results of the one-month diary study.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Tasks	244	142	336	118	243	169	214	271	162	235
Diff.	46	35	44	35	47	31	32	14	35	35
SR	52%	80%	59%	43%	53%	45%	38%	43%	57%	54%
Round	2.3	2.4	1.1	1.5	1.6	2.2	1.3	2.2	1.9	2
Time	9.4	10.5	5.1	5.8	9.2	8.8	5.1	10.5	6.2	6.1
	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20
Tasks	197	223	391	113	190	135	239	420	228	207
Diff.	37	22	42	12	25	29	22	55	34	29
SR	49%	23%	67%	42%	52%	76%	55%	45%	35%	76%
Round	2.2	2.4	1.9	2.2	2.2	1.9	2.2	2	1.3	2
Time	10.4	9.9	8.6	8.9	5.3	7.5	9.8	8.9	5.8	6.5

Table 5
Results of answering questions & Grades of attributes.

Attributes	Confident answers	Uncertain answers	Ignored	Grades
FCD	53	89	519	195
PE	551	43	67	1145
UAL	113	332	216	558
FAD	229	153	279	611
RE	438	176	47	1052
Coverage	561	45	55	1167
LAT	186	398	77	770
CPT	245	332	84	822
Tables	547	67	47	1161
UAT	63	198	400	324
Images	504	119	38	1127
Bib	295	253	113	843
Formulae	468	75	118	1011
Pages	412	175	74	999
AF	485	101	75	1071
FS	42	159	460	243
Authors	231	152	278	614

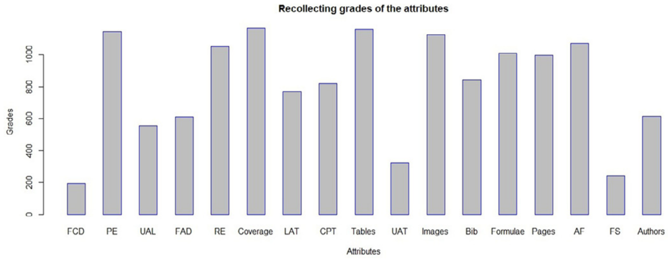


Fig. 5. Evaluating the participants' memory about the attributes.

are scored 0; questions with uncertain answers are scored 1. For multi-round re-finding tasks, if it succeeded, only the last round is counted; if it failed, only the first round is counted.

There are 661 difficult re-finding tasks were recorded by the system during the study. For each of the tasks, the participants were asked 17 questions listed in Tables 1 and 2. In total, 11,237 questions were answered during the experiment. 48.3% of the questions are given a confident answer, which can be used as good filters for the target; 25.5% of them are given an uncertain answer; and 26.2% of them are ignored, which implies the users cannot recall those attributes. Table 5 shows the results of answering questions and grades of the attributes. An attribute's grade is the sum of each grading about the attribute for all participants.

Fig. 5 illustrates a histogram of the attributes' memory recollecting grades based on the data of Table 5. It can be concluded that people usually have a good memory of "Printing experiences", "Coverage percentage", and whether a document contains tables, images. "Access frequency", "Pages", and "Formulae" can also be used as good re-finding attributes; however, some attributes like "File size" and "File creation date" are not inclined to be recollected confidently.

Since the system uses some attributes (such as "Bib", "Formulae", and "Tables") that are more popular in research papers, it is more useful for re-finding research papers than general PDF documents.

5. Discussion

People's reading behaviors and recall habits are always complicated and unpredictable. In order to develop a well-performing file re-finding system, we discuss some other fundamental issues and further possible improvement.

5.1. Privacy issues

The re-finding system will inevitably record essential document data and some of users' experiences, any these recorded data may involve one's privacies. Fortunately, the re-finding system mainly works for a user privately, and runs on the user's personal computer, where all the recorded data is stored personally. The startup of the system is password protected. To be safer, information about sensitive documents can be deleted manually.

5.2. Ask minimum questions

It is troublesome and impractical to ask the user all the questions during a re-finding process. A more appropriate way is to ask minimum and essential questions. A good question should be discriminative so that the search space can be reduced quickly, and be more likely recollected by the user. Further research is necessary to make a smart selection of the most appropriate questions for a particular search task. E.g., if the user never used a printer, the question about "Printing experience" should not be present in the question list; or the user has a poor memory about the "File sizes" of the documents, the question about "File size" should also be excluded. In addition, asking the participants many questions during the experiment might introduce some bias to the results. Because it imposes some burden to the participants when answering the questions, and disperses their energy of answering the questions that are really necessary. Besides analyzing the characteristics of a particular re-finding task, the system can learn the user's question answering habits and show only questions which are inclined to get confident answers.

5.3. Cooperating with conventional re-finding methods

Conventional re-finding methods like navigation and keywords retrieval can be combined with our method. Boardman and Sasse find people like to organize their files [32], the main purpose is for easy re-finding [33]. 97% of their files are placed in an intended folder, only 3% of their files are left in some default locations. The wizard can provide a list of frequently accessed folders, and let the user select a subset of them which he believes containing the target. If the user can provide some keywords (in a difficult re-finding context, these keywords are usually inaccurate to locate the target), the wizard can extend the keywords with their synonyms, then documents which do not contain the keywords and their synonyms can be excluded, diminishing the search space.

5.4. Order of the questions

The questions are roughly ordered with intrinsic attributes first, then user experience attributes. Since memories for different aspects of a document are not independent of one another, answering one question may lead to remembering other aspects. Further study should be conducted to examine the relationships of the attributes, and order the questions in a more elaborate way, improving efficiency and effectiveness of recollecting, meanwhile, mitigating users' cognitive burden.

5.5. What if a recollection is incorrect?

It is likely that the user has an incorrect memory about a particular attribute, this incorrect information may lead to a failed re-finding. Further work is necessary to find out which attributes are inclined to be recollected inaccurately, hence, excluding them from the question list. Pruning the question list also helps improve the efficiency of the wizard.

6. Conclusion

In this paper, we propose a new method to tackle difficult re-finding tasks, which actively collects a user's memory pieces about a document, then exploits the information to filter out the quested document. We argue that if the user remembers the existence of a document, he should remember some specialties of it that differentiate it from other documents, such as the keywords or folder of it. In a difficult re-finding scenario, the formulated *keywords* and *path* usually fail to locate the target. We introduce new attributes like "Coverage percentage", "Access frequency", "Number of images", and "Number of tables", which may serve as specialties of a document to locate it. To alleviate the user's cognitive burden during memory recollection, we employ a divide-and-conquer strategy with a Q&A wizard interface to split an intractable question into small steps. A PDF document re-finding system is developed to verify the feasibility of the method. A user study shows the PDF document re-finding system can attain 52% of difficult re-finding tasks when conventional methods fail. 90% of participants confirm the Q&A wizard interface can alleviate their cognitive burden during recollecting. Further extension of the system involves re-finding other file types (e.g., multimedia files), in which case some attributes should be substituted by other attributes like genre or duration etc.

References

[1] C. Jensen, H. Lonsdale, E. Wynn, J. Cao, M. Slater, T.G. Dietterich, The life and times of files and information: a study of desktop provenance, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2010, pp. 767–776.

[2] J.S. Sachs, Recopition memory for syntactic and semantic aspects of connected discourse, Percept. Psychophys. 2 (9) (1967) 437, doi:10.3758/BF03208784.

[3] E. Tulving, Precis of elements of episodic memory, Behav. Brain Sci. 7 (02) (1984) 223–238.

[4] F.I. Craik, R.S. Lockhart, Levels of processing: a framework for memory research, J. Verbal Learn. Verbal Behav. 11 (6) (1972) 671–684.

[5] D.L. Schacter, C.R. Savage, N.M. Alpert, S.L. Rauch, M.S. Albert, The role of hippocampus and frontal cortex in age-related memory changes: a pet study., Neuroreport 7 (1996) 1165–1169.

[6] R.C. Atkinson, R.M. Shiffrin, Human memory: a proposed system and its control processes, Psychol. Learn. Motivation 2 (1968) 89–195.

[7] L.L. Jacoby, J.P. Toth, A.P. Yonelinas, Separating conscious and unconscious influences of memory: measuring recollection., J.Exp. Psychol. 122 (2) (1993) 139.

[8] E. Tulving, Z. Pearlstone, Availability versus accessibility of information in memory for words, J. Verbal Learn. Verbal Behav. 5 (4) (1966) 381–391.

[9] E. Tulving, D.M. Thomson, Encoding specificity and retrieval processes in episodic memory., Psychol. Rev. 80 (5) (1973) 352.

[10] W. Jones, Personal information management, Annu. Rev. Inf. Sci. Technol. 41 (1) (2007) 453–504.

[11] B. McKenzie, A. Cockburn, An empirical analysis of web page revisitation, in: System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on, IEEE, 2001, pp. 9–pp.

[12] J. Teevan, E. Adar, R. Jones, M.A. Potts, Information re-retrieval: repeat queries in yahoo's logs, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2007, pp. 151–158.

[13] D. Elsweiler, M. Harvey, M. Hacker, Understanding re-finding behavior in naturalistic email interaction logs, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2011, pp. 35–44.

[14] M.W. Lansdale, The psychology of personal information management, Appl. Ergon. 19 (1) (1988) 55–66.

[15] D.K. Barreau, Context as a factor in personal information management systems, J. Am. Soc. Inf. Sci. 46 (5) (1995) 327–339.

[16] O. Bergman, R. Beyth-Marom, R. Nachmias, N. Gradovitch, S. Whittaker, Improved search engines and navigation preference in personal information management, ACM Trans. Inf. Syst. (TOIS) 26 (4) (2008) 20.

[17] S.K. Tyler, J. Teevan, Large scale query log analysis of re-finding, in: Proceedings of the Third ACM International Conference on Web Search and Data Mining, ACM, 2010, pp. 191–200.

[18] J. Teevan, C. Alvarado, M.S. Ackerman, D.R. Karger, The perfect search engine is not enough: a study of orienteering behavior in directed search, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2004, pp. 415–422.

[19] A. Azagury, M.E. Factor, Y.S. Maarek, B. Mandler, A novel navigation paradigm for xml repositories, J. Am. Soc. Inf. Sci. Technol. 53 (6) (2002) 515–525.

[20] B. Wei, J. Liu, Q. Zheng, W. Zhang, C. Wang, B. Wu, Df-miner: domain-specific facet mining by leveraging the hyperlink structure of wikipedia, Knowl.Based Syst. 77 (2015) 80–91, doi:10.1016/j.knosys.2015.01.001.

[21] M.G. Armentano, D. Godoy, M. Campo, A. Amandi, Nlp-based faceted search: experience in the development of a science and technology search engine, Expert Syst. Appl. 41 (6) (2014) 2886–2896, doi:10.1016/j.eswa.2013.10.023.

[22] T. Deng, L. Zhao, H. Wang, Q. Liu, L. Feng, Refinder: a context-based information refinding system, Knowl. Data Eng. IEEE Trans. 25 (9) (2013) 2119–2132.

[23] P. Qvarfordt, S. Tretter, G. Golovchinsky, T. Dunnigan, Searchpanel: framing complex search needs, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2014, pp. 495–504.

[24] D.H. Chau, B. Myers, A. Faulring, What to do when search fails: finding information by association, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2008, pp. 999–1008.

[25] S. Dumais, E. Cutrell, J.J. Cadiz, G. Jancke, R. Sarin, D.C. Robbins, Stuff i've seen: a system for personal information retrieval and re-use, in: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2003, pp. 72–79.

[26] M. Ringel, E. Cutrell, S. Dumais, E. Horvitz, Milestones in time: the value of landmarks in retrieving information from personal stores, in: Proceedings of Interact, 2003, 2003, pp. 184–191.

[27] S. Trullemans, A. Sanctorum, B. Signer, Pimvis: Exploring and re-finding documents in cross-media information spaces, in: P. Buono, R. Lanzilotti, M. Matera, M.F. Costabile (Eds.), Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI 2016, Bari, Italy, June 7–10, 2016, ACM, 2016, pp. 176–183, doi:10.1145/2909132.2909261.

[28] S. Sadeghi, R. Blanco, P. Mika, M. Sanderson, F. Scholer, D. Vallet, Predicting re-finding activity and difficulty, in: A. Hanbury, G. Kazai, A. Rauber, N. Fuhr (Eds.), Advances in Information Retrieval - 37th European Conference on IR Research, ECIR 2015, Vienna, Austria, March 29, - April 2, 2015. Proceedings, Lecture Notes in Computer Science, 9022, 2015, pp. 715–727, doi:10.1007/978-3-319-16354-3_78.

[29] F. Meier, D. Elsweiler, Going back in time: An investigation of social media re-finding, in: R. Perego, F. Sebastiani, J.A. Aslam, I. Ruthven, J. Zobel (Eds.), Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17–21, 2016, ACM, 2016, pp. 355–364, doi:10.1145/2911451.2911524.

[30] M. Franceschet, Pagerank: standing on the shoulders of giants, Commun. ACM 54 (6) (2011) 92–101, doi:10.1145/1953122.1953146.

[31] S.E. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Foundations Trends Info. Retrieval 3 (4) (2009) 333–389, doi:10.1561/1500000019.

[32] R. Boardman, M.A. Sasse, Stuff goes into the computer and doesn't come out: a cross-tool study of personal information management, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2004, pp. 583–590.

[33] D. Barreau, B.A. Nardi, Finding and reminding: file organization from the desktop, ACM SIGCHI Bull. 27 (3) (1995) 39–43.