Online Activity Graph for Document Importance and Association

Charlie Abela
Digital Enterprise Research Institute,
National University of Ireland,
Galway, Ireland
charlie.abela@deri.org

Chris Staff
Department of Intelligent Computer Systems,
University of Malta,
Malta
chris.staff@um.edu.mt

Siegfried Handschuh
Digital Enterprise Research Institute,
National University of Ireland,
Galway, Ireland
siegfried.handschuh@deri.org

ABSTRACT

The way in which a user interacts with her desktop while performing some task generates an information trail that can be used to identify the task context and the user's interests. This new information can in turn be fed back into the system to increase the level of support available to the user for both current and future tasks. In this paper we present research which analyses user-activity log files to explore how a user's activities evolve with time. Resources fall in and out of a task based on the user's mental model for tackling that task. We assign time-varying, *importance* and *association* values to each resource, based on the dwell-time and the resource-switching patterns exhibited by the user while browsing. Furthermore, we propose a new dynamic graph algorithm called *OnlineActivityGraph* which leverages on these values to generate document clusters and short-term user models. We further present a discussion about the encouraging results obtained from our preliminary experiments.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.2.8 [**Problem Solving, Control Methods, and Search**]: Graph and tree search strategies

General Terms

Theory

Keywords

Dynamic Graphs, User Model, Time decay

1. INTRODUCTION

The way in which a user interacts with her desktop while performing some task generates an information trail that can be used to identify the task context and the user's interests. This new information can in turn be fed back into the system to increase the level of support available to the user for both current and future tasks.

The identification of the task-context is challenging in a number of ways, most importantly because it tends to be elusive, since the user might change her task (or subtask) depending on the goals she wants to reach and the ways through which she wants to achieve them. We define the notion of a task with regards to a Knowledge Worker (KW) as follows:

"A task is considered to be an abstraction over an evolving collection of resources, which the KW mentally relates, based on their perceived importance, to successfully address some goal."

When resources are accessed to address some task-related need, the temporal dimension plays an important role [16, 2]. The information that the user sifts through provides for a moving and evolving information context that affects also the user's short-term interests.

The relation between time spent going over a document and the relative user's interests in that document have been explored by [13, 6] whereby it was found that there was a strong tendency to spend time reading through an article because it was found to be interesting. Nevertheless, other contextual information and the user's interests need to be considered so that it becomes useful [11, 12]. In [8] resources have been identified as pertaining to a task based on three criteria, (i) the temporal access of the resources, (ii) the temporal vicinity of the subsequent activities, as well as (iii) the similarity between the content embedded within these resources.

In this paper we propose the inclusion of a fourth criteria, which is the dwell time, that is the amount of time that the user spends interacting with each resource, during the various accesses. This criteria is a good indicator of the importance that a resource has for the user at some particular point in time, thus making it core in the user's task.

We analyse user-activity log files to explore how users' activities evolve with time. We assign time-varying, *importance* and *association* values to each resource, based on the dwell-time and the resource-switching patterns exhibited by the user while browsing. While the *importance* value is used to quantify the importance of an accessed document during its access life-cycle, the *association* value, is used to quantify the strength of the relation between one document and all the other documents accessed during the same session.

To capture the dynamicity of accessed resources we de-

I-SEMANTICS 2011, 7th Int. Conf. on Semantic Systems, Sept. 7-9, 2011, Graz, Austria

veloped an online graph algorithm which we call *OnlineActivityGraph*, which is inspired by the *association graph* described in [4] and the work on graphs reported by [1] and [10]. The algorithm takes as input a stream of desktop activities and computes both the evolving nature of the document's importance as well as its relation to other switched-to documents. The latter is used to identify those documents which are most-likely to pertain to the same task.

In our experiments we apply the algorithm on phrases extracted from the accessed documents and display them to the user depending on the level of importance gained, or lost, by their parent documents, as well as the relatedness that documents have instilled between each other.

The remainder of the paper is organised as follows. In Sections 2. and 3. we discuss the *importance* and the *association* heuristic functions. In Section 4. we present the *OnlineActivityGraph* algorithm and explain how we integrated it with an keyword-extraction component to generate short-term user models. We present our initial experiments and results in Section 5. and conclude this study in Section 6.

2. IMPORTANCE HEURISTIC

Whenever the KW performs some research task a number of documents are accessed and examined with varying degrees of attention. This attention by the user tends to change over time, [7, 9, 14].

We assume that documents gain importance when they are accessed and time is spent by the KW reading through, and that, once the KW switches to some other document, a decay process is triggered which erodes the importance value acquired up till that point in time. Furthermore, this process of acquiring importance and loosing it, is ongoing and reflects the KW's document usage behaviour. We further assume that once a KW decides to access a document for the first time, it is as if the document has already acquired some importance.

To be able to represent this process of importance gain and decay, we draw an analogy with the charging and discharging operations within a capacitor. For more background, we direct the reader to [15] or any other similar textbook.

We adapted the capacitor's features to those of documents being accessed, abandoned and possibly re-visited by the KW. To make our approach more realistic we consider the dwell time on a document together with the switching behaviour, thus factoring in more of the user's activity context.

Based on this analogy we define the following mapping:

i. We consider the capacitor to represent the accessed document.

ii. We map the voltage V to the user's attention. When a KW accesses a document she is devoting attention to it, similar to how voltage is applied to a capacitor.

iii. We consider the charge Q to represent the importance I that the document acquires due to the time that the user spends reading through.

iv. We consider the *time constant*, $\tau = \frac{1}{RC}$ to be an averaged dwell-time constant. Based on findings by [12], the average dwell time on a page was found to be around 70 seconds in 80% of the 205,873 pages that they considered in their experiments.

v. The time t during document *charging*, is taken to be the dwell time $t_{access} - t_{switch}$, where t_{access} is the timestamp when the document is accessed, while t_{switch} is the time when user switches to some other document. During *discharging*, t is taken to be the time $t_{curraccess} - t_{switch}$, where $t_{curraccess}$ is the time when the document is re-visited.

vi. The maximum importance value that a document can attain is 1, while the minimum is 0. In theory though, this minimum value will never be completely reached, unless the user closes this document, which is taken to mean that the user is not interested in that document, at that point in time.

vii. The default importance is not set to 0, and arbitrarily is initially set to 0.1. However we plan to investigate whether this value is realistic or not, whether it should be lower or higher.

We then use the following equations to compute the importance gain and decay for a document A:

$$I(A) = \begin{cases} \left\{(I_{min}(A) - 1) * e^{-t/\tau}\right\} + 1 & dwell > 0 \\ I_{max}(A) * e^{-t/\tau} & decay > 0 \end{cases} \quad (1)$$

where I_{min} is the residual importance or current decay value and I_{max} is the importance acquired by the document when the user dwelled on it last.

The triggering of the *charging* and *discharging* processes are totally controlled by the user because the decision to open a new document, close an existing one and/or switch to another document, lies with him. We envision however that documents which have been dwelled upon for some time will maintain importance over time and therefore whenever the user switches back to these documents, the importance value will be computed, taking into consideration, the decay value reached at that time.

Based on this approach for computing the importance values of accessed documents, we are able to analyse how this importance dynamically evolves with time.

3. ASSOCIATION HEURISTIC

When the KW switches between one document and another, her behaviour is considered to be an indication that they are somewhat associated. Research discussed in [5, 8] has shown that this relation between frequently and closely accessed documents exists. [3] refers to this relation as a *fuzzy association* which somehow bonds documents together to various degrees of strength.

In this research we built on the aforementioned work on document association and combine it with the importance value explained earlier to be able to generate evolving clusters of documents which depend solely on the user's behaviour. The *Association* metric is based on that discussed in [3] and is defined for any two documents A and B, as a combination of two weighted, switching ratios $WtRatio(A, B)$ and $WtRatio(B, A)$. We first present the equation for the weighted ratio:

$$WtRatio(A, B) = \frac{s(A, B)}{\sum_{X \neq A} s(A, X)} \times \frac{I(A)}{I(A) + I(B)} \quad (2)$$

where $s(A, B)$ is the number of switches that the user has performed between A and B and $s(A, X)$ is the number of switches that the user has switched between A and any other document. $I(A)$ and $I(B)$ are the importance of documents A and B at some point in time.

The $Association$ is then computed as:

$$Assoc(A, B) = WtRatio(A, B) + WtRatio(B, A) \quad (3)$$

In this manner, we allow the importance of a document to be dependent on the dwell time, and the association between documents, to be dependent on both the number of switches as well as the importance value, gained or decayed, by the documents at that point in time.

4. ONLINEACTIVITYGRAPH

Since the KW's document-access cycle provides streams of real-time data we designed our *OnlineActivityGraph* algorithm as an online algorithm which maintains and allows operations on the user's *activity graph*.

We consider this graph to be sparse with some nodes getting into and out of clusters as the graph evolves. The weight of each node depends on the importance heuristic mentioned above. The switches between documents, performed by the user, are represented by edges and the computed edge weights depend on the importance and association heuristic functions. The user's behaviour can essentially be represented by a weighted, directed multigraph, however we transform this into a weighted, undirected simple graph and find paths along the *activity graph* that maximize the association value.

Currently we only considered logging data when the user interacts with Firefox, and for this reason we have adapted the *Dragontalk* [1]logger.

Each user's *action* is logged in with a timestamp, a URL to uniquely identify it, and other information about the type of action, such as *ChangeTab*, *NavigateToURL* or *FollowLink*.

We define a *maximal association path*, to be a path with edge weights that exceed an association threshold value, T. This is motivated by what [10] refers to as an *influence region*, which represents the association strength between two nodes and is defined as follows:

Definition 1. Given a graph G and a node s, the influence region $region(s)$ of s is

$$d \in region(s) \Leftrightarrow influence(s, d) > T \quad (4)$$

4.1 Node and Edge insertions

Insertion of a node is actually an appending since nodes arrive in sequence. We create a new node based on the information being logged for that action and connect it to the last[2] added node via a new edge. The importance of the last node is computed and that of the new node set to a default value. This operation causes a perturbation across the whole graph since the newly computed importance value of the last node effects the weights of the connected edges.

Whenever a node is revisted, an edge is created. The importance of the last accessed node is computed together with an update operation on the association weight between the last node and the re-visited one.

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

[2]last node: node that has been appended in the previous cycle

4.2 Updating weights

When the user accesses the first document, a node is created with a default importance value. For each new node being added, we use equations (1) to compute the last node's importance value, and for all the other nodes, except the current node (unless it is a revisted node), we compute the decay value and assign it to the importance value of that node. We set default values for the time constant τ used in both the importance and decay equations.

Since the association weights depend on the importance values and the introduction of new edges, these have to be recomputed for each edge in the graph. If the node is being revisited, a new edge is created and assigned the computed assocation value.

4.3 Finding the maximal Association paths

To find the maximal association paths within the undirected weighted graph, we use a breadth-first search. The algorithm starts from the first node in the graph and for each one of its neighbours it checks the association value on the connecting edge. If that value exceeds a pre-defined threshold, which we initially set to 0.5, then the edge is added to an *edge bucket* associated with that path. The process for edge searching then continues along the other neighbours of the current node.

If an association value is found to be less then the threshold, it is temporarally discarded, however the connecting node is fed back into the algorithm, so that if the association values connecting it to its neighbours are higher then the threshold, a new *edge bucket* is created for this new path.

Once all edges have been checked, the result is a set of *edge buckets* with varying *maximal association path* values.

5. EXPERIMENTS

We experimented with the above algorithms and analysed the results for varying threshold values. The experiments were performed on a very small scale, (which included the authors of this paper) within the same institutions. Various activities performed over different time windows were logged and analysed.

Since changes to τ effect the slope of the increase or decrease of the importance and decay relations, we performed various experiments to check the effect of these values on the whole process, as well as whether a balance between the time constants used for the importance and the decay functions could be found, that generated acceptable results.

Preliminary results showed that the default value of 70s for the importance and decay time constants was not ideal for situations where the user spent consistently long dwell time on each of the accessed documents. It performed best when the dwell time was short. Based on a set of logged sessions that we analysed, we came up with an averaged dwell time of 187s which seemed to work better.

We experimented with different combinations of time constant values. We assumed that if we increased the time constant in the importance equation, this will at the end translate into stronger associations. The rationale being that since the importance value would have increased at a lower rate, it would need more dwell time for it to approach the maximum. Furthermore we used a greater time constant (arbitrarily we chose to use twice the importance time constant) in the decay equation. In this case we assumed that a document that has gained importance, would loose it at a

lower rate, thus allowing also residual importance to diminish at a lower rate. We were however aware of the *"rich-get-richer"* scenario pointed out by [9] which in our case would favour long dwell time and few number of switches, over short dwell time and larger number of switches, and tried to find a working balance.

We then performed preliminary analyses on how effective was the clustering process on the documents, based on the *maximal association paths*. With the default *association* value set at 0.5 we noticed that some documents kept entering and falling out of a cluster fairly quickly. When we increased the value to 0.7, many documents never managed to cluster with others. There are possibly various reasons behind this, such as the browsing mood of the user, which may result in shorter or longer dwell times.

When we decreased the threshold value below 0.5 we started to notice the builing up of some stronger clusters, made up of documents that clustered together and kept to this cluster over time.

We further used our approach to generate clusters of phrases pertaining to the clustered documents. The phrases were extracted from each document with a key phrase extractor called Xtrak4Me [3].

Based on the *maximal association paths*, those documents which pertained to the same cluster were displayed to the user with the same colour. We also combined the confidence score assigned to each phrase by the extractor, with the document importance value generated by the importance or decay functions on that document. This global score is defined as $FI\lambda/F_{max}$. Where F is the score assigned by the extractor, λ is a scaling factor and F_{max} is the highest confidence score assigned to a phrase within the whole phrase set. We used this global score to compute the size of that phrase when it is displayed to the user in a tag cloud.

Though the results are encouraging, at this preliminary stage in our research we cannot make any strong claims about this approach and a more elaborate evaluation based on a combination of a naturalistic approach and a longitudinal evaluation is planned.

6. CONCLUSIONS

In this paper we presented research which analysed user-activity log files to explore how user's activities evolve with time. We assign time-varying, *importance* and *association* values to each resource, based on the dwell-time and the resource-switching patterns exhibited by the user while browsing. Furthermore, we elaborated on our dynamic graph algorithm called *OnlineActivityGraph* which leverages on these values to create clusters of documents based on what we call *maximal association paths*. We also presented some discussion about the encouraging results obtained from our preliminary experiments, in particular how our approach can be used to generate short-term user models.

7. REFERENCES

[1] C. C. Aggarwal and P. S. Yu. Online analysis of community evolution in data streams. In *Proceedings of SIAM International Data Mining Conference*, 2005.

[2] O. Alonso, M. Gertz, and R. Baeza-Yates. On the value of temporal information in information retrieval. *SIGIR Forum*, 41(2):35–41, December 2007.

[3] http://smile.deri.ie/projects/keyphrase-extraction

[3] M. Bernstein, J. Shrager, and T. Winograd. Taskpose: Exploring fluid boundaries in an associative window visualization. In S. Cousins and M. Beaudouin-Lafon, editors, *UIST '08: Proceedings of the 21st annual ACM symposium on User Interface Software and Technology*, pages 231–234, New York, NY, USA, 2008. ACM Press.

[4] J. Chen, H. Guo, W. Wu, and W. Wang. imecho: an associative memory based desktop search system. In D. W.-L. Cheung, I.-Y. Song, W. W. Chu, X. Hu, and J. J. Lin, editors, *CIKM*, pages 731–740. ACM, 2009.

[5] P. Chirita, S. Costache, J. Gaugaz, and W. Nejdl. Desktop context detection using implicit feedback. In *SIGIR 2006 Workshop on Personal Information Management, Seattle WA*, USA, 2006.

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

[7] G. Cormode, V. Shkapenyuk, D. Srivastava, and B. Xu. Forward decay: A practical time decay model for streaming systems. In *International Conference on Data Engineering*, pages 138–149, 2009.

[8] S. Costache, J. Gaugaz, E. Ioannou, C. Niederee, and W. Nejdl. Detecting contexts on the desktop using bayesian networks. In *DESKTOP Search Workshop co-located with SIGIR*, 2010.

[9] E. M. Daly. Harnessing wisdom of the crowds dynamics for time-dependent reputation and ranking. In *Advances in Social Network Analysis and Mining*, pages 267–272, 2009.

[10] S. Günnemann and T. Seidl. Subgraph mining on directed and weighted graphs. In *Proc. 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2010), 21-24 June, 2010 - Hyderabad, India. Lecture Notes in Artificial Intelligence (LNAI)*, pages 133–146. Springer, 2010.

[11] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *Sigir Forum*, 37:18–28, 2003.

[12] C. Liu, R. White, and S. Dumais. Understanding web browsing behaviors through weibull analysis of dwell time. In *SIGIR*, pages 379–386, 2010.

[13] M. Masahiro and S. Yoichi. Information filtering based on user behavior analysis and best match text retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pages 272–281, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[14] G. Papadakis, C. Niederée, and W. Nejdl. Decay-based ranking for social application content. In *WEBIST (1)*, pages 276–281, 2010.

[15] A. H. Robbins and W. C. Miller. *Circuit Analysis with Devices: Theory and Practice*. SDelmar Cengage Learning, 4th edition, 2006.

[16] A. Tsymbal. The problem of concept drift: Definitions and related work. Technical report, Department of Computer Science, Trinity College, Dublin, Ireland, 2004.