The Forgotten Needle in My Collections: Task-Aware Ranking of Documents in Semantic Information Space

Tuan Tran[1], Sven Schwarz[2], Claudia Niederée[1], Heiko Maus[2], Nattiya Kanhabua[1,3]

[1]L3S Research Center / Leibniz Universität Hannover, Germany
[2]German Research Center for Artificial Intelligence (DFKI) / Kaiserslautern, Germany
[3]Department of Computer Science, Aalborg University, Denmark
[1]{ttran, niederee}@l3s.de, [2]{heiko.maus, sven.schwarz}@dfki.de, [3]nattiya@cs.aau.dk

ABSTRACT

With the growing amount of content stored in personal and organizational information spaces, finding and re-finding documents becomes both more crucial and challenging. In this work, we propose an approach to reduce information overload in navigation by automatically focusing on important documents, adaptively to the tasks at hand. Based on the idea of managed forgetting, we present a ranking method, which unifies activity logs and semantic information about documents into a common framework to identify important documents to the user's current tasks. Our experiments on two real-world datasets, both collected from knowledge work activities in professional scenarios, show that our ranking approach outperforms the baseline methods for both subsequent access prediction and the effectiveness in ranking important documents. Furthermore, we implemented and demonstrated a system for decluttering information spaces as a proof of concept of our managed forgetting approach.

1. INTRODUCTION

The advance of technologies has supported the idea of massive digital content creation and sharing, as well as the reluctance of content removal. This leads to a growing number of data, making finding and re-finding a particular resource increasingly difficult. In non-public sources such as the personal and organizational information spaces, navigation is a preferred mechanism to find a document for a task at hand, due to its lower cognitive load, its consistency and the strength of the location metaphor [6]. However, with the huge amount of digital content generated every day, access via navigation becomes challenging. The mission is more frustrating in large heterogenous information ecosystems such as in different devices, or in systems that get cluttered after long-time working on different tasks.

For decluttering such information spaces and supporting the finding or re-finding of resources according to the user's short-term interest, we propose a *ranking* approach to ease the navigation, based on the idea of *managed forgetting*. The idea is inspired by science of forgetting and remembering [22, 24]: The human brain is very effective in focusing on important things, while forgetting irrelevant details. This trait is reflected in human practices of organizing their collections, i.e., they often create shortcuts to easily navigate to relevant resources in the desktop environment or mobile home screens, or bookmark important Web pages. Managed forgetting aims to relieve such manual efforts by automatically computing the *short-term* importance of a document with respect to the user attention. It replaces the binary decision on importance by a gradually changing value: Information sinks away from the user with a decreasing value, which we call *memory buoyancy*. This value can be used for ranking important resources for a task at hand, thereby decluttering the information spaces adaptively.

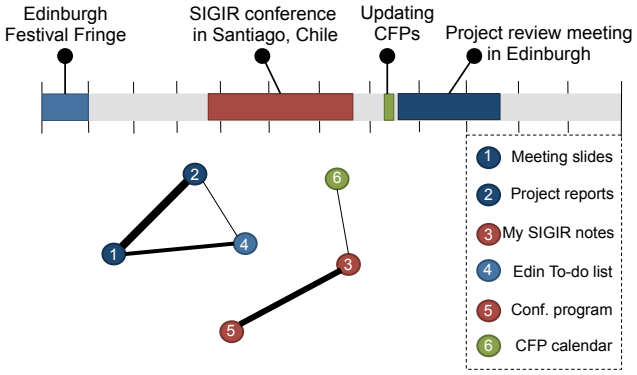


Figure 1: Recommended documents in a personal information management system: The upper bar corresponds to the timeline of tasks. Bottom right is the recommended documents. User accessed the documents when conducting the tasks (displayed with the same color). Graph on bottom left represents the document relations. Width of lines indicates the strength of the connections. The forgotten document 4 resurges due to its relevance to the current task.

As in human forgetting, memory buoyancy is driven by resource usage, importance decay, and semantic associations [2]. In Information Retrieval, different algorithms and systems have been proposed to identify important documents in an information space [12, 25]. Much of existing approaches rely on activity logs and assumes that recently accessed documents are also more likely to be accessed in the future [34]. As a result, many proposed algorithms assess documents based on recency and frequency access evidences. They ig-

CHIIR '16, March 13-17, 2016, Carrboro, NC, USA

DOI: http://dx.doi.org/10.1145/2854946.2854971

nore a wide variety of other factors, which equally influence the importance of a document for a task at hand. For instance, according to the associative character of the brain, a document might be important because of its relatedness to the current task, even if the document has not been touched for a long time [2, 28]. In the example shown in Figure 1, while preparing for a business trip to Edinburgh, the user might recall or might have forgotten useful notes from her private holiday in Edinburgh some years ago. Ideally, a system should bring this information up again, but it is infeasible when only relying on the activity history alone. This example demonstrates the need for a more comprehensive ranking method, taking into account the intrinsic relatedness of documents to current tasks.

Our method combines evidences from activity logs with semantic associations between documents to devise a unified document ranking framework. The idea is that a document is important to the user's current task, if it either has been frequently accessed by the user, or is highly related to other important documents. This is illustrated in Figure 1 for six documents. The user's intensive accesses to "Meeting slides" and "Project reports" during the preparation for the project meeting in Edinburgh (dark-blue part of the upper horizontal bar) give the documents higher ranks. Meanwhile, the connections between these documents and the "To-do list" from a past trip to Edinburgh endorses this list to the current task, bringing it back to the user's attention.

The importance of considering document relationships to rank documents, for example, in personal information management (PIM) systems, has been shown in [31, 8]. The common idea is to propagate the "importance score" of a document to other related documents in a graph of different document relationship. However, most of existing work studies the document relations in isolation, assumes they are equally important, or puts arbitrary weights to the relations in an ad-hoc manner [31]. In contrast, we propose a unified framework based on machine-learning methods. We conduct studies in different settings, and systematically evaluate the effectiveness of our framework from the quantitative to qualitative perspectives. In summary, our contributions are:

1. A novel framework for realizing managed forgetting based on a machine-learning method. It unifies activity log information and semantic relationships to rank documents based on user's current tasks.

2. Experiments on two real-world data sets acquired from daily usages in professional scenarios, which show the effectiveness of our proposed method.

3. Implementation of the managed forgetting into a real-world Semantic Desktop system for decluttering professional information spaces.

2. RELATED WORK

In the area of personal information management (PIM) systems, there has been a rich body of work studying how to organize and find information effectively [18, 12, 16, 7]. Based on findings about user preferences in information navigation [18, 7, 32], different methods have been proposed to improve the finding and re-finding process [12, 16]. Existing work relies on structural signals of documents, which do not reflect the user preferences in finding, but rather in organizing the file systems [18]. Other work, such as studies by Teevan et al. [32] and by Bergman et al. [5], address the retrieval needs (findding and re-finding of documents); but they lack the deep understanding of user intents when interacting with documents of different semantics. The limitation comes from the fact that such semantics of document attributes and associations are not easily observed in traditional PIM systems. The advances of annotation and information extraction technologies have enabled this with the emerging of Semantic Desktop systems [29]. The NEPOMUK Semantic Desktop [1] has built an infrastructure providing a semantic layer over conventional documents (e.g., emails, Web pages, documents, pictures, etc.) with a personalized vocabulary resembling the mental model of the user [30]. A similar approach was also employed in other systems, including Haystack [27], MyLifeBits [17], etc. This trend opens a new line of research in document finding and re-finding, exploiting the semantic information. To the best of our knowledge, our study is the first in this direction.

In the context of document ranking, many solutions have been proposed for PIM systems, which can be classified in two categories: activity logs-based and relation-based. The former involves analysing user behaviour from the past activities, ranking documents based on their recency and frequency testimony. This can be estimated by different decay functions, each has a different cognitive plausibility and encodings [26, 15, 10, 2]. It can also be modeled via Markovian processes [11], or balanced with short- and long-term aspects [34]. The relation-based method, on the other hand, exploits the relationships between documents to propagate their importance scores along the relation graph. Such "propagation" technique has been proven to be useful in a wide variety of domains and retrieval scenarios, from personal file search [31, 33] to collaboration spaces [14]. The document relations can come from different evidences. Most established methods attempt to identify related documents by their correlation in access behaviour (e.g., Web pages that are often visited together, files that are opened in the same session, etc.). These methods then employ techniques such as approximating sessions and file operation timelines [33], or use session signals in some applications such as web browser [25]. Recently, there are attempts to combine these two categories into one framework, using one-step propagation to actitivy-based ranking in a layered approach, or linearly combine in a hybrid model [3]. In contrast to existing work, we propose a new unified framework that leverages both activity logs and document relationships in principled manner, based on machine-learning models with no manual-defined trainind data. Our model also works for any heterogenous relation networks, and to that extent, generalizes existing work, which focuses on individual relations [25].

There has been extensive work in the area of *desktop search*. The common approach is to combine content-based techniques from traditional ad-hoc text retrieval with contextual information learnt from user's activities. For instance, the system can complement search with document metadata such as attributes and types [9, 7, 13], or exploit additional information from environments (related tasks, structures of directories, or relationships with other documents) to adjust the document relevance. Typical ranking approach includes graph-based propagation methods such as random walks [8, 31, 33]. From the perspective of search interface, Bergman et al. [6] suggested that navigation is a better way

to interactively explore the search results due to its cognitive advantages. The idea is also realized in systems such as Stuff I've Seen [12], Search Directed Navigation [16]. While these systems study conventional types of desktop, we provide the first implementation in a Semantic Desktop setting.

3. APPROACH

In this section, we describe our approach to managed forgetting, which exploits document's access information, and subsequently applies different forgetting (or decay) functions as well as propagates the importance of related documents via semantic relationships.

3.1 Preliminaries

A semantic information space in a PIM system is a collection of documents or resources, which is denoted as D. A document or a resource d can be of different types (e.g., photo, office document, folder, web page, etc.) and has different attributes (e.g., title, authors, and creation time). Between any two documents d_1 and d_2 can exist multiple relations with different semantics. For instance, d_1 and d_2 are both created by the same author, or d_1 is the containing folder of the file d_2. Relations can be associated with some scores indicating the strength of their relation, for instance, the cosine score for content similarities. Let R denote a set of all semantic relations. For each pair (d_i, d_j), we have an $|R|$-dimensional vector $X_{ij} = (x_{ij1}, x_{ij2}, \ldots, x_{ij|R|})^T$, where $x_{ijk} \geq 0$ represents the score of the k-th relation between d_i and d_j, $x_{ijk} = 0$ if the d_i and d_j are not connected by the relation. Usually, the number of relations is small compared to the number of all documents in the information space. The collection of relation scores $X = \{X_{ij}\}$ forms the weights of edges in a multigraph, where nodes are all documents in D, and each edge corresponds to a semantic relation.

In our work, we model time as the sequence of equal time intervals and denote by $T = (t_1, t_2, ..., t_m)$, where the time point t_i is the index of i-th interval from the beginning of the PIM system. In one interval, a document can be accessed and used by one or multiple users. Each access is represented by a triple $a = (d, u, t)$, indicating that the user u performed an action on the document d at time t. Given a user u (or a group of users $U = \{u_i\}$), a document d and a time point of interest t, the sequence of actions on all documents of D, performed by u (or in the case of U, by at least one user u_i), happened before t and in chronological order, forms an activity history of u (or U) in the information space, and is denoted by $L_t = (a_1, a_2, ..., a_n)$. Given a document d and time t, we refer to as a document access history, denoted by $L_{d,t}$, as those actions performed on d. A sequence of time points t_i for actions in $L_{d,t}$ (can be repeated because of multiple accesses to d within one interval) constitutes the access times of d, denoted by $T_{d,t}$. The most recent access time to d before t (last time point in $T_{d,t}$) is denoted by t_d.

Problem. Given a collection of documents D, a set of relation scores X, time of interest t, and an activity history L_t corresponding to a user u, or to a group of users U, identify documents with highest importance with respect to u's or U's tasks at time t, as inferred from L_t.

We tackle the aforementioned problem in two steps. In the first step, we mine the activity history and devise a memory buoyancy scoring function based on the recency and frequency (see Section 3.2), so that more recently and frequently accessed documents get higher memory buoyancy scores. In the second step, we employ a *propagation* method that identifies highly connected documents, to transfer the activity-based scores of documents along the connection. This is similar to the layered approach by Kawase et al. [19]. However, while the authors merely identify connections from sessions of the activity history, we devise a generalized framework that works with different heterogenous relations.

Method name	Function	Parameters
Most Recently Used	MRU(d,t)=$\frac{1}{t-t_d+1}$	None
Polynomial Decay	PD(d,t)=$\frac{1}{(t-t_d)^{\alpha}+1}$	α: Decay rate
Ebbinghaus Curve	Ebb(d,t)=$e^{\frac{(t_d-t)}{S}}$	S: Relative memory strength
Weibull Distribution	Wei(d,t)=$e^{-\frac{\alpha(t-t_d)^s}{s}}$	s: Forgetting steepness, α: Volume of what can be remembered

Table 1: List of Activity-based ranking functions

3.2 Activity-based Memory Buoyancy

In order to compute the memory buoyancy scors, we use the access times of the document from the access history. We estimate the score of a document through the distances of previous access time points and the time of interest. The scoring function is formally defined as follows:

Definition 1. *An activity-based memory buoyancy scoring function is a function that takes as input the time t and document d, and outputs a value $v(d,t) \in [0,1]$ (memory buoyancy score) such that:*

1. $v(d,t) = 0$ *if* $T_{d,t} = \emptyset$
2. $v(d,t_{i+1}) < v(d,t_i)$ $\forall t_i, t_{i+1} \in T_{d,t}$
3. $v(d_1,t) < v(d_2,t)$ *if* $|T_{d_1,t}| < |T_{d_2,t}|$ *or* $t_{d_1} < t_{d_2}$

The above conditions ensure that the memory buoyancy scores, if no other evidences present, is driven by the decay effect. In Table 1, we present different activity-based scoring functions studied in this work, each corrsponds to one decay function. Each of these functions only considers the most recent time t_d, and can be considered as a basic **recency-based** method.

Frequency. In [2], Anderson et al. suggest that the frequency of interactions also play an important role in the human's recalling process of a resource, as by the re-learning effect. Hence, for each of the functions in Table 1, we introduce a "frequency"-based variant, which aggregates the effect of decays in different time points:

$$v_f(d,t) = \sum_{t_i \in W} v_r(d,t_i) \quad (1)$$

where $v_r(d,t_i)$ can be any of recency-based functions in Table 1. The sequence $W \subseteq T_{d,t}$ represents the time window in which all time points are taken into consideration for the ranking. For instance, if $W = T_{d,t}$ and $v_r = MRU$, we have the well-established *most frequently used* method in cache replacement policies. If $W = T_{d,t}$ and $v_r = PD$, we have the decay ranking model in [25]. The *Frequency* algorithm used in Mozilla Firefox [10], on the other hand, constructs W from only the last ten items of $T_{d,t}$, in order to avoid the convolution of too old accesses into the current rank. In this work, we follow this idea, and only aggregate from the last ten time points of accesses for each document.

3.3 Propagation Methods

The drawback of recency-based and frequency-based scoring functions is that they consider each document in isolation. In practice, however, humans tend to recall and find documents together within some contexts, e.g., they can follow some cues and associate a document with other related documents which are easier to recall and navigate. This exploitation of document relations is inspired by the cognitive science of associative memory [8], and is studied in a rich body of work [31, 8, 33, 19]. Most of the related work employs a propagation method in the document relations graph, which "transfer" the ranking score of each individual document to other related documents along the edges of the graph. However, these methods are non-learning and largely based on heuristics to combine relations weight, which requires extensive tuning, as mentioned in [31].

In our work, we develop a propagation method that combines different relations into a unified framework, and learns the weighting for the combination automatically. We model the process that the user finds an important document as a Markov process, where she recalls and searches for important documents via the related resources. For each pair of connected documents (d_i, d_j), we define the transition probability from document d_i to document d_j as:

$$p_{ij}(w) = \begin{cases} \frac{\sum_{k=1}^{|R|} w_k x_{ijk}}{\sum_{l=1}^{|D|} \sum_{k=1}^{|R|} w_k x_{ilk}} & \text{if } X_{ij} \neq \emptyset \text{ and } L_{d_j,t} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where w is the weighting vector for the semantic relations in R. The condition $L_{d_j,t} \neq \emptyset$ ensures that the propagation has no effect on the documents that have not been created before the time t, i.e., no propagation to the future. Similarly, the indices l's run only over the documents d_l with $L_{d_l,t} \neq \emptyset$. Consequently, we have $\sum_j p_{ij} = 1$ for all documents d_i. In practice, to avoid rank sink when performing the propagation process, if a document has no relations at all, we assume a dummy edge from it to all other documents with zero probability.

Next, we describe our propagation framework. Let P denote the transition matrix of documents in D, we follow PageRank model and define the propagation as the iterative process, where in each iteration, the memory buoyancy values of documents are updated by the following equation,

$$\mathbf{s}^{(n+1)} = \lambda P^T \mathbf{s}^{(n)} + (1-\lambda)\mathbf{v} \tag{3}$$

where $\mathbf{s}^{(n)} = (s(d_1,t), s(d_2,t), ..., s(d_m,t))$ is the vector of documents' memory buoyancy values at iteration n, (m is the number of documents appearing in L_t), $\mathbf{v}$ is the vector of values obtained by an activity-based scoring method, and λ is the damping factor. In Equation 3, we need to learn the model of weighting parameters w in order to complete the transition matrix P, described in the following.

3.4 Learning to Propagate

The aim of the learning is to identify the weights $w_1, .., w_{|R|}$ of the semantic relations with which we obtain the best prediction of document rankings. In this work, we propose to exploit the activity history to learn the optimal w. In particular, we simulate the navigation of the user at each time points t' in the past, and compare the computed ranks of the documents with the ranks based on the frequency of access in the time point $t'+1$. The idea is to learn w so as to minimizes the number of mis-ranked pairs, i.e., pairs (d_1, d_2) with $s(d_1,t') > s(d_2,t')$ but d_1 has been accessed less frequently than d_2 until $t'+1$.

Formally, we define the label $y_{ij} = s(d_i,t') - s(d_j,t')$ and the groundtruth $\hat{y}$, $\hat{y}_{ij} = -1$ if d_i has less access than d_j at $t'+1$ and $\hat{y}_{ij} = 1$ otherwise. We learn w by the following optimization problem:

$$\min_w F(w) = \|w\|^2 + \theta \sum_{(d_i,d_j)\in A} h(y_{ij}) \tag{4}$$

where A is the training data, θ is the regularization parameter that controls the complexity of the model (i.e., $\|w\|^2$) while minimizes the mis-ranked pairs in A via the loss function h. In this work, we apply the simple hinge loss function: $h(y) = \max(0, 1-\hat{y}.y)$. Next, we detail how to solve Equation 4 and to how we collected the training data A.

Optimization Solution. Following [4], we solve the Equation 4 by a gradient descent method. The partial derivative of $F(w)$ with respect to each relation weight w_k is:

$$\frac{\partial F}{\partial w_k} = 2w_k + \theta \sum_{(d_i,d_j)\in A} \frac{\partial h(y_{ij})}{\partial y_{ij}} \left(\frac{\partial s(d_i,t')}{\partial w_k} - \frac{\partial s(d_j,t')}{\partial w_k}\right) \tag{5}$$

The derivative of the hinge function is trivial. For $s(d,t)$, we have from Equation 3:

$$\frac{\partial s(d_i,t')}{\partial w_k} = \lambda \sum_j \left(p_{ji} \frac{\partial s(d_j,t')}{\partial w_k} + s(d_j,t') \frac{\partial p_{ji}}{\partial w_k}\right) \tag{6}$$

and:

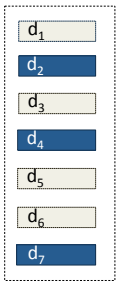
$$\frac{\partial p_{ji}}{\partial w_k} = \frac{x_{jik} \sum_i \sum_k w_k x_{jik} - (\sum_k w_k x_{jik})(\sum_i x_{jik})}{(\sum_i \sum_k w_k x_{jik})^2} \tag{7}$$

From Equations 6 and 7, we can easily calculate the gradients $\frac{\partial s}{w_k}$ by a power-iteration algorithm such as in [4], and then apply a gradient descent-based learning method, for instance L-BFGS method [21], into Equation 5 to learn w.

Soft Labeling. We start with identifying the training time points t' to observe the subsequent accesses. In principle, any time points before the time of interest t can be chosen. In practice, however, we observe that time points during the burst periods, i.e., the period where there is a significantly higher number of access than usual, are more "interesting" to observe both correctly and falsely ranked pairs. To this extent, we apply the Kleinberg algorithm [20][1] to identify the burst periods from the times series of the document accesses. Then, for each period, we pick up the time point with highest number of access for the training.

Next, we build a balanced set of positive and negative pairs for the training data as follows (Figure 2). For each training time point t', we extract the set of all documents accessed by the user at $t'+1$, denoted by $D_{t'+1}$. Then, we apply a baseline activity-based scoring method with respect to t for documents in D, sort them in descending order of the memory buoyancy values. From this sorted list, we get the top-scored documents until all documents in $D_{t'+1}$ are included. We call this set S. The set $E = S \setminus D_{t'+1}$ consists of documents with high memory buoyancy scores but not

[1]We use implementation in http://wiki.cns.iu.edu/display/CISHELL/Burst+Detection with default parameters



(d_1, d_2)	-1	(d_3, d_4)	-1
(d_1, d_4)	-1	(d_3, d_7)	-1
(d_1, d_7)	-1	(d_4, d_5)	1
(d_2, d_3)	1	(d_4, d_6)	1
(d_2, d_5)	1	(d_5, d_7)	-1
(d_2, d_6)	1	(d_6, d_7)	-1

Figure 2: Example training data. Ranks of documents (left-hand side) are produced by a baseline scoring method. Documents in dark blue are accessed in the next time point. All documents from the first rank to the lowest rank of the accessed documents (d_7) are used for the training. Table in the right-hand side consists of training pairs, together with the labels.

accessed in the next time points, i.e., the false positives. Finally, we construct the training pairs (d_i, d_j) by picking d_i from the sets $D_{t'+1}$ or E, such that d_i, d_j are not in the same set, and that the estimated memory buoyancy score of d_i is higher than of d_j. As an example from Figure 2, we have (d_1, d_2) is a training pair, but (d_2, d_1) is not, because the estimated score of the document d_2 is smaller than d_1's. Similarly, $(d_1, d_3) \notin A$ because $d_1, d_3 \in E$.

To assign the training labels, if $d_i \in D_{t'+1}$ and $d_j \in E$, then we assign $\hat{y}_{ij} = 1$. Otherwise, if $d_i \in E$ and $d_j \in D_{t'+1}$, we set $\hat{y}_{ij} = -1$ (Figure 2).

3.5 Semantic Graphs

The semantic relations R play an important role in the propagation method. Depending on specific domains of applications and scenarios, we can have different types of semantic relations, which can fall in two categories:

Explicit Relations: This category consists of relations that are observable from the structures of the documents and the information space, e.g., references or hyperlinks from one document to others, the containment relations between folders and files, etc. The relations can be specified by users with the help of some software components and interfaces, such as aforementioned NEPOMUK Semantic Desktop. Recently, with the proliferation of Semantic Web and RDF technologies, some semantic relations are standardized as predicates between documents, such as hasPartOf, hasAttachment, etc.

Implicit Relations: This category includes relations that are inferred from the contents of documents, or from user activity patterns; for instance, content similarities, the correlation of two documents being accessed frequently in the same or close sessions or time. The advantage of this type of relations as compared to the explicit ones is that it can be constructed automatically without much human effort. In the following, we will focus on this type of relations. Note that while many explicit relations are asymmetric, the implicit relations discussed below are symmetric. To unify them in the same relation space R, for each implicit relation between documents d_i and d_j, we fill the score into the corresponding dimension in both X_{ij} and X_{ji}. In this work, we consider the following types of implicit relations:

Text-based Relation: This type of relation relies on the similarity of document contents. More specifically, for each document, we build the bag of words from its main content body, and calculate the Cosine similarity of the two documents to measure the strength of their relation.

Atribute-based Relation: The text-based relation is only applicable for rich-text types such as e-mails, web pages, office documents, etc. For other types of documents, we assume the Cosine similarity is zero, as no text can be extracted. For these documents, we rely on metadata specified by the users or software components. These attributes are often represented in form of <attribute, set of values>. For instance, tags of a photo, list recipients of an email, etc. We define one relation for each specific attribute, and measure the relation strength by calculating the Jaccard similarity over the two corresponding sets of values.

Time-based Relation: This relation is derived from the activity history. It is based on the assumption that two documents are highly related with respect some latent tasks, if the user accessed to both of them in many sessions. To identify the "good" sessions for the two documents d_1, d_2, we apply the same heuristic as for building the training data: We extract all time points from the burst time periods of $T_{d_1,t}$ and $T_{d_2,t}$ to create the two sub-sequences for d_1, d_2 respectively. We then calculate Jaccard similarity between these two sequence and use as the time-based relation strength.

4. EXPERIMENTS

4.1 Datasets

In this work, we conduct experiments on two real-world datasets with different characteristics. Table 2 summarizes the statistics of these datasets. In the following, we give more detailed information for each dataset.

4.1.1 Semantic Desktop

The first dataset (named Person) consists of personal collections obtained via a Semantic Desktop infrastructure described in [23] and deployed at the DFKI. The resulting knowledge base consists of resources, their semantic representations and relations with concepts spanning a semantic graph based on the PIMO (Personal Information Model), a state-of-the-art ontology for PIM [30]. At the time of evaluation, the Semantic Desktop infrastructure has been used for over 3 years in the Knowledge Management team at DFKI on daily basis by 17 users, who are employees and students in DFKI. Among these users, 7 are active with usage of 4 to 8 hours per day, others are occasional users such as interns or assistant students. The PIMO data is stored in a knowledge base on a central server and is related to professional or research activities, e.g., business meetings, project proposals, tasks, and notes, etc. The knowledge base is accessed via the Semantic Desktop infrastructure consisting of components such as a plug-in embedded into the Windows File Explorer, a Firefox add-on, a plug-in to an Email client, and a web-based stand-alone application.[2] These are installed on each individual's computer at work, and are used on a daily basis. It enables the user to easily annotate documents when they conduct their regular tasks: Browsing the Web, reading emails, managing files on hard disks or creating calendar events. The user is also enable to create and use semantic concepts, such as topics, locations, persons, tasks, events, or documents. To this extent, a semantic layer is built over

[2]For a detailed explanation and videos see our ForgetIT Pilot documentation at https://pimo.opendfki.de/wp9-pilot/

Figures	Person	Collaboration
No. of documents	20363	1437
Time span	Sep 2011 - Sep 2014	Oct 2008 - Sep 2014
Users	17	268
Relations	155539	126326
Activity log entries	337528	217588

Table 2: Statistics of the Datasets

the "physical" information objects, e.g., in the file system, mapping each document or concept to a *resource*. In our experiments, we apply our methods to provide ranks for these semantic PIMO concepts instead of the actual documents.

To capture the activity history, a monitoring tool[3] was used capturing each event on the user's computer in a centralized database (only the owner can explore the log in raw format). To guarantee the privacy of personal data, real data resides in the knowledge base, only encoded information of document metadata and action logs (no content and physical files) are sent to our system for the experiments.

Relationship. All explicit semantic relations are represented in the form of RDF predicates (Table 3). Some explicit relations are symmetric, while others have inverse relations (displayed in Table 3 in parentheses). Concerning the implicit relations, as we could not obtain the contents due to privacy, we only construct a number of attribute-based and time-based relations (Table 3).

4.1.2 Collaborative Wiki

The second dataset (named Collaboration) is a intranet portal used within L3S Research Center as part for communicating daily research activities. The portal has been continuously used in the course of 6 years and includes research information such as collaboration projects with external partners, internal research activities, as well as administration information of the lab. Documents are mostly in hypertext format, but also include digital files uploaded to the portal. In contrast to the Person dataset, the Collaboration dataset has no full-fledged ontologies, and the documents are not associated with abstract concepts. Nevertheless, the portal runs on top of the DokuWiki platform[4], and support a few annotation via different plugin: Tagging with words, showing document author and contributors, etc. For the activity history, it uses Squid cache[5] to log HTTP access requests to the portal resources. In addition, an archiving tool is developed to log all revisions of the portal documents, together with their edit activities. The dataset are obtained in the form of an archive with all raw data content.

Relationship. Compared with the dataset Person, the documents in this dataset have much less explicit semantic relations, and all come from the structure of the portal (Table 3). However, as we have access to contents of the documents, we can build more implicit relations, listed in Table 3. The tagged-token-based (TTB) relations are constructed as follows. First, we extracted the content of all documents from the portal using a developed Dokuwiki parser (As for the MIME-typed documents, we use the software toolkit Tika[6] to parse the main content bodies). Next, we sample the documents related to 4 different collaborative research projects, tokenize the corresponding texts, and remove over-popular words and stop words. An experienced colleague working in numerous projects is asked to annotate the tokens with respect to 6 different classes: 1) Person or Person role; 2) Location; 3) Organisation; 4) Technical word (e.g., middleware); 5) Professional domain (e.g., meeting); 6) Project-specific terms. Each class is then treated as one attribute of the documents, with their tagged tokens treated as values, for calculating the corresponding attribute-based relation.

[3]http://usercontext.opendfki.de/
[4]https://www.dokuwiki.org/dokuwiki
[5]http://www.squid-cache.org
[6]https://tika.apache.org

4.2 Experiment Setup

4.2.1 Baselines

We evaluate our system against the following baselines:

Recency-Frequency: This set of baselines use values of the activity-based scoring functions to provide the final ranking, without using propagation. This includes the two recency-based methods MRU and Ebb, and their frequency-based variants, denoted by FMRU and FEbb. For polynomial and Weibull functions, we evaluate only the frequency-based methods, denoted by FPD and FWei, as they are shown to outperform the recency versions [26].

PageRank: This baseline ranks the documents by their authority scores, estimated in a graph of documents relations. The scores of documents are initialized equally. It can be thought of as the propagation method without the activity-based rankings and the semantics of relation. PageRank is shown to be effective in file retrieval tasks in non-semantic systems [31]. In our case, we adapt the PageRank algorithm by aggregating all relations between two documents into one single relation, with the weighting score obtained by averaging out all the individual relation weights.

SUPRA: Papadakis et al. [25] proposed combining the activity-based ranking results with a one-step propagation in a layered framework. The relations are constructed simply by identifying documents accessed in the same sessions. In our scenarios, we define the "sessions" to be one unit time step, which is one hour. We only study the MRU decay prior and simple connectivity transition matrix for this baseline, as it is among the best performing variants and requires no parameter tuning. We use the implementation provided by the authors[7].

4.2.2 Parameter Tuning

In all experiments, we set the granularity of the time intervals to be one day. We use MRU as the baseline scoring method for building the trading data. The parameters of the activity-based scoring functions are chosen empirically via grid search with respect to the success rate at 1 for the access prediction task (see Section 4.3). The best performing for each function is as follows. For FEbb, $S = 90$; for FPD, $\alpha = 1.5$; for FWei, $s = 0.9, \alpha = 0.3$. The damping factor is set to $\lambda = 0.25$, as for the standard PageRank as well as for our propagation method. As for the regularization parameter θ, we experiment with a few number of different values and see no significant changes in the performance, possibly due to the small number of dimensions of our relation weight vectors X. We empirically set $\theta = 1$.

4.3 Experiments on Revisit Prediction

The first experiment aims to evaluate how well the system performs in the revisit prediction task, i.e., predicting the likelihood that a document will be accessed by the user in the

[7]http://sourceforge.net/projects/supraproject

Dataset Person		Dataset Collaboration	
Relation(s)	*Description*	*Relation(s)*	*Description*
hasPart (isPartOf)	Relations between container document (folder, albums, etc.) and individual files	page_namespace	relations between a page and a dokuwiki namespace
hasNewerVersion (isNewerVersionOf)	Two revisions of a document	hyperlink	a webpage is linked to other page
hasLocation (isLocationOf)	A document is tagged with a location	attachement	a webpage is attached with a file
hasRecipient (isRecipientOf)	Relations between emails and the recipient		
hasSender (isSenderOf)	Relations between emails and the recipient		
creates (isCreatedBy)	Relations between documents and owners		
isRelatedTo	Two document contents are related		
hasTopic (isTopicOf)	A resource has a topic, which is another resource		
Attribute(s)	*Description*	*Attribute(s)*	*Description*
member	Relations based on shared number of members annotated with the documents	contributors	relations between shared number of contributors to the page
containedThing	Relations based on related Thing instances	TTB	Tagged token-based relations
task	Relations based on tasks tagged to the documents	tag	relations between tags of each dokuwiki page

Table 3: Selected semantic relations used in two datasets Person and Collaboration. The upper part corresponds to the explicit relations, the lower-part corresponds to the attribute-based implicit relations

subsequent time point. This is the well-established task in research on web recommendation [10], personal file retrieval [15], etc. We evaluate the correlation between the predicted rank of a document at a time point t and the real document accesses at the time point $t+1$. Inspired by [19], we employ the following evaluation metrics:

1. *Success at 1* (**S@1**): It quantifies the fraction of time points t (from all time points of study) at which the first-ranked documents according to a ranking method is truly accessed at $t+1$. This resembles the Precision at 1 (P@1) metric in traditional retrieval tasks.

2. *Success at 10* (**S@10**): It quantifies the fraction of documents truly accessed in the next time point, from all documents ranked at top 10, averaging over all time points of study in the micro-average manner (i.e., per-document average).

3. *Average Ranking Position* (**ARP**): This metric starts from the subsequent document access backwards. It computes the average ranking position of accessed documents as produced by a ranking method. The lower the value is, the better the performance of the corresponding ranking system.

For each dataset, we run the burst detection algorithm to identify the "interesting" time points (Section 3.4), resulting in 122 points in the dataset Person and 203 points in Collaboration. We partition each set of time points into the training and testing sets using 5-fold cross validation.

Results. The average results over the two datasets are summarized in Table 4. Among the ranking methods, PageRank has the worst predictive performance. This is because it ignores the recency and frequency signals of the documents. Other interesting observation is that for activity-based ranking methods, adding frequency into the ranking function did not really help in revisit prediction: FMRU performs worse than MRU and FEbb performs worse than Ebb in all metrics, although the differences are not significant. At the first look, this contradicts somewhat to previous findings on the influence of frequency in document ranking [26]. However, analysing deeper, we believe that the cause stems from the fact that a revisiting action typically involves very recent documents, as also argued in [19]. Aggregating recency scores over a time span (10 day-window as in our case) can introduce some documents belonging to different tasks and thus bring more noise to the ranking results. One possible way to solve this is to design a more flexible time window size which adapt to the user's task. We leave this direction to be explored in the future.

Method	S@1	S@10	ARP
MRU	0.162	0.310	76
FMRU	0.131	0.291	87
Ebb	0.213	0.357	65
FEbb	0.193	0.328	70
FPD	0.195	0.331	68
FWei	0.220	0.378	60
PageRank	0.120	0.231	112
SUPRA	$0.320^{\triangle}$	$0.671^{\triangle}$	39
MRU+Prop	$0.353^{\triangle}$	$0.710^{\blacktriangle}$	34
FMRU+Prop	$0.402^{\triangle}$	$0.762^{\blacktriangle}$	30
Ebb+Prop	$0.416^{\triangle}$	$0.733^{\triangle}$	42
FEbb+Prop	$0.452^{\blacktriangle}$	$0.780^{\blacktriangle}$	25
FPD+Prop	$\mathbf{0.512}^{\blacktriangle}$	$\mathbf{0.818}^{\blacktriangle}$	**20**
FWei+Prop	$0.430^{\triangle}$	$0.750^{\blacktriangle}$	40

Table 4: Results on the revisit prediction task. The upper part reports baseline results, the lower part reports results of the proposed system. Symbol $^{\triangle}$ confirms significance against the baseline MRU. Symbol $^{\blacktriangle}$ confirms both significance against the baselines MRU and SUPRA

Compared to the sole activity-based ranking methods, adding propagation shows clear improvements in prediction, starting from the baseline SUPRA. Bringing semantic relations into the propagation improves even further, producing significantly higher performance for all case of temporal priors. The best performing method, propagation with polynomial decay prior, improves the results by 60% as compared to SUPRA. In addition, in contrast to the observed trend in the activity-based ranking, here the combination of frequency and recency with the propagation actually produces better results than the only combination between recency and the propagation. This is because using frequency makes the scores of all documents higher (Equation 3.2), thus enhance the contribution in the propagation point, as there will be more documents with non-zero scores than in the case of

	Dataset Person				Dataset Collaboration			
Method	P@1	P@10	NDCG@10	MAP	P@1	P@10	NDCG@10	MAP
MRU	0.365	0.283	0.219	0.207	0.461	0.375	0.285	0.267
FMRU	0.329	0.307	0.221	0.213	0.457	0.346	0.271	0.258
Ebb	0.407	0.350	0.258	0.218	0.507	0.392	0.287	0.256
FEbb	0.391	0.292	0.217	0.213	0.493	0.357	0.275	0.260
FPD	0.382	0.290	0.214	0.220	0.480	0.400	0.301	0.288
FWei	0.443	0.402	0.324	0.293	0.552	0.424	0.319	0.290
PageRank	0.318	0.251	0.195	0.164	0.388	0.325	0.195	0.204
SUPRA$^{\triangle}$	0.547	0.502	0.426	0.389	0.590	0.469	0.345	0.333
MRU+Prop$^{\triangle}$	0.518	0.456	0.358	0.333	0.561	0.448	0.334	0.340
FMRU+Prop$^{\triangle}$	0.592	0.511	0.431	0.366	0.630	0.493	0.400	0.361
Ebb+Prop$^{\triangle}$	0.615	0.529	0.503	0.481	0.752	0.642	0.501	0.476
FEbb+Prop$^{\blacktriangle}$	**0.728**	0.621	**0.556**	**0.540**	**0.821**	**0.679**	**0.528**	**0.519**
FPD+Prop$^{\blacktriangle}$	0.710	**0.635**	0.523	0.510	0.780	0.667	0.500	0.482
FWei+Prop	0.678	0.575	0.521	0.478	0.715	0.634	0.479	0.460

Table 5: Performances of ranking methods in the user study. Symbols $^{\triangle}$,$^{\blacktriangle}$ indicate the significance test in all scores of the method against MRU and SUPRA respectively.

using recency only. This effect is similar to smoothing in standard information retrieval.

4.4 User-perceived Evaluation

We next aim to evaluate the effectiveness of our proposed system with respect to the user perception and appreciation. We do this by simulating the way users re-access and re-assess the documents in their collections. The experiment is set up as follows. We first ask the user to pick up different time periods of one week length from the past, such that each week covers some prominent events or tasks, and thus manifests considerable amount of user activities on many documents. For example, the user can choose the week when she conducts intensive work on a scientific publication, or on a project review. Within each week of study, we extract the set of documents that draw high attention from the user back in the time. These documents can be chosen manually from the user (e.g., dataset Person), or from the set of highest-frequently accessed document (e.g., dataset Collaboration). Then, the user is presented with the document information and contents[8], and is asked the question "What do you prefer to do with this document as for now ?". The options are:

1. *Pin*: The document is needed for now, I would keep it as short-cut or highlight.
2. *Show*: I would keep the document, but not in the highlight.
3. *Fade*: I would not keep the document.
4. *Trash*: I would delete the document *now*.

Each option corresponds to the user's perception of the current importance of the document, from the highest score (*Pin*) to the lowest one (*Trash*). From the perspective of information retrieval, these can be treated as the relevance feedback, and to that extent, we can use standard IR metrics to evaluate the ranking system.

In the dataset Person, each assessor chose 4 weeks to evaluate. For the dataset Collaboration, 2 assessors are asked to choose 3 weeks per each, all are related to joint events they participated in. The activity history is constructed according to this pair of users. The ranking methods are configured to provide the ranks of documents with respect to the same time step of the user's evaluations. For the Person, we cannot calculate the inter-agreement as the documents to be ranked are usually private. For the Collaboration, the inter-agreement under the Cohen's Kappa is 0.6, suggesting the shared perception of the raters on the evaluated documents.

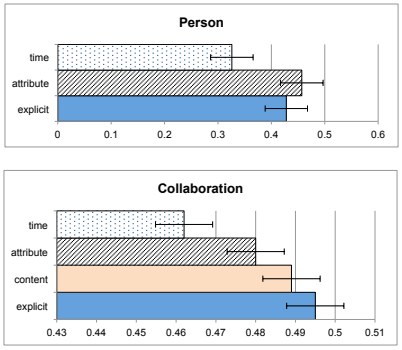


Figure 3: Performance of propagation framework (measured in MAP) with ablated relation sets

Result. The results are summarized in Table 5 for each dataset, as measured as precision, NDCG and MAP scores. The same trend as the prediction task can be observed here: The activity-based ranking methods perform better than PageRank but worse than SUPRA and our propagation variants. Similarly, the frequency-based functions perform worse than the recency ones as isolated methods, but improve the results when combining with the propagation. All propagation methods except the MRU prior-based give higher results than SUPRA. In addition, compared to the prediction task, the performance of all methods in the user-perceived study are slightly higher. This suggests that many documents, although not accessed subsequently, are still deemed "important" to the user. Of the two datasets, methods produce higher performance in the Collaboration than in Person. This can be explained in two ways. Firstly, data in Collaboration is more homogenous, and the model is learnt with respect to the group of users. In contrast, in Person, the

[8]For the Person dataset, the user study is conducted in each computer of the assessor, thereby all information and contents are accessible

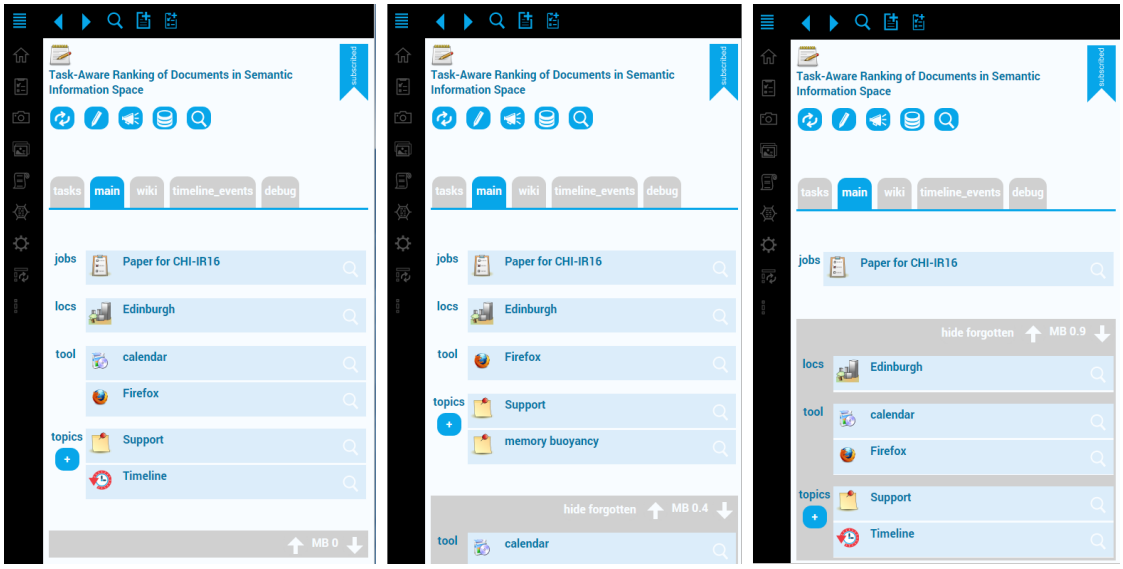


Figure 4: Illustration of decluttering functionality in the Semantic Desktop. From left to right: The list of recommended documents at different cut-off (MB) thresholds, 0 (left), 0.4 (middle), 0.9 (right)

data are highly diverse and the model is learnt over each user, resulting in higher variance level. Secondly, when assessing documents in the collaboration environment, users tend to be more skeptical, and will not likely to assess one document as "Trash" unless completely certain. This results in a higher number of relevant documents than in the case of the dataset Person.

Influence of Semantic Relations.

Next, we study the influence of different types of semantic relations. We use *FEbb+Prop* method as it performs the best in the user study. The evaluation is done via an ablation study: We repeatedly remove from the semantic graph the relations of a certain group (see Section 3.5), then re-execute the framework and re-evaluate using the user study. Finally, we observe the reduction in the performance of the system measured by MAP score (Figure 3). In both datasets, removing time-based relations cause the biggest loss in the performance, suggesting the highest influence of this relation type. This also agrees with the existing findings [19, 33]. In the dataset Person, removal of the explicit relations affects more than removal of attribute-based relations[9]. This suggests the higher contributions of human-defined relations in the dataset. On the other hand, in Collaboration, the attribute-based relations has higher influence than the explicit one. This reflects the characteristics of the dataset, as the tagged tokens are Dokuwiki tags are more representative than the other evidences.

4.5 Application: Reducing Information Overload in PIM

In addition to the empirical experiments, we also demonstrate the effectiveness of our system in real-world usage. We integrated memory buoyancy into the Semantic Desktop infrastructure (Section 4.1.1) used for *decluttering* information in the HTML5 UI of the Semantic Desktop designed to be used on desktop and mobile devices (see Figure 4).

[9]Recall that we could not construct the content-based relations in this dataset

For instance, after days of working in the Semantic Desktop accomplishing various tasks, many things were created in the knowledge base such as new tasks and sub-tasks, notes, new topics, persons, events, annotated web-pages, documents in different versions or only temporary relevant. That means, the user's desktop as well the UI's of the Semantic Desktop start to clutter. Imagine this over several years, lots of once relevant and now irrelevant material is still shown and piles up, i.e., the PIM application is on the verge to build an information overload to its users.

Therefore, we apply the approach presented in this paper to enable the user to focus on the main concepts and resources such as documents of the current attention, thereby to "declutter" the PIM application without manual re-organisation efforts.[10] Figure 4 illustrates this functionality: a note is shown containing a text of this paper which was used to prepare the writing. Now choosing the "Main" tab, the related things are shown categorised according to their supertype (e.g., Location (e.g., City, Building, Room) or Job (Process, Task). Now to focus on the currently most relevant things, only those things are shown above a certain threshold (displayed in the interface as MB or "Memory Buoyancy" in the gray bar; per default this bar is closed, but can be expanded to show "forgotten" things). The user is able to change the cut-off thresholds to show more forgotten things or to focus on most relevant from user interaction. The default threshold is set differently on the desktop (currently 0.5) vs. being on a mobile (currently 0.8) to account for a more focused information provision on a mobile (to reduce cognitive load and data consumption). A second functionality is to propose files to be forgotten (e.g., removing from the desktop and keeping them just in the Semantic Desktop cloud) if they drop below a certain threshold, thus also decluttering the user's desktop. All these things are associated in the semantic graph and the activity history logs. This shows the generalizability of our method: It does not only work with digital files, but can also be applied to any concepts, given the availability of their activity history logs.

[10]see also: https://pimo.opendfki.de/wp9-pilot/#9

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented an adaptive ranking approach for identifying documents that are important to the current focus and task of the user. This contributes to helping the user in navigating and decluttering the growing information spaces. Based on the idea of managed forgetting, our framework unifies evidences from activity logs and semantic relations in a principled way for computing the memory buoyancy of resources. In our method we employ machine learning techniques that automatically learn from the user access history without manual supervision efforts.

Our experiments with two real-world datasets have shown that incorporating the importance propagation via semantic relations between resource significantly improves the performance of the method. As a proof of concept, we have also developed a prototypical system for decluttering personal information space using an existing Semantic Desktop.

This work is just the first step towards realizing managed forgetting. We plan to extend our approach to better tailor it to particular scenarios, such as navigation on desktop or mobile phones. We are also planning a more in depth user study in order to better understand the user expectation in several dimensions such as interactions and injecting human preferences (For instance, in which scenarios or domains the activity-based system works better than the activation, etc.) Other direction is the consideration of more complex user tasks in the learning model, for instance, investigating cross-device tasks or recurrent tasks.

Acknowledgements. This work was funded by the EU FP7 project ForgetIT (Grant No. 600826)[11].

6. REFERENCES

[1] B. Adrian, M. Klinkigt, H. Maus, and A. Dengel. Using iDocument for document categorization in Nepomuk Social Semantic Desktop. In *I-SEMANTICS*, pages 638–643, 2009.

[2] J. R. Anderson and L. J. Schooler. Reflections of the environment in memory. *Psychological Science*, 6(2):396–408, 1991.

[3] M. Awad, L. Khan, and B. Thuraisingham. Predicting www surfing using multiple evidence combination. *The VLDB Journal*, 17(3):401–417, 2008.

[4] L. Backstrom and J. Leskovec. Supervised random walks: Predicting and recommending links in social networks. In *WSDM*, pages 635–644, 2011.

[5] O. Bergman, R. Beyth-Marom, and R. Nachmias. The user-subjective approach to personal information management systems. *JASIST*, 54(9):872–878, 2003.

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

[7] T. Blanc-Brude and D. L. Scapin. What do people recall about their documents?: Implications for desktop search tools. In *IUI*, pages 102–111, 2007.

[8] J. Chen, H. Guo, W. Wu, and W. Wang. iMecho: An Associative Memory Based Desktop Search System. In *CIKM*, pages 731–740, 2009.

[9] 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.

[10] M. Connor and S. Spitzer. The places frecency algorithm. https://developer.mozilla.org/en-US/docs/Mozilla/Tech/Places/Frecency_algorithm, (Accessed Aug 31, 2015).

[11] M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. *ACM TOIT*, 4(2):163–184, 2004.

[12] 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*, pages 72–79, 2003.

[13] D. Elsweiler, M. Baillie, and I. Ruthven. Exploring memory in email refinding. *ACM TOIS*, 26(4):21, 2008.

[14] D. Fisher and P. Dourish. Social and temporal structures in everyday collaboration. In *SIGCHI*, pages 551–558, 2004.

[15] S. Fitchett and A. Cockburn. AccessRank: Predicting what users will do next. In *SIGCHI*, pages 2239–2242, 2012.

[16] S. Fitchett, A. Cockburn, and C. Gutwin. Improving navigation-based file retrieval. In *SIGCHI*, pages 2329–2338, 2013.

[17] J. Gemmell, G. Bell, and R. Lueder. Mylifebits: a personal database for everything. *CACM*, 49(1):88–95, 2006.

[18] W. Jones. *Keeping Found Things Found: The Study and Practice of Personal Information Management*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.

[19] R. Kawase, G. Papadakis, E. Herder, and W. Nejdl. Beyond the usual suspects: Context-aware revisitation support. In *ACM Hypertext*, pages 27–36, 2011.

[20] J. Kleinberg. Bursty and hierarchical structure in streams. *KDD*, 7(4):373–397, 2003.

[21] D. C. Liu and J. Nocedal. On the limited memory BFG-S method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[22] R. H. Logie. The functional organization and capacity limits of working memory. *Current Directions in Psychological Science*, 20(4):240–245, 2011.

[23] H. Maus, S. Schwarz, and A. Dengel. Weaving personal knowledge spaces into office applications. In *Integration of Practice-Oriented Knowledge Technology: Trends and Prospectives*, pages 71–82. Springer, 2013.

[24] C. Niederée, N. Kanhabua, F. Gallo, and R. H. Logie. Forgetful digital memory: Towards brain-inspired long-term data and information management. *ACM SIGMOD Record*, 44(2):41–46, 2015.

[25] G. Papadakis, R. Kawase, E. Herder, and C. Niederée. A layered approach to revisitation prediction. In *ICWE*, volume 6757, pages 258–273. 2011.

[26] M.-H. Peetz and M. De Rijke. Cognitive temporal document priors. In *ECIR*, pages 318–330. Springer, 2013.

[27] D. Quan, D. Huynh, and D. R. Karger. Haystack: A platform for authoring end user semantic web applications. In *ISWC*, pages 738–753. 2003.

[28] M. Ringel, E. Cutrell, S. Dumais, and E. Horvitz. Milestones in Time: The value of landmarks in retrieving information from personal stores. In *INTERACT*, volume 2003, pages 184–191, 2003.

[29] L. Sauermann, A. Bernardi, and A. Dengel. Overview and Outlook on the Semantic Desktop. In *Proceedings of the First Semantic Desktop Workshop at ISWC'05*, 2005.

[30] L. Sauermann, L. van Elst, and A. Dengel. PIMO – A Framework for Representing Personal Information Models. In *I-SEMANTICS*, J.UCS, pages 270–277, 2007.

[31] C. Soules and G. Ganger. Connections: Using context to enhance file search. In *SOSP*, pages 119–132, 2005.

[32] J. Teevan. How people recall search result lists. In *CHI (Extended Abstract)*, pages 1415–1420, 2006.

[33] T. Watanabe, T. Kobayashi, and H. Yokota. Fridal: A desktop search system based on latent interfile relationships. In *Software and Data Technologies*, pages 220–234. 2013.

[34] R. W. White, P. N. Bennett, and S. T. Dumais. Predicting short-term interests using activity-based search context. In *CIKM*, pages 1009–1018, 2010.

[11] http://www.forgetit-project.eu/