What Belongs Together Comes Together. Activity-centric Document Clustering for Information Work.

Alexander Seeliger, Benedikt Schmidt, Immanuel Schweizer, Max Mühlhäuser
Technische Universität Darmstadt
Darmstadt, Germany
{seeliger, schmidt, schweizer, max}@tk.tu-darmstadt.de

ABSTRACT

Multitasking and interruptions in information work make frequent activity switches necessary. Individuals need to recall and restore earlier states of work which generally involves retrieval of information objects. To avoid resulting tooling time an activity-centric organization of information objects has been proposed. For each activity a collection with related information objects (like documents, websites etc.) is created to improve information access and serve as a memory aid.

While the manual maintenance of such information collections is a tedious task and becomes an interruption on its own, the automatic maintenance of such collections using activity mining is promising. Activity mining utilizes interaction histories to extract unique activities based on the stream of interaction with information objects. For activity mining, existing work shows varying success in limited study setups.

In this paper, we present a method for activity mining to generate activity-centric information object collections automatically from interaction histories. The technique is a hybrid approach considering all information types used in previous work – activity stream and accessed content related information. Method performance is evaluated based on interaction histories collected during real work data from eight information workers collected over several weeks. For the dataset our hybrid approach shows on average a performance of 0.53 ARI up to 0.77 ARI, outperforming single metric-based approaches.

Author Keywords

Activity Mining; Document Organization; Information Work

ACM Classification Keywords

H.3.3. Information Search and Retrieval: Clustering

IUI 2016, March 7–10, 2016, Sonoma, CA, USA.

ACM 978-1-4503-4137-0/16/03 ...$15.00.
http://dx.doi.org/10.1145/2856767.2856777

INTRODUCTION

Information work is a challenging work type, characterized by multitasking, frequent interruptions and information overload [20]. During the last decades the execution of information work changed with the advent of a variety of assistive technologies and tools. Notebooks, smart phones and tablets provide access to various networks, and therefore to services enabling advanced means of collaboration and information interaction, anywhere, anytime. Despite this transformation, the characteristics and challenges of information work remained unscathed. In fact, the amount of available information permanently increases [25] and with more devices and more apps there are more sources for interruptions.

The interruption with a task switch is a specific challenge for information work. In this moment, the information worker needs to concentrate on continuing an earlier task and will restore a work environment. Apps need to be opened or closed and information need to be retrieved. Thus, mental and physical *tooling time* (time to get back to the work environment before interruption) emerges. The overall process increases the probability of prospective and retrospective memory failures. Efficiency drops and mistake likelihood increases [1].

To address the challenges of interruptions and activity switches a myriad of personal information management tools have been proposed. One direction is activity-centric information management which organizes information based on activities, the information is used in [26, 2, 7, 12]. Such lists not only improve the access of relevant information but also serve as a memory aid for the prospective and retrospective memory. However, most activity-centric information management approaches require manual work. Individuals create activity objects (sometimes also referred to as tasks) and assign information objects. This is a tedious process which can be an interruption on its own, resulting in a neglected and outdated activity lists.

An alternative is the automatic detection of activities from interaction histories to cluster those information objects which belong to the same activity – a type of activity mining. In this paper, we focus on such activity mining approaches. Systems like CAAD [19], Swish [16] or Transparency [22] track the interaction of a user with information and use the resulting interaction histories to automatically create document clusters which belong to different activities. These systems use interaction history data like access times, access frequencies and access sequences to cluster information belonging to similar activities. This activity mining is a complex problem, because multitasking and interruptions disguise the borders between different activities. A user might search a plane for a business trip while reporting project data with a messenger application. Despite the complexity, existing work shows promising results using (1) activity-based and (2) content-based properties. Unfortunately, there exists no comparison of the usefulness of different property types for activity mining (activity-centric document clustering). Furthermore, all considered approaches use limited study setups, frequently using lab setups. No long term studies which show the performance of such approaches with real data exist at all.

In this paper, we provide two main contributions:

1. An activity mining method for the automatic clustering of activity related information based on interaction histories, using a mixture of activity and content based properties is presented. The method analyzes an interaction history and identifies activities and the information objects used in the activity, even if the activity was interrupted or executed in parallel with other activities. Our method considers activity and content based properties we found in the related work and implements mechanisms to reason about the relevance of the properties for activity-centric document clustering.

2. While related work only contains small studies with data of few hours up to few days [19, 16, 22, 5], we provide an evaluation of our approach with raw data from 8 information workers with 21 up to 47 work day interaction histories, each. Furthermore, we systematize information from the existing body of work on activity-centric document clustering and show the relevance of different activity and content properties. The overall dataset includes 28,045 access events on 7,968 different documents. Our evaluation shows that activity mining as activity-centric document clustering provides useful results to support information workers during their workday.

The paper is structured as follows. First, we introduce related work. Second, we introduce the interaction history properties we use to cluster documents. Third, we specify our activity-centric document clustering method using a hybrid mix of activity and content based properties. Next, we describe our long term study, evaluate our method and discuss the results. Finally, we conclude with a discussion and a short summary.

RELATED WORK

One direction for the support of information work is activity-centric task (or activity) management, using task lists which collect information objects related to the task. These lists require the manual creation and maintenance of the tasks and collections. An early example is Groupbar [26] which provides users with a task-aware window manager. The activity-centered task assistant [2], working spheres [7] and UMEA [12] are just a few examples of such tools with a manual maintenance of task lists with related information objects, utilizing support e.g. from interaction histories or specific visualizations.

Apart from the manual support types, different semi-automatic research prototypes for an activity-centric support of information work exist. Recently, Laevo [11] was introduced as a semi-automatic desktop interface for organizing window configurations of three knowledge work processes (archiving, multitasking, planning). With TaskTracer [24], a semi-automatic approach with automatic detection of task switches was realized. TaskPredictor [24] automatically assigns used documents to tasks. It assumes, that the behavior of users is a mixture of different activities and that each activity is related to a set of information objects. By permanently observing the users actions (document accesses, window focuses, webpage navigation etc.), new assignments can be predicted.

Other work focuses on the automatic creation of activity-centric document clusters from interaction histories. A first example is the Groupbar extension SWISH [16]. SWISH aims at the automation of task-specific window groupings from an interaction history with recorded transitions between windows, considering the window titles. The automatic maintenance of tasks with attached information objects was focused by the Context-Aware Activity Display (CAAD) [19] and the Transparency tool with hierarchical task instance mining [22]. CAAD traces the users behavior and tries to detect behavioral changes. A pattern mining algorithm analyzes the users workflow of used program names and accessed documents to generate context structures which are activity-specific. The hierarchical mining approach uses a simple hierarchical clustering based on semantic similarity of the accessed document content to identify task structures. Brdiczka [5] performs activity mining based on document usage patterns by clustering access events of documents up to a threshold. The evaluations of the automatic approaches are rather limited. Oliver et al. report for SWISH [16] performance values for two data sets. For a data set of one user and five tasks, collected over approximately four hours they report an F-measure of 0.58. The results are improved to a recall of 0.76 by using 1 hour chunks of data and application. The CAAD system by Rattenbury et al. [19] was evaluated by a usefulness study over 1 week usage of 7 persons. Schmidt et al. with hierarchical task instance mining [22] evaluated task execution in a lab simulation with 8 persons working on 7 complex, frequently interrupted tasks. They generated a gold-standard and showed that their algorithm has an F-measure of 0.72 in comparison. Brdiczka [5] evaluates the work data for 3 days of work of 10 persons. His result is an F-measure of 0.32 with a precision of 0.20 for 50 tasks. By limiting the data set to the six most frequent tasks Brdiczka obtains an F-measure of 0.74. All approaches use a different selection of activity and content properties to create activity clusters.

The results show that the automatic detection of activities with related information objects is promising. All approaches we are aware of use activity properties (access sequences, durations, etc.) or content properties (document content, window titles) from interaction histories and use clustering-like techniques. However, none existing work makes use of the combination of content and activity properties. All approaches are evaluated in different setups and show very different results. There exists no comparison of the approaches or the used properties. Furthermore, the evaluations consider only short time spans between few hours up to one week.

ACTIVITY AND CONTENT BASED PROPERTIES FOR DOCUMENT CLUSTERING

In this paper, we develop a method for the automatic clustering of activity related information based on interaction histories. The state of the art has shown that two types of properties are frequently used to create such clusters: (1) activity-based and (2) content-based properties. Activity-based properties have a wider look at how documents are related to each other during activity execution, using environmental and behavioral aspects. Content-based properties just focus on the concrete content of the corresponding document, thus implying that information objects which belong to the same activity also are related on content level.

Defining a Document Relatedness Metric

Before defining concrete document relations, a generic function that expresses relatedness between documents is needed. We define relatedness as a function that consumes both documents A and B and returns a similarity value that is bound between 0 and 1. This function is a *semi-metric* on a document corpus D and is defined as

$$rel : D \times D \rightarrow \mathbb{R} \in [0, 1]$$

and fulfills the following properties:

1. $rel(A, B) \geq 0$ (non-negativity)
2. $rel(A, B) = 0$ if $A = B$
3. $rel(A, B) = rel(B, A)$ (symmetry)

Low values of $rel(A, B)$ indicate that there exist a high relatedness between document A and B. High values indicate that both documents are not related to each other. Most notably, we are interested in activity relatedness which means that two documents are related if they belong to the same activity.

Defining Document Relations via Activity Properties

The first category of document properties tries to extract activity knowledge from interaction histories of documents. By recording the behavior of users working with documents, correlations between documents can be mined.

Before actually defining concrete activity properties, a short introduction to the work execution process of the information worker is given to better understand the difference between activity and task. Each information worker has several goals to achieve but limited capabilities and resources require them to pick a subset of goals based on priorities. To achieve a goal the prospective work is considered as a task which structures and specifies the upcoming activities. The execution of a task is performed by an activity. Activities then transform an object into an outcome which may be interpreted as the achievement of the selected goal. The user decides when a created outcome fulfills the goal [21].

We will focus ourselves on interaction histories of information workers using modern operating systems which follow a WIMP[1]-style interaction paradigm, allowing multiple applications to run at the same time. In the following we present information which can be collected from such a system, using an application which tracks user-system interactions and creates an interaction history. The tracking application we used for our work is presented later in this paper, here we focus on the general information types we have identified from our research and the related work, to be relevant information sources to be used for activity-centric document clustering.

Application Usages

Switching between windows (a change of the active foreground window) indicate an application switch which can – but not necessarily is – an indicator for an activity switch. While the switch can mean a change of focus, a set of applications can also be used together to realize a specific outcome (e.g. a web search and document editing to create a survey). Still, the active window is an important indicator of the awareness.

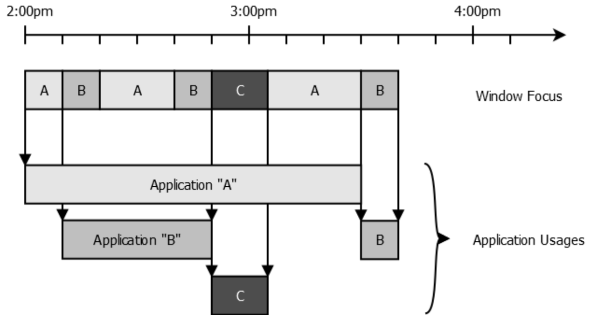


Figure 1. Single window switches will be merged together if the interval between those switches is less than 30 minutes. This merging results in multiple parallel time-lines.

Realization: The use of foreground information helps to determine active application usage times. We assume that a window not necessarily becomes irrelevant once it is hidden. If the resumption time is smaller than 30 minutes, we merge active window times together (see Figure 1). Iqbal and Horvitz [10] have conducted a user study, measuring the probability of task resumption depending on the time spent before the suspension and the resumption time. In this study, the probability for returning to a task within 30 minutes is at least 70%.

A1: Context Applications

Some activities may require the user to use the same subset of applications over and over again. So it is obvious to use all currently used applications as an indicator for the activity relatedness between documents.

Realization: Context applications can be modeled using a vector U containing the current state of each application a_i at a given time t. $U_t = (a_1, a_2, ..., a_A)$. If the application i is used $a_i = 1$ and otherwise $a_i = 0$. When multiple document accesses have been recognized, we can compare all seen application usages for each visit and get a better prediction which applications are really specific for the compared documents.

[1] Abbrv.:Windows, Icons, Menus, Pointers

A2: Time-Related (1): Mostly used together

Often multiple documents are used together, like presentations and notes. We can make use of this and group documents together, when they are used together multiple times.

Realization: Besides collecting the application usage intervals, document accesses are also collected. Documents d_1 and d_2 can then be compared by calculating the proportion of the number of application intervals S_{d_i} in which both documents have been accessed to the number of application intervals in which only d_1 or d_2 was accessed.

$$rel(d_1, d_2) = 1 - \frac{2 \cdot S_{d_1 \cap d_2}}{S_{d_1} + S_{d_2}}$$

This formula returns high relatedness between documents d_1 and d_2, if the number of application usages in which both documents have been accessed is higher than the number of documents in which only one of the documents has been accessed.

A3: Time-Related (2): Parallel Work

Another indicator for activity-relatedness between two documents is parallel work and the according permanent document switching (i.e. keyboard shortcut *Alt-Tab*).

Realization: The calculation of the relatedness for parallel work between d_1 and d_2 with the access times $U_i = (a_{i,1}, a_{i,2}, ..., a_{i,A})$ is defined as the following:

$$rel(d_1, d_2) = 1 - \frac{\sum_{i=1}^{|U_1|} \sum_{j=1}^{|U_2|} \theta(a_{1,i}, a_{2,j})}{|U_1| \cdot |U_2|}$$

with

$$\theta(t_1, t_2) = \begin{cases} 1 & \text{if } abs(t_1 - t_2) < 5 \text{ minutes} \\ 0 & \text{otherwise} \end{cases}$$

We assume parallel work with documents when the time span between their document accesses is less than 5 minutes. Because the metric delivers high relatedness values for a small number of access times, it only delivers values for documents with at least two access times.

Defining Document Relations via Content Properties

Until now we have only highlighted activity-centric properties based on interaction histories as relatedness metrics. In this section we will have focus on the content of documents. Documents may have the same topic and use the same vocabulary. If documents address a similar topic it can be an indicator that the documents are used in the same activity. So comparing the content of documents is the most obvious approach to determine activity relatedness.

C1: TF-IDF

A well-known content similarity measure in information retrieval is TF-IDF [17]. The TF-IDF is the product of term frequency and inverse document frequency. Term frequency $tf(t, d)$ is defined as the number of occurrences of the term t in the document d. Inverse document frequency is defined as the importance of a word to distinguish it from a common word. Then the TF-IDF is defined as

$$\mathsf{tf} - \mathsf{idf}(t, d, D) = \mathsf{tf}(t, d) \cdot \mathsf{idf}(t, D)$$

To compare two documents, the normalized $\mathsf{tf} - \mathsf{idf}$ factor [15] is calculated for each term and each document, resulting in a document-term matrix (TD-matrix). Now the distance between two documents can be computed by comparing its $\mathsf{tf} - \mathsf{idf}$ vectors using the cosine similarity.

C2: Latent Semantic Analysis

Latent Semantic Analysis [14] (LSA) is a statistical method based on the TD-matrix. After building the term-document-matrix a *Singular Value Decomposition* (SVD) [3] is applied. SVD decomposes the matrix into the product of three other matrices, thus we can read the singular values, which characterize the importance of a term in the document corpus. We can then determine which dimensions can be reduced by setting less relevant singular values to zero and recompute the TD-matrix.

In comparison to the TF-IDF metric LSA does not compare the occurrences of words, it compares higher level semantic structures. The reduction of the TD-matrix generalizes conceptual equal terms and removes less important terms. This generalization let LSA solve the problem of synonymy.

C3: Latent Dirichlet Allocation

Blei et al have presented another document classification approach for large collections called *Latent Dirichlet Allocation* [4] (LDA). It is a generative probabilistic three-level hierarchical Bayesian model in which a document is modeled as a random mixture of latent topics. A topic is characterized by a distribution over a fixed vocabulary. In a probabilistic model data is treated as observations that come from a generative probabilistic process including hidden variables.

The goal of LDA is to infer the unknown topic structures from the distribution of words in the documents. The result of LDA is the topic distribution of a document, which can then be compared using a similarity metric over distributions. We use the *JensenShannon* divergence which is based on the *Kullback-Leibler* divergence with some differences. It is symmetric and always delivers a finite value. Furthermore, it is bounded by 1.

DOCUMENT CLUSTERING METHOD USING A HYBRID FEATURE SET

In the following, we present our method for activity-centric document clustering. Until now a couple of relatedness metrics (see table 1) have been presented which can all be used standalone for document clustering. However, using only single metrics could lead to misleading results. Only focusing on activity data will not be able to identify multitasking (e.g. parallel reporting about an activity and programming will be grouped together). Only focusing on document similarity will ignore relatedness which is not on a topic level (e.g. booking a flight and calculating the cost in parallel). To address these issues, our approach focuses on the combination of multiple relatedness properties. A mixture of activity and content related properties will be used at the same time (hybrid feature set). As it turns out, combining multiple document properties to calculate relatedness is not an easy task, especially, when trying to combine different property types, which use different information sources. A first step was already done by

#	Property	Type	Description
A1	**Context Applications**	*Activity*	The applications which are used indicate which activity is currently performed.
A2	**Mostly Used Together**	*Activity*	Documents which are used together may have a relationship between each other.
A3	**Parallel Work**	*Activity*	Documents which are used parallel may have a relationship between each other.
C1	**TF-IDF**	*Content*	TF-IDF calculates the similarity between documents by comparing word occurrence.
C2	**Latent Semantic Analysis (LSA)**	*Content*	Similar to TF-IDF but comparing higher semantic structured.
C3	**Latent Dirichlet Allocation (LDA)**	*Content*	LDA calculates the similarity of documents by comparing topic distributions.

Table 1. Summary of all presented activity and content properties.

defining all relatedness metrics the same way and normalize the outcome. This makes sure that calculating the sum of two metrics does not deliver a value which does not represent the relatedness between documents anymore. This also avoids complicated transformations for the different approaches.

A very easy method to combine multiple document properties is to use the average value of the calculated metrics. However, this approach has two main issues: (1) the quality of the different metrics may vary and (2) the value distribution over the metrics is different. Both challenges need to be addressed.

1. Although it seems quite obvious to give the different relatedness metrics the same amount of influence to the overall combined metric, it can be somehow problematic and counterproductive. Depending on the domain, the amount and the quality of documents, some relatedness metrics may work better than others. To uncover hidden structures of the documents, equal influence for all relatedness metrics is not the best solution. A weighting factor can be used to overcome this issue:

$$rel_{total}(d_j, d_{j'}) = \sum_{i=0}^{n} \rho_i \cdot rel_i(d_j, d_{j'}) \text{ with } \sum_{i=0}^{n} \rho_i = 1$$

   The weighting factor ρ_i controls the influence of each relatedness value rel_i to the overall combined metric. This weighting can be used to give lower quality metrics less influence but high quality metrics a higher influence.

2. When each weight ρ_i is set to the same value ($\rho_i = \frac{1}{n} \forall i$), that does not necessarily give all relatedness metrics equal influence. The influence of the i-th metric of $sim(d_j, d_{j'})$ depends on the relative contribution to the average document relatedness over all pairs [8]. Some metrics may have in average lower or higher values than others. This difference has an impact on the relative proportion of the corresponding metric. To overcome this difference, a normalization factor is calculated from the average contribution:

$$rel_{total}(d_j, d_{j'}) = \sum_{i=0}^{n} rel_i(d_j, d_{j'}) \cdot \frac{\rho_i}{\overline{d}_i}$$

   with

$$\overline{d}_i = \frac{1}{N^2} \sum_{j=0}^{N} \sum_{j'=0}^{N} rel_i(d_{ji}, d_{j'i})$$

   The relative influence of the i-th relatedness metric without the correction is $\rho_i \cdot \overline{d}_i$. Thus it is necessary to correct $rel_i(d_j, d_{j'})$ with the factor $\frac{1}{\overline{d}_i}$. For example, if the influence of all relatedness metrics should be equal, the weighting has to be set to $\rho_i = 1/(n \cdot \overline{d}_i)$.

The relevance of both issues is even higher when combining different types of relatedness metrics – activity and content relations. To address these issues optimal weighting parameters depending on the collection of documents must be found and each metric needs to be normalized so that the relative contribution of each metric is adjusted.

Finding the optimal weightings is a complex task because of the independence of each metric and the use of different data sources for calculating relatedness between documents. Fixed parameters for specific domains may not result in good results for every document corpus because in most cases documents have different domains and topics. Even the behavior of each information worker is different and everyone has an own personal bias for organizing documents. It is needed to develop a solution that finds weighting parameters for each document corpus and information worker individually.

Finding Optimal Parameters for Weighting and Clustering

The finding of the optimal weighting parameters can be formulated as an optimization problem which optimizes the generated activity-centric document clusters. Each weighting parameter $\rho_0, \rho_1, ..., \rho_n$ is a variable in the optimization problem which controls the outcome of the clustering. The number of variables depends on the number of metrics that will need to be combined together. The bounds of the variables are the following:

$$\rho_0, \rho_1, ..., \rho_n \in [0, 1] \qquad \text{with} \qquad \sum_{i=0}^{n} \rho_i = 1$$

as the side condition.

We will use an optimization problem solver with continuous variables which tries to find the best solution from all feasible solutions. The optimization function looks like the following:

$$f(\rho_0, \rho_1, ..., \rho_n, D) \to \mathbb{R} \qquad \text{with} \qquad \sum_{i=1}^{n} \rho_i = 1$$

where f is the document clustering function which returns the quality of the activity-centric document groups and D the

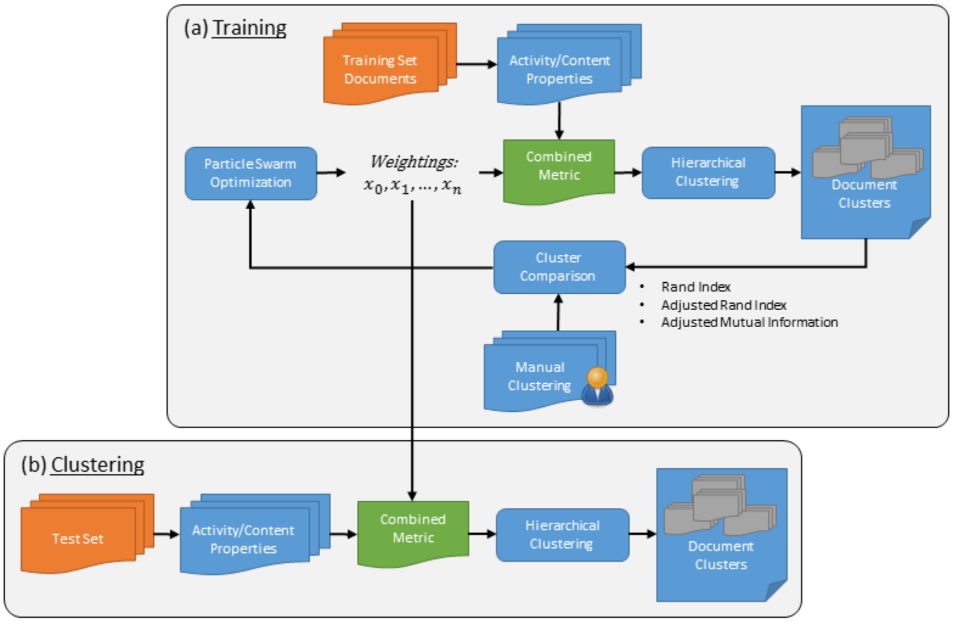


Figure 2. This figure shows the overall process of activity-centric document clustering for information work. The process is twofold: (a) the training phase calculates the weighting parameters for combining the different document properties using a small manual clustered document part and the Particle Swarm Optimization and (b) the actual clustering using the learned parameters from training.

document corpus. The optimization problem optimizes f to its maximum, which means it optimizes the activity-centric document clusters. We define f as the function which returns a value between 0 and 1 and determines if the calculated document clusters have the optimal structure. We let the user manually cluster a small part of the given document corpus and then use this information as a desired result for the clustering of these documents. That means that the system tries to adjust the weighting parameters to exactly match the manual document clusters using the optimization solver. We can then say, that f is a function which returns how much the calculated and the manual assigned clusters differ from each other.

The optimization problem can be solved by an optimization algorithm which will return the best parameter values $\rho_0^*, \rho_1^*, ..., \rho_n^*$.

Particle Swarm Optimization

The Particle Swarm Optimization [13] (PSO) is a computational algorithm which tries to find an optimum by iteratively improving a candidate solution given a quality function. PSO is an evolutionary algorithm trying to improve the quality of the solution stepwise. It is initially inspired by bird flocking, especially, the group dynamics of the birds' social behavior. Reynolds and Heppner argued that the synchrony of flocking requires each bird to maintain an optimum distance between its neighbors. A fundamental hypothesis of PSO is that "social sharing of information among conspeciates offers an evolutionary advantage" [13, 29].

Based on both assumptions PSO works as the following: a population of candidate solutions (particles) moves around in the search-space. The movement is determined by a formula using the particle's position and velocity. It is influenced by the local best known position and better positions found by other particles. This process is repeated over and over again until the maximal number iteration is reached or a given error threshold was passed. In result the swarm is moving towards the best solution but there is no guarantee that a satisfying solution is found.

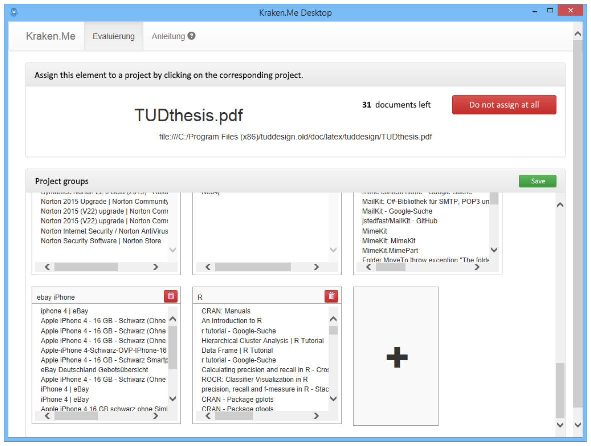
Creating Activity Related Document Groups

The last step is to create the activity-centric document clusters by performing the actual clustering. For each document pair a feature vector is calculated using the different relation properties presented in this paper. The calculated feature vector is then normalized using the given average contribution value and combined to a single value given the calculated weighting parameters. The result of the combined value for each document pair can be easily represented as a relatedness matrix. The final step is to use the calculated matrix to generate activity-centric document groups.

The *Agglomerative Hierarchical Clustering* algorithm [30] (HAC) is used to cluster documents into activity-centric document groups given a relatedness matrix. It works as follows: at the beginning each document has its own cluster. Document clusters with the highest relatedness values are then merged together until a given threshold has been reached. We have decided to use HAC instead of K-means because we do not know the number of clusters beforehand. Furthermore, HAC has shown significant better results than K-means with varying k in our tests.

EVALUATION

This paper provides a method for the automatic clustering of activity related information based on interaction histories, by combining activity and document properties. In this section a user study will be presented which shows how the presented



Assign this element to a project by clicking on the corresponding project.

TUDthesis.pdf

31 documents left

file:///C:/Program Files (x86)/tuddesign.old/doc/latex/tuddesign/TUDthesis.pdf

Project groups

Symantec Norton 22.0 Beta (2015) - Roku
Norton 2015 Upgrade | Norton Community
Norton 2015 (V22) upgrade | Norton Comi
Norton 2015 (V22) upgrade | Norton Comi
Norton Internet Security / Norton AntiVirus
Norton Security Software | Norton Store

Neo4j

mime content name - Google-Suche
MailKit: C#-Bibliothek für SMTP, POP3 un
MailKit - Google-Suche
jstedfast/MailKit · GitHub
MimeKit
MimeKit: MimeKit
MimeKit.MimePart
Folder MoveTo throw exception "The folde

ebay iPhone
iphone 4 | eBay
Apple iPhone 4 - 16 GB - Schwarz (Ohne
Apple iPhone 4 - 16 GB - Schwarz (Ohne
Apple iPhone 4 - 16 GB - Schwarz (Ohne
Apple-iPhone 4-Schwarz-OVP-IPhone-16
Apple iPhone 4 - 16 GB - Schwarz Smartp
eBay Deutschland Gebotsübersicht
Apple iPhone 4 - 16 GB - Schwarz (Ohne
iPhone 4 | eBay
iPhone 4 | eBay
Apple iPhone 4 16 GB schwarz ohne Siml

R
CRAN: Manuals
An Introduction to R
r tutorial - Google-Suche
Hierarchical Cluster Analysis | R Tutorial
Data Frame | R Tutorial
r tutorial - Google-Suche
Calculating precision and recall in R - Cros
ROCR: Classifier Visualization in R
precision, recall and f-measure in R - Stac
CRAN - Package gplots
CRAN - Package gtools

Figure 3. Regularly, the participant of the user study is asked to assign collected documents into activity-centric document groups. The user is shown a document which should be assigned to a group. By clicking on a document group, the shown document is assigned. Furthermore, the user can give each group a short description. If a document is not relevant at all, the document can be marked as irrelevant. Finally, documents can be moved from one group to another.

system (see figure 2) performs on real world data (no predefined scenario nor a restriction was given).

Study Design

Our participants installed a tool to collect interaction histories on their work machines. For this purpose, we used Kraken.me [23]. Kraken.me is a multi-device user tracking suite which offers a variety of different tools for tracking, storing and analyzing all kinds of data on multiple devices. For our purpose Kraken.me was used to generate the interaction histories without any user interruption.

For calculating the performance of the predicted activity-centric document groups, it is needed to have a standard against which the prediction is tested. During the evaluation the user will be asked by the desktop client once a day to assign recorded documents into groups (see figure 3). The user has to manually create new document groups as needed, and give these groups a short description. The user can also decide to remove a document from the list when it is not related to any created group. The manual assignment by the user will then be used as a gold standard to measure the performance of the automatic document clustering algorithm.

Figure 3 shows how the evaluation window looks like and what information is shown to the user. In case of a website, the user sees the title and the URL of the website. In case of a file, the file name and the absolute storage path is shown. When the user wants to create a new group, the user can click on the plus symbol to create one. A title for each activity related document group can be given. The shown document on the top can be assigned by simply clicking on the corresponding document group. Additionally, the user can see the remaining number of documents.

Participants

In the user study eight (8) male persons have participated. All study participants use the computer every day, are technical versed and can be considered as knowledge workers. The following table shows the age structure of the study participants:

Age group	*21 - 30*	*31 - 40*	$\geq$ *60*
Participants	6	2	1

Table 2. Age structure of the user study participants.

Timeframe for the Evaluation

Getting enough data for the document clustering algorithm is essential for evaluating the performance with real world data. The collected data must at least reveal multiple activity-centric document groups, so a longer period of usage is needed. Also activity patterns are only getting visible when the user is observed over a longer time. Finally the evaluation was performed from 15. September to 31. October 2014.

Not all participants had time to participate for the full time of the study. From 3 participants (P1, P2, P3) we have data from 47 work days, P4 and P5 have delivered data from 40 work days, P6 delivered data from 26 days and P7 and P8 have participated for 21 work days.

Collected Data

As seen in table 3, we have collected $28,045$ data items in total. From the 8 participants $7,968$ unique documents have been tracked with $15,129$ accesses. Finally $3,902$ software usage intervals were recorded. We did no data cleaning at all, only participants could mark documents as not related.

In this study we have collected a large number of real usage hours. We distinguish between real and parallel usage times. Real usage times describe the active working time on the computer when applications are open. Parallel application usages are the summed usage hours of applications running at the same time, for example when working with Excel and PowerPoint at the same time, the usage times are multiplied by two. Because we monitor each active window change using the foreground process changed event, we can also measure the number of application switches. In total we have monitored $18,336$ application switches over all study participants, so in average each user has switched applications $2,292$-times. It is interesting to see the high average number of application switches of 30.3 times per hour.

Finally, we have calculated the average active duration of a single application window. According to table 3 each application is shown in the foreground for about 7.5 minutes in average. When users are often switching between applications this number is lower than when users are working on one application for a long time.

Performance Measurements

The performance of the presented system can be calculated by comparing the predicted activity related document groups with the gold standard. We calculate the differences between them using cluster comparison measurements. There exist multiple performance measurements for this task:

- **Rand Index** [18] uses a pairwise comparison instead of looking if a single object was classified correctly. Rand

User	Sum	*P1*	*P2*	*P3*	*P4*	*P5*	*P6*	*P7*	*P8*
Documents	*7,968*	2,114	1,197	1,321	588	250	21	322	2,155
Doc. visits	*15,129*	2,985	2,633	1,320	1,251	628	205	496	5,611
Software usages	*3,902*	971	567	1087	261	116	529	134	237
Usage time (real) in hours	*1,113.3*	71.7	79.1	678.8	56.6	15.5	35.2	75.5	100.8
Usage time (parallel) in hours	*2,665.4*	939.7	240.6	888.3	145.7	7.9*	138.4	109.7	195.0
Avg. app usage in minutes	*7.5*	15.5	4.1	13.6	5.2	2.9	3.4	9.8	5.1
Switches	*18,336*	3,634	3,530	3,926	1,684	165	2,455	669	2,273
Switches per h	*30.3*	50.7	44.6	5.8	29.7	10.7	69.7	8.9	22.6

Table 3. Statistics of the collected data set grouped by participants. *: Parallel usage hours for this participant could not be calculated correctly due to an installed application that influenced our observer.

Index has a range from 0 (the predicted classification is not equal to the true classification) to 1 (predicted and true classification are the same). This measure has the disadvantage that the result is highly dependent on the number of clusters and the measure convergence to 1 as the number of clusters increase [28, 6].

- The **Adjusted Rand Index** [9] overcomes the problem of Rand Index. It returns a constant value for two random cluster sets.
- The **Adjusted Mutual Information** uses the entropy and mutual information formula and corrects the effect of a higher rating due to a larger number of document groups. The mutual information is a function which measures the equality of two assignments, ignoring permutations.

  The expected value for the mutual information can be calculated by the formula described by Vinh, Epps, and Bailey (2009) [27].

Study Evaluation

In this section the results of the user study will be presented. First the performance of each relatedness metric at its own is shown. With this knowledge, the best metrics will be selected for a second evaluation. The different metrics will be combined and used to train the weights using the Particle Swarm Optimization. Then the calculated weights will be used on the test set and the performance will be measured.

Evaluation of the Relatedness Metrics at its own

In the first evaluation we look at how the presented relatedness metrics perform standalone. We want to get more details about the quality of each of the relatedness metrics. This result will then be used to select a list of the best metrics for the second evaluation in which we will combine them together.

The evaluation is performed on the complete dataset. For the 8 participants all relatedness metrics were used to predict activity-centric document groups from the collection of documents. In the study we use the Agglomerative Hierarchical Clustering. The threshold θ is varied from 0 to 1 stepwise (0.01) and the best performance values are aggregated. Table 4 shows the aggregated mean (over all best individual values) and best values for each relatedness metric.

Metric	AMI	ARI	RI
C1: TF-IDF	**0.3706** (*0.5387*)	**0.3962**	**0.5606**
C2: LSA	0.3317 (*0.7119*)	0.3475	0.5300
C3: LDA	0.3361 (***0.7519***)	0.3350	0.5226
C4: Document Title	0.2202 (*0.3093*)	0.2004	0.4640
C5: Document Path	0.2891 (*0.5979*)	0.2604	0.4737
A1: Context Apps	0.2314 (*0.3699*)	0.1418	**0.4704**
A2: Time-Related (1)	0.1931 (*0.5029*)	0.1824	0.3562
A3: Time-Related (2)	***0.2950*** (**0.5986**)	**0.1917**	0.3671
R: Random	0.0396 (*0.1546*)	0.0289	0.4194

Table 4. Evaluation of each relatedness metric in average and best values in brackets.

The comparison of all evaluated relatedness metrics (see figure 4) shows a better performance of the content-related relatedness metrics. Only the path metric which uses the document path as the input delivers worse performance. The title metric also does not perform as well as the metrics which use the full content of the document.

In all cases content-related metrics outperform the activity-based approaches. The overall best performance delivered TF-IDF followed by LDA150. Although some paper [4, 14] show better performance of LDA or LSA, in our study the

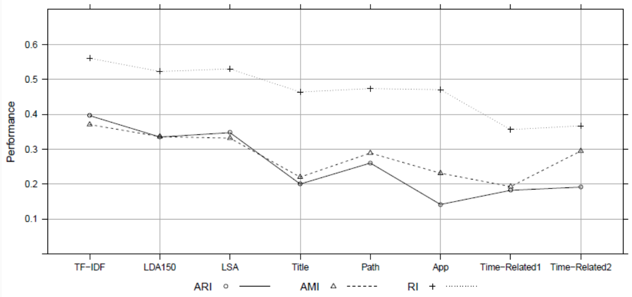


Figure 4. Average single relatedness metric performance comparison

TF-IDF approach was superior. The reason for this may be the smaller set of documents, compared to the used corpora in above mentioned papers. LSA and LDA are designed to work better with a larger number of documents. Overall, this result shows that the content of a document is a very strong indicator to determine activity-centric document groups.

Evaluation of the Combined Hybrid Feature Set

For evaluating the performance of the document clustering using a hybrid feature set it is needed to first learn the combining weightings for a part of the gathered data. Therefore, the collected dataset will be split into two sets (training and test) of equal size. Only the training set will be used to learn weightings using an optimization algorithm. The test set is then used to calculate the quality performance with the learned weightings, so the algorithm only sees the unknown data set. The dataset is separated by time, so that the training set consists of items which were collected in early days of the study and the test set contains items of the second time frame of the study.

From the result presented in the previous section a set of relatedness metrics are selected as candidates for the hybrid feature set approach:

1. **Mix of content and activity metrics**: TF-IDF, Path, App Context, Time-Related (1) and Time-Related (2)
2. **Only activity-based metrics**: App Context, Time-Related (1) and Time-Related (2)
3. **Only content-based metrics**: TF-IDF, Title, Path

The clustering threshold parameter will not be learned during the training, because this parameter works as a personal adjustment setting. The higher the number, the smaller the document groups will become. For the evaluation we will change the parameter stepwise between 0 and 1 by 0.01 steps and calculate the performance. During the training the parameter will be set to a fixed value (here: 0.7). It is not necessary to get the global maximum in the training phase because the threshold value will be varied when calculating the performance of the test set.

Training of Weighting Parameters

In the training phase the best weighting parameters for each study participant are calculated. Therefore, the training set is used as the input for the optimization algorithm which uses the combined relatedness model and adjusts the weighting parameters. The Adjusted Rand Index performance measurement is the function to be optimized. Particle Swarm Optimization is performed with 50 evolutions and 5 particles.

Testing Phase and Results

In the testing phase the real performance of the hybrid feature set document clustering approach is evaluated. The computed weightings which control the influence of each relatedness metric will be used on an unseen set of data. Then the performance of the predicted document groups will be measured.

1. Mix of content and activity metrics

In the first configuration a mixture of content- and activity-based relatedness metrics is used. This configuration is the desired feature set combining both worlds, thus it should deliver the best performance of all configurations.

Metric	*Mean*	*Best*	*Worst*
ARI	0.5287	0.7688	0.1908
AMI	0.5079	0.6942	0.2662
RI	0.6636	0.8711	0.4194

Table 5. Performance results of document clustering approach using a hybrid feature set combining Time-Related (1), Time-Related (2), App Context, TF-IDF, Path.

The document clustering approach with the hybrid feature set outperforms the performance of all single relatedness metrics. The combination of multiple metrics increases the quality of the predicted document groups. In average the Adjusted Rand Index is 0.5287 with a best performance of 0.7688. In comparison to the best single metric (TF-IDF) the combined model is about 0.13 basis points better.

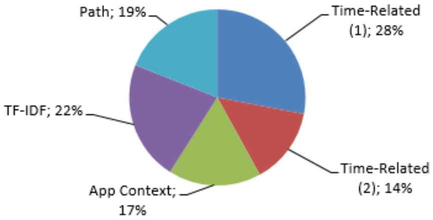


Figure 5. Average weighting parameter values. Configuration: Time-Related (1), Time-Related (2), App Context, TF-IDF, Path.

When looking at the learned weighting parameters the proportion (see figure 5) of the content-based relatedness metrics (41%) are slightly lower than the activity-based approaches (59%). The distribution over all metrics is nearly balanced. Only App Context is rated slightly lower than all other metrics. This result also shows that activity-based approaches help to better understand the structure and relationships between documents.

2. Only activity-based metrics

The second configuration only uses activity-based metrics as a comparison. This allows us to determine the influence of activity approaches in comparison to content-based approaches.

Metric	*Mean*	*Best*	*Worst*
ARI	0.3205	0.6361	0.0256
AMI	0.3522	0.6466	0.1116
RI	0.5582	0.7187	0.4194

Table 6. Performance results of hybrid feature set document clustering approach combining Time-Related (1), Time-Related (2), App Context.

In this configuration the results are getting worse in comparison to the mixed model. The best values from configuration 1 could not be reached without including the documents content.

The proportion of each metrics is almost balanced and can be seen in figure 6. Only the Time-Related (1) metric has a slightly larger influence on the result than Time-Related (2) and App Context.

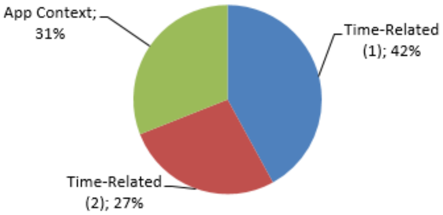


Figure 6. Average weighting parameter values. Configuration: Time-Related (1), Time-Related (2), App Context

3. Only content-based metrics

The last configuration only focuses on content-based metrics. In this configuration the TF-IDF, Path and Title metric is used to determine the relatedness between documents.

Metric	*Mean*	*Best*	*Worst*
ARI	0.3374	0.5263	0.0674
AMI	0.3532	0.5598	0.1533
RI	0.5582	0.6711	0.4194

Table 7. Performance results of the document clustering approach using a hybrid feature set combining TF-IDF, Path, Title.

In table 7 the performance of the combination of content-based relations is presented. This configuration also delivers a lower performance than the mixed model.

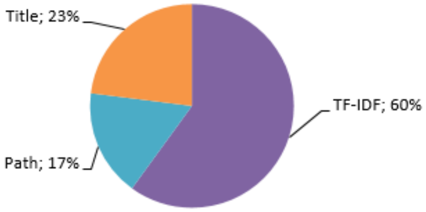


Figure 7. Average weighting parameter values. Configuration: TF-IDF, Path, Title.

The optimization algorithm delivers best results when TF-IDF similarity metric influences 60% of the result. Title and Path metrics equally divide the rest.

Evaluation Summary

The first evaluation has shown that the performance of content-based similarity metrics is better in comparison to activity-based approaches on a real world data set. The best performance was delivered by the TF-IDF method followed by LDA and LSA. The best activity-based metric was Path followed by Title, Time-Related (1), Time-Related (2) and App Context.

In the second part of the evaluation the combined hybrid feature set model was used to predict activity-centric document groups from a collection of documents. Three different configurations were evaluated, the first using activity- and content-based properties, the second using only activity-based approaches and the third using only content-based methods. The mix configuration outperformed all other configurations and has shown a significant performance increase when a mixture of different metrics is used to cluster documents. With an average performance of 0.5287 (ARI) and a best performance of 0.7688 (ARI) the combined model has shown a good performance. The main goal of this paper was to prove if a mixture model of different relatedness metrics can deliver a higher quality of the document groups than using a single approach. The evaluation has exactly shown that this is possible.

DISCUSSION AND CONCLUSION

In this paper we presented two main contributions. On the one hand we introduced an automatic clustering approach of activity related information based on interaction histories, using a mixed model of activity and content properties. Thus, we built on existing work in this domain. However, this paper is the first paper which combines the different methods. On the other hand we provided a long-term user study in an open world scenario, evaluating our and existing approaches on information work of 8 participants. The user study has shown that the hybrid feature set approach outperforms every single approach used in related work. Our approach also adapts to the users preference, and therefore delivers a user-centric view of documents.

Overlapping Clusters

This paper proposes a single cluster assignment of related information. However, information may consist of multiple topics and a single cluster assignment may not be sufficient. One solution is the creation of overlapping clusters by modifying the clustering algorithm and making changes to the cluster comparison metric. Overlapping clusters provide a better structure for multi-topic information but are also more complicated to understand for end-users. Another solution is the use of hierarchical structures which are more familiar to users that know the concept of file systems. Future research in this topic is required that also covers the graphical representation of clusters.

Interactive System

We provided an automatic hybrid clustering approach which has shown promising results in the user study. This opens up the development for an interactive system allowing users to organize their information work that provides far better clustering results than other systems. We could imagine that the weighing parameters can be determined by letting users cluster some information objects manually and/or provide default weightings and let the user correct wrong assignments. Using this knowledge automatic information clustering of the rest of the collection can be achieved. Future research which addresses the user interface (e.g. displaying clusters) and interaction (e.g. manual correction) with the system, may also help to further improve the results.

ACKNOWLEDGEMENTS

This project (HA project no. 479/15-21) is funded in the framework of Hessen ModellProjekte, financed with funds of LOEWE Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz, Förderlinie 3: KMU-Verbundvorhaben (State Offensive for the Development of Scientific and Economic Excellence).

REFERENCES

1. Bailey, B. P., and Konstan, J. A. On the need for attention-aware systems : Measuring effects of interruption on task performance , error rate , and affective state. *Computers in Human Behavior 22* (2006), 685–708.

2. Bellotti, V. Managing Activities with TV-ACTA : TaskVista and Activity- Centered Task Assistant. In *Personal Information Management Workshop*, ACM Press (2006), 8–11.

3. Berry, M., Dumais, S., and O'Brien, G. Using linear algebra for intelligent information retrieval. *SIAM review*, December (1995).

4. Blei, D., Ng, A., and Jordan, M. Latent dirichlet allocation. *the Journal of machine Learning research 3* (2003), 993–1022.

5. Brdiczka, O., Su, N., and Begole, J. Temporal task footprinting: identifying routine tasks by their temporal patterns. In *Proceedings of the 15th international conference on Intelligent user interfaces*, ACM Press (2010).

6. Fowlkes, E. B., and Mallows, C. L. A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association 78* (1983), 553–569.

7. González, V., and Mark, G. Constant, constant, multi-tasking craziness: managing multiple working spheres. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, vol. 6, ACM Press (2004), 113–120.

8. Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning*. 2009.

9. Hubert, L., and Arabie, P. Comparing partitions. *Journal of Classification 2*, 1 (1985), 193–218.

10. Iqbal, S. T., and Horvitz, E. Disruption and recovery of computing tasks: Field study, analysis, and directions. In *Proc. CHI 2007*, ACM (2007), 677–686.

11. Jeuris, S., Houben, S., and Bardram, J. Laevo: A temporal desktop interface for integrated knowledge work. In *Proc. UIST 2014*, ACM (2014), 679–688.

12. Kaptelinin, V. UMEA: translating interaction histories into project contexts. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, no. 5, ACM Press (2003), 353–360.

13. Kennedy, J. Particle swarm optimization. *Encyclopedia of Machine Learning 4* (2010), 1942–1948.

14. Landauer, T. K., Foltz, P. W., and Laham, D. An introduction to latent semantic analysis. *Discourse Processes 25*, 2-3 (Jan. 1998), 259–284.

15. Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*, vol. 1. 2008.

16. Oliver, N., Smith, G., Thakkar, C., and Surendran, A. C. Swish: Semantic analysis of window titles and switching history. In *Proc. IUI 2006*, ACM (2006), 194–201.

17. Rajaraman, A., and Ullman, J. D. Data mining. In *Mining of Massive Datasets*. Cambridge University Press, 2011, 1–17. Cambridge Books Online.

18. Rand, W. M. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association 66* (1971), 846–850.

19. Rattenbury, T. *An activity based approach to context-aware computing*. PhD thesis, 2008.

20. Reinhardt, W., Schmidt, B., Sloep, P., and Drachsler, H. Knowledge Worker Roles and Actions - Results of Two Empirical Studies. *Knowledge and Process Management 18*, 3 (2011), 150–174.

21. Schmidt, B. *Information Work Support Based on Activity Data*. PhD thesis, TU Darmstadt, Mai 2013.

22. Schmidt, B., Kastl, J., Stoitsev, T., and Mühlhäuser, M. Hierarchical Task Instance Mining in Interaction Histories. In *Proceedings of the 29th annual international conference on Design of communication (SIGDOC)*, ACM Press (2011).

23. Schweizer, I., and Schmidt, B. Kraken.me: Multi-device user tracking suite. In *Proc. UbiComp 2014*, ACM (2014), 853–862.

24. Shen, J., Li, L., Dietterich, T. G., and Herlocker, J. L. A hybrid learning system for recognizing user tasks from desktop activities and email messages. In *Proc. IUI 2006*, ACM (2006), 86–92.

25. SINTEF. Big data, for better or worse: 90% of world's data generated over last two years. *ScienceDaily* (May 2013).

26. Smith, G., Bausdich, P., Robertson, G., Czerwinski, M., Meyers, B., Robbins, D., and Andrews, D. Groupbar: The taskbar evolved. In *Proc. OZCHI 2003* (January 2003).

27. Vinh, N. X., Epps, J., and Bailey, J. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proc. ICML 2009*, ACM (2009), 1073–1080.

28. Wagner, S., and Wagner, D. *Comparing clusterings: an overview*. No. 001907. 2007.

29. Wilson, E. O. *Sociobiology: The new synthesis*. Harvard University Press, 2000.

30. Witten, I. H., Frank, E., and Hall, M. A. *Data Mining: Practical Machine Learning Tools and Techniques (Google eBook)*. 2011.