Use of Implicit Graph for Recommending Relevant Videos: A Simulated Evaluation

David Vallet[1,2], Frank Hopfgartner[2], Joemon Jose[2]

[1] Universidad Autónoma de Madrid, Madrid, Spain
[2] University of Glasgow, Glasgow, UK
david.vallet@uam.es, {hopfgarf, jj}@dcs.gla.ac.uk

Abstract. In this paper, we propose a model for exploiting community based usage information for video retrieval. Implicit usage information from a pool of past users could be a valuable source to address the difficulties caused due to the semantic gap problem. We propose a graph-based implicit feedback model in which all the usage information can be represented. A number of recommendation algorithms were suggested and experimented. A simulated user evaluation is conducted on the TREC VID collection and the results are presented. Analyzing the results we found some common characteristics on the best performing algorithms, which could indicate the best way of exploiting this type of usage information.

1. Introduction

In recent years, the rapid development of tools and systems to create and store private video enabled people to build their very own video collections. Besides, the easy to use Web applications such as YouTube and Google Video, accompanied by the hype produced around social services, motivated many to share video, leading to a rather uncoordinated publishing of video data. Despite the ease with which data can be created and published the tools that exist to organise and retrieve are insufficient in all terms (effectiveness, efficiency and usefulness). Hence, there is a growing need to develop new retrieval methods that support the users in searching and finding videos they are interested in. However, video retrieval is affected by the semantic gap [5] problem, which is the lack of association between the data representation based on the low-level features and the high-level concepts users associate with video

One promising approach taken from the textual domain is the integration of relevance feedback to improve retrieval results. However, as in text retrieval, giving explicit relevance feedback is a cognitively demanding task and can affect the search process. A solution is to take implicit relevance feedback into account. However, which of these feedback possibilities in video retrieval are positive indicators about the relevance of a result has rarely been analysed.

In this paper, we are interested in using implicit relevance feedback from previous users of a digital video library to form a collaborative model of user behaviour, helping users find results which match their information need. We believe that the combined implicit relevance feedback of a larger group can be used to provide users with positive recommendations. Although part of the data used in our evaluation comes from a user study, our main interest was evaluating a relatively high number of recommendation algorithms, which made the possibility to extend the user study to all the algorithms highly costly. Our evaluation required the possibility of being repeatable, allowing the study of different variables within a reasonable amount of time. Therefore, we introduce an approach of analysing implicit relevance feedback mechanisms based on a simulation-based evaluation.

The remainder of the paper is structured as follows. A brief summary of related work on implicit feedback applied to Multimedia Information Retrieval (MIR) and simulation-based evaluation is presented in section 2. Section 3 introduces a graph-based implicit pool representation, along with different recommendation strategies and subsequently in section 4, we describe the simulation based evaluation methodology. Section 5 will discuss the simulation results and will conclude in section 6 with some final thoughts.

2. Background

2.1 Implicit Feedback in Multimedia Information Retrieval

Deviating from the method of explicitly asking the user to rate the relevance of retrieval results, the use of implicit feedback techniques helps learning user interests unobtrusively. The main advantage is that users are relieved from providing feedback. While the techniques have been studied intensively in the textual domain [7], rarely anything is known in the multimedia domain. Hopfgartner and Jose [4] identified various implicit indicators of relevance in video retrieval when comparing the interfaces of state-of-the-art video retrieval tools. They introduced a simulation framework to analyse the effect of implicit relevance feedback in video retrieval, concluding that the usage of implicit indicators can influence retrieval performances. However, which of these implicit measures are useful to infer relevance has rarely been analysed in detail. Kelly and Belkin [6] criticise the use of display time as relevance indicator, as they assume that information-seeking behaviour is not influenced by contextual factors such as topic, task and collection. Therefore, they performed a study to investigate the relationship between the information-seeking task and the display time. Their results cast doubt on the straightforward interpretation of dwell time as an indicator of interest or relevance.

Usage information from a community of previous users can aid multimedia information retrieval. Usage information in the form of click-through data has been exploited [1]. When a user enters a query, the system can exploit the behaviour of previous users that issued a similar query. In this work, we are interested in approaches regarding MIR and graph-based representations of usage information. White et al. [10] introduced the concept of query and search session trails, where the interaction between the user and the retrieval system is seen as a path that leads from the first query to the last document of the query session or the search session (i.e. multiple queries). They argue that the last document of these trails is more likely to be relevant for the user. In our approach, we adopt this introduced concept of search trails. Furthermore, we are interested in representing and exploiting the whole interaction process. In video retrieval, the interaction sequence is a reasonable way to track the user's information need. Craswell and Szummer [1] represent the clickthrough data of an image retrieval system as a graph, where queries and documents are the nodes and links are the clickthrough data. We adopt also a graph-based approach, as it facilitates the representation of interaction sequences. While the authors limit the graph to clickthrough data, we propose to integrate other sources of implicit relevancy into the representation, as following [4].

2.2 Simulation Frameworks

In the de facto standard evaluation methodology known as Cranfield evaluation, users interact with a system searching for given search topics in a limited dataset. An analysis of recorded transaction log files and the retrieval results is then used to evaluate the research hypothesis. An alternative way of evaluating such user feedback is the use of simulated interactions. In such an approach, a set of possible steps are assumed when a user is performing a given task with the evaluated system [3,4,11].

Finin [2] introduced one of the first user simulation modelling approaches. This "*General User Modelling System*" (GUMS) allowed software developers to test their systems in feeding them with simple stereotype user behaviour. White et al. [11] proposed a simulation-based approach to evaluate the performance of implicit indicators in textual retrieval. They simulated user actions as viewing relevant documents, which were expected to improve the retrieval effectiveness. In the simulation-based evaluation methodology, actions that a real user may take are assumed and used to influence further retrieval results. Hopfgartner et al. [3] introduce a simulation framework to evaluate adaptive multimedia retrieval systems. In order to develop a retrieval method, they employed a simulated evaluation methodology which simulated users giving implicit relevance feedback. Hopfgartner and Jose [4] extended this simulation framework and simulated users interacting with state-of-the-art video retrieval systems. They argue that a simulation can be seen as a pre-implementation method which will give further opportunity to develop appropriate systems and subsequent user-centred evaluations. In this work, we will use the concept of simulated actions, although we will simulate user actions based on the past history and behaviour of users, trying to mimic the interaction of past users with an interactive video retrieval system.

3. Implicit Graph Recommendation Approaches

In this section, we present a set of recommendation algorithms on the graph representation. The approaches have been adapted to exploit the implicit graph, introduced in this section. The implicit graph models the historical data of interaction across all users and sessions. The main two characteristics of this graph model are 1) the representation of all the user interactions with the system, including the interaction sequence and 2) a scalable aggregation of the implicit information into a single representation. The implicit graph facilitates the analysis and exploitation of past implicit information, resulting in a model that is easy to build on top of different recommendation algorithms.

3.1 Implicit Graph Representation

The representation of the implicit graph can be seen in two different layers: the first one, a Labelled Directed Multigraph (LDM), gives a full detailed representation of the implicit information, and the second, a Weighted Directed Graph (WDG), is inferred from the previous, simplifying the interpretation of the LDM. It is on top of the WDG where the different recommendation rankings will be defined. Note that the WDG is not dependent on the LDM, and can be computed directly.

A user session s can be represented as a set of queries Q_s, which were input by the user, and the set of multimedia documents D_s the user accessed during the session. Queries and documents are therefore the nodes $N_s = \{Q_s \cup D_s\}$ of our graph representation $G_s = (N_s, A_s)$, in which the arcs are the set of actions $A_s(G) = \{n_i, n_j, a, u, t\}$ indicating that, at a time t, the user u performed an action of type a that lead the user from node n_i to node n_j, and $n_i, n_j \in N_s$. Note that n_j is the object of the action and that actions can be reflexive, for instance when a user clicked to view a video and then navigate through it. Actions types depend on the kind of actions recorded by the video retrieval system, like clicking, playing for an interval, navigating through the video or browsing to the next keyframe etc... Links can contain extra associated metadata, as type specific attributes, e.g. length of play in a play type action. The graph is multilinked, as different actions can have same source and destination nodes. All the session-based graphs are aggregated into a single graph $G = G(N, A)$, $N = \bigcup_s N_s$, $A = \bigcup_s A_s$ which can be seen as an overall pool of implicit information.

In order to enable the exploitation of the previous representation by the recommendation algorithms, we simplify the LDM by using no-labelled weighted links and collapsing all links interconnecting two nodes into one. This process is done in two steps: the first step computes a weighted graph $G_s = (N_s, W_s)$ that represents the user interactions during a single session. Links $W_s = \{n_i, n_{j,}\mathrm{w_s}\}$ indicate that at least one action lead the user from node n_i to n_j. The weight value $\mathrm{w_s}$ represents the final relevance value calculated for node $n_{j,}$, its *local relevance* $lr(n_{j,})$. This value is obtained from the accumulation of implicit relevance evidences, given by the function $lr(n) = 1 - \frac{1}{x(n)}$, where $x(n)$ is the total of added weights associated to each type of action in which node *n* is object of. This subset of actions is defined as $A_s(G_s, n) = \{n_i, n_j, a, u, t | n_j = n\}, n \in N_s$. The x(n) weights are natural positive values returned by a function $f(a): A \rightarrow \mathbb{N}$, which returns higher values as the action are understood to give more evidence of implicit relevance. For instance, a user navigating through a video is a somehow good indication of implicit relevance. On the other hand, playing duration has proved to be a not as good indication [6], thus having a lower weight. This analysis on the impact of implicit feedback importance weights is based on a previous work by Hopfgartner et al. [4]. The accumulation of implicit relevance weights can thus be calculated as $x(n) = \sum_{a \in A_s(G_s,n)} f(a)$. Figure 1 depicts an example of LDM and its correspondent WDG for a given session.

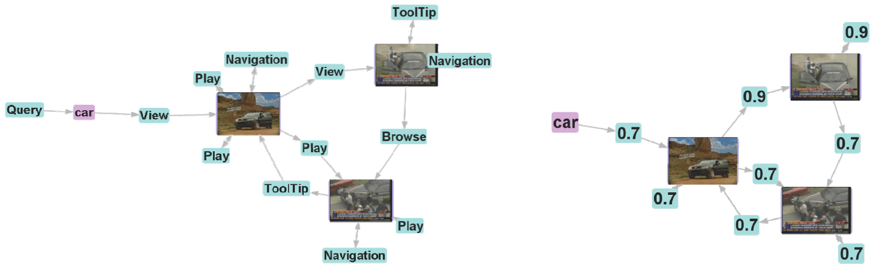

Fig. 1. Correspondence between the LDM (left) and WDG (right) representations

In the second step all the session-based WDGs are aggregated into a single overall graph $G = (N, W)$, which represents the implicit relevance pool, as collects all the implicit relevance evidence of all users across all sessions. The nodes of the implicit pool are all the nodes involved in any past interactions, $N = \bigcup_s N_s$, whereas the weighted links are a simple aggregation of the session-based values $W = \{n_i, n_j, w\}$, $w = \sum_s w_s$. These links represent the overall implicit relevance that users, which actions lead from node n_i to n_j, gave to node n_j. Figure 2 shows an example of implicit relevance pool.

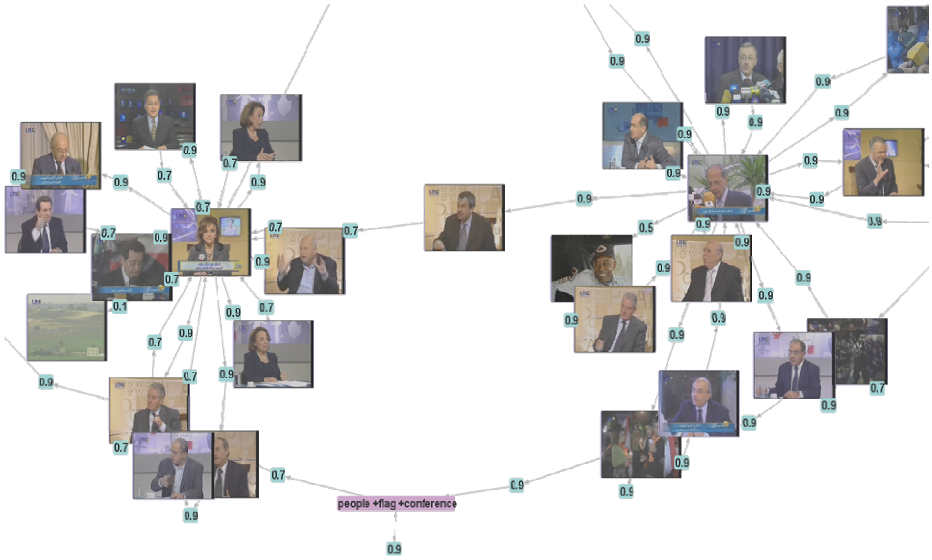

Fig. 2. Typical graph structure, where some relevant nodes receive a large number of links

3.2 WDG based recommendation algorithms

As the user interacts with the system, a session-based WDG is constructed. The current user's session is thus represented by $G_{s'} = (N_{s'}, W_{s'})$. This graph is the starting point of the recommendation algorithms presented next, and can be seen as a form of actual context for the user.

Neighbourhood. As a way of obtaining related nodes, we define the node neighbourhood of a given node n as:

$$NH(n) = \{n_1, \ldots, n_M | distance(n, n_m) < D_{MAX}, n_m \in N\}$$

which are the nodes that are within a distance D_{MAX} of n , without taking into consideration the link directionality. These nodes are somehow related to n by the actions of the users, either because the users interacted with n after interacting with the neighbour nodes, or because there are the nodes the user interacted with after interacting with n.

Using the properties derived from the implicit graph, we can calculate the overall relevance value for a given node, this value indicates the aggregation of implicit relevance that users gave historically to n, when was involved in the users' interactions. Given all the incident weighted links of n, defined by the subset $W_s(G_s, n) = \{n_i, n_j, w | n_j = n\}, n \in N_s$, the overall relevance value for n is calculated as follows:

$$or(n) = \sum_{w \in W_s(G_s, n)} w$$

Given the current session of a user and the implicit relevance pool we can then define the node recommendation value as:

$$nr(n, N_{s'}) = \sum_{n_i \in N_{s'}} lr'(n_i) \cdot or(n) | n \in NH(n_i)$$

where $lr'(n_i)$ is the local relevance value for the current session of the user, using the subset of actions $A_s(G_{s'}, n)$. We can then define two different recommendation values: the query neighbourhood $nh_q(n, N_{s'}) = nr(n, Q_{s'}) | Q_{s'} \in N_{s'}$, which recommends nodes related to the actual queries of the user and, similarly, the document neighbourhood $nh_d(n, N_{s'}) = nr(n, D_{s'}) | D_{s'} \in N_{s'}$, which recommends instead nodes related to the documents involved in the user's interactions.

Interaction Sequence. This recommendation approach tries to take into consideration the interaction process of the user, with the scope of recommending those nodes that are following this sequence of interactions. For instance, if a user has opened a video of news highlights, the recommendation could contain the more in-depth stories that previous users found interesting to view next. It is defined as follows:

$$is(n, N_{s'}) = \sum_{n_i \in N_{s'}} \left((lr'(n_i) \cdot \xi^{l-1} \cdot w) \middle| \begin{array}{c} \exists\, p = n_i \rightsquigarrow n_j \rightarrow n \\ w \in \{n_j, n, w\} \\ l = length(p) \\ l < L_{MAX} \end{array} \right)$$

where p is the path between any node n_i and node n, taking into consideration the link directionality. l is the length of the path (counted as the number of links), having a distance is lower than a maximum length L_{MAX}. Finally, ξ is a length reduction factor, set to 0.8 in our experiments.

Query Destination. This algorithm is adapted from the work of White et al. [10] on query and search trails. White suggests that the last documents that a user visits within a search or query session has a high relevancy. We choose the query destination measure, which they proved that was best for explorative tasks (used in the evaluation process). The query destination value ranks by popularity the query trails' destinations. In our own representation is defined as:

$$qd(q, d) = S(d, q) \cdot \sum_{p} w \left| \begin{array}{c} \exists\, p = q \rightsquigarrow d_j \rightarrow d \rightarrow n_q \\ d_j, d \in D_s, n_q \in Q_s \\ w \in \{d_j, d, w\} \end{array} \right.$$

where S(d,q) is the *tf.idf* similarity measure between document d and the last query $q \in last(Q_{s'})$ input by the user. Note that the links between documents in the WDG are essentially trail links, but we don't limit these trails to clicks, but extended them with more types of actions. The popularity value is defined by the weight aggregation of all incident links within the paths of the different historical query trails defined between q and d.

Random Walk. Craswell and Szummer [1] exploit the clickthrough data with a random walk algorithm. The random walk computation will end, in theory, with a higher probability on those nodes that previous users found (implicitly) relevant after issuing the query (forward walk approach) or on those documents that represent the information need of the query (backward walk approach). For this computation, a probability of going from node n_k to n_j is needed:

$$P_{t+1|t}(n_k | n_j) = \begin{cases} (1 - s)\, C_{jk} / \sum_i C_{ji} \ \forall k \neq j \\ s \text{ when } k = j \end{cases}$$

where s is the probability of staying in the same node (set to 0.9) and the click count is $C_{ij} = w \in \{n_i, n_j, w\}$, thus taking into considerations the aggregation of implicit evidences. Using these probabilities, we compute a backwards random walk $rw_B(q)$ and a forward random walk $rw_F(q)$, $q \in last(Q_{s'})$. Both random walks were computed using 11 steps.

4. Simulated User Behaviour for Interactive Retrieval Evaluation

To analyse the performance of each recommendation methodology we had to construct a graph pool with implicit data from previous users and evaluate the performance of each recommendation algorithm. The graph pool was constructed by monitoring the interaction of 24 users, mostly postgraduate students and research assistants, with a video retrieval system introduced by Urban et al. [9]. The participants' group consisted of 18 males and 6 females with an average age of 25.2 years and an advanced proficiency with English. Each of the users performed the same selection of four explorative tasks from TRECVID 2006 [8], spending 15 minutes for each task. We decided to use those tasks that performed the worst in TRECVID, mostly due to their multifaceted and ambiguous nature, while still being quite specific, therefore being the most challenging for current multimedia retrieval systems. The four tasks were:

- Find shots with a view of one or more tall buildings (more than four stories) and the top story visible (Task 1)
- Find shots with one or more soldiers, police, or guards escorting a prisoner (Task 2)
- Find shots of a group including at least four people dressed in suits, seated, and with at least one flag (Task 3)
- Find shots of a greeting by at least one kiss on the cheek (Task 4)

Our intention was to analyse if the recommendation algorithms are able to improve the performance of these difficult tasks. As advanced retrieval techniques such as search-by-concept of search-by-example did not perform well on these tasks within TRECVID, here implicit feedback could be a promising approach to aid users with their search. A post search questionnaire confirmed that the tasks were, in general, indeed perceived as difficult for the users, with a special mention for Task 4.

We therefore constructed the implicit pool WDG, which contained the interaction information of each user, including also noisy data, obtained from two training tasks which users performed for ten minutes each. Once we filled the implicit pool with the user data, a natural next step would be to use users to evaluate each system. However, having six different recommendation strategies makes this evaluation step too costly in both time and human resources. Instead of this, we opted to create a simulation framework that used the statistical data mined from the original 24 users. Using this data, we simulated users that interact with a hypothetical extension of the original retrieval system, with the addition of both query and video recommendations.

The evaluation system thus simulates a user interacting with this extension of the original video retrieval system and receiving recommendation from the evaluated algorithm. We used the statistical information from the 24 training users in order to simulate probabilities of the user performing certain types of actions. A new interaction was added: selecting a recommended query. In order to evaluate the recommendation algorithms, we made the following assumption: after a query is launched, users first review the five top recommended results before they continue to look into the query result set. Therefore, the five recommended results are added on top of the result set. Note that there are various recommendation approaches that can be updated as soon as new implicit information is obtained. However, in order to evaluate the algorithms evenly we choose to update the recommendation by issued query. Table 1 shows the probability values obtained from the user study.

Table 1. Probability and normal distribution measures for observed action types

Action type	Probability	Action type	μ	σ
Click relevant result	0.8	Navigation	0.5	2
Click irrelevant result	0.2	Play duration (3 sec interval)	2	3
Tooltip results[1]	0.8	Browsing near keyframes	0.25	1

The simulation system, based on a system introduced by Hopfgartner et al. [3], simulates a user performing one of the four tasks, using ten interactions (i.e. queries) for each task, and interacting with ten documents per query, which were the averages observed during the user experiments. Given the generic recommendation algorithm ra, the steps of each interaction for task t are as follows:

1) With probability $\mathrm{p_q}$ (fixed to 0.6 in our experiments) execute first recommended query $\mathrm{q} \in \mathrm{ra(WDG_{s'})}$, otherwise execute a random query $\mathrm{q} \in \mathrm{Q}$ from task *t*.
2) Collect $\{\mathrm{top5(ra(WDG_{s'}))}, \mathrm{top20(query\ results)}\}$ as the result set of the interaction, and until the user has clicked ten results:
   - With probability $\mathrm{p(tooltip\ result)}$ tooltip result
   - With probability $\mathrm{p(click|relevant)}$ click result
   - If result clicked
     - Simulate browsing steps: $\mathrm{N(\mu(browsing), \sigma(browsing))}$
     - Simulate navigation actions: $\mathrm{N(\mu(navigation), \sigma(navigation))}$
     - Simulate playing duration: $\mathrm{N(\mu(play), \sigma(play))}$

The recommendation algorithm has access to the current session information, i.e. $WDG_{s'}$. Therefore, the recommendation algorithms has access to the interaction sequence, the last input query and the last accessed documents. There is one exception with the query destination algorithm, which does not recommend queries, in this case the queries are always chosen at random.

5. Experiments

The simulations results are discussed in this section. Each recommendation strategy was simulated through 50 runs, which proved to be statistically relevant. Figure 3

[1] In the retrieval system, when the user leaves the mouse on top of a result for one second, a tooltip appears showing the nearby keyframes for the video.

depicts the overall performance of each system, including the baseline system, which is a simulation with no recommendation whatsoever. The evaluation measure is the average of the P@N points for every run. Following an interactive evaluation methodology, we take as final result set the rank-based merge of the results sets for each of the 10 interactions, which include on top the first five recommended results.

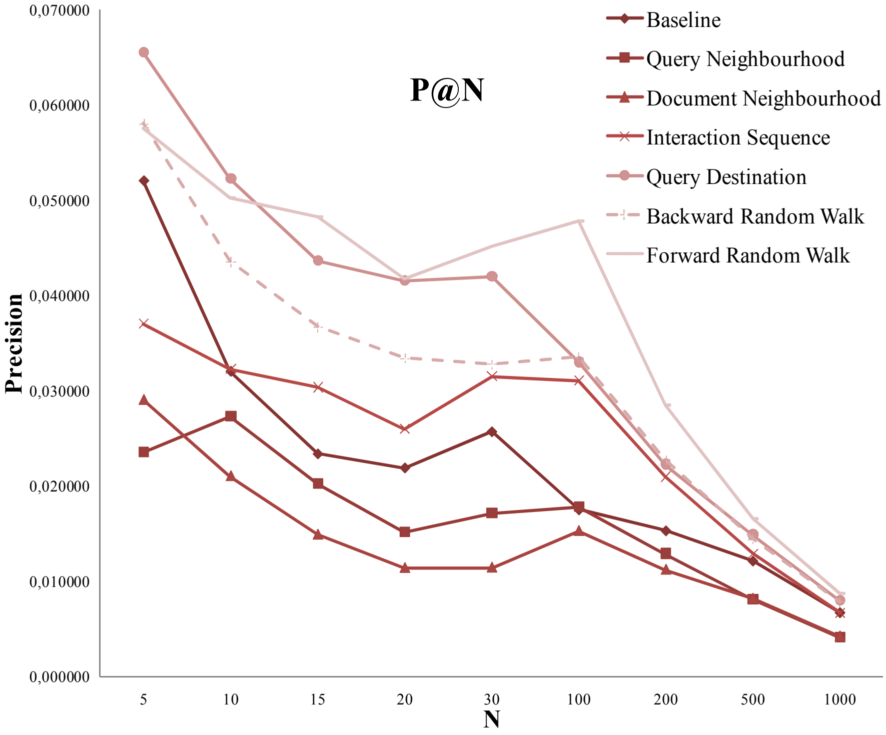

Fig. 3. Precision cut-off points for each recommendation strategy.

The recommendation strategy that overall appears to perform best is the query destination recommendation, followed by the interaction sequence and the forward random walk. One singular characteristic of the query destination approach is that the similarity between the last query and the recommended documents is taken into consideration, apart from the popularity measure. The interaction sequence algorithm performance does highlight the importance of exploiting the search and query trails similarities. The random walk approach also exploits these trails. This could be the reason why the forward random walk performance is close to the interaction sequence. Surprisingly, the backward random walk has a sensible loss of performance against the forward approach, although Craswell and Szummer report the contrary. The poor performance of the neighbourhood based strategies suggests that the link directionality has indeed to be taken into consideration, as well as the density of the paths that point from the node to its neighbours.

Although the query destination performs the best on average, the results per topic show that the performance of each algorithm varies meaningfully for each task. Figure 4 shows the performance of this four recommendation strategies for each topic.

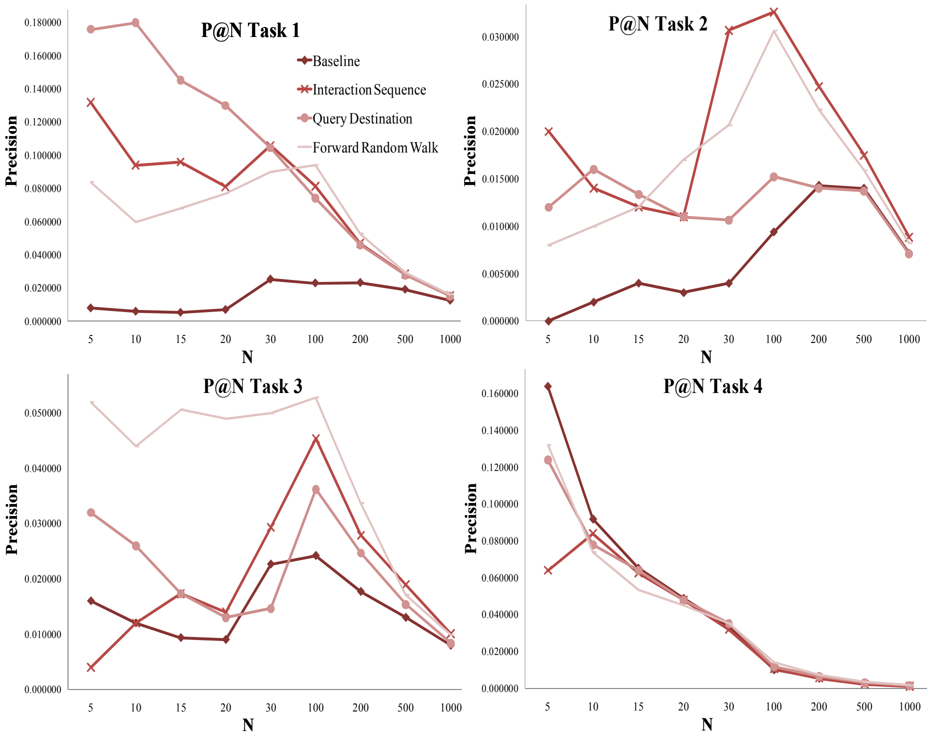

Fig. 4. Precision cut-off point for the best four strategies and the four evaluated tasks

Note that there is a different algorithm that performs better in the tree first tasks: query destination in Task 1, interaction sequence in Task 2 and forward random walk in Task 3. Finally, no recommendation approach was able to outperform the baseline in Task 4. The reason was probably that users showed an erratic behavior in this task, as they confessed a great difficulty on meeting the semantic of the task at hand with the videos' textual metadata.

6. Conclusion

In this work, we have explored the exploitation of community usage feedback information to aid users in difficult video retrieval tasks. The presented integrated model includes an efficient and scalable way of representing this past information and, even more important, eases the use of any desired recommendation strategy. The implicit graph representation has proven to facilitate the analysis of the diverse types of implicit actions that a video retrieval system can provide, thus allowing an easy extension. In addition, an evaluation framework is introduced, of which the main goal is to facilitate evaluation of new recommendation strategies.

Using the presented evaluation framework, we have reported a set of experiments on different recommendation approaches, either created by us, or adapted from related work. We have observed that the performance of each evaluated strategy varied significantly with each specific task, indicating that there could be different complementary approaches for video retrieval recommendation. The use of the overall popularity of the document, the exploitation of interaction trails and taking into consideration the last submitted query were some of the characteristics of the evaluated recommendation strategies that performed the best.

7. Acknowledgements

This research work was partially supported by the European Commission under contracts: K-Space (FP6-027026), SALERO (FP6- 027122) and SEMEDIA (FP6 - 045032). Authors acknowledge Martin Halvey for help with the experiments and for fruitful discussions.

References

1. Craswell, N., Szummer, M.: Random walks on the click graph. SIGIR '07: Proceedings of the 30th annual international ACM SIGIR 2007. ACM 239-246
2. Finin, T. W.: GUMS: A General User Modeling Shell. In: User Models in Dialog Systems. Berlin, Heidelberg: Springer Verlag (1989) 411-430
3. Hopfgartner, F., Urban, J., Villa, R., Jose, J.: Simulated Testing of an Adaptive Multimedia Information Retrieval System. Content-Based Multimedia Indexing, 2007. CBMI '07. International Workshop, Bourdeaux, France (2007) 328-335
4. Hopfgartner, F., Jose, J.: Evaluating the Implicit Feedback Models for Adaptive Video Retrieval. ACMMIR '07. 323-331
5. Jaimes, A., Christel, M., Gilles, S., Ramesh, S., Ma, W.: Multimedia Information Retrieval: What is it, and why isn't anyone using it?. ACM MIR'05. ACM Press 3-8
6. Kelly, D., Belkin, N. J.: Display time as implicit feedback: understanding task effects. ACM SIGIR 2004. ACM , 377-384
7. Kelly, D., Teevan, J.: Implicit Feedback for Inferring User Preference: A Bibliography. SIGIR Forum 37(2): 18-28 (2003)
8. Over, P., Ianeva, T.: TRECVID 2006 Overview. TRECVid 2006 – Text Retrieval Conference TRECVID, Gaithersburg, MD, 2006
9. Urban, J., Hilaire, X., Hopfgartner, F., Villa, R., Jose, J., Chantamunee, S., Goto, Y.: Glasgow University at TRECVID 2006. TRECVID, Gaithersburg, MD, 2006
10. White, R., Bilenko, M., Cucerzan, S.: Studying the use of popular destinations to enhance web search interaction. ACM SIGIR '07. ACM Press 159-166
11. White, R., Jose, J. M., van Rijsbergen, C. J., Ruthven, I.: A Simulated Study of Implicit Feedback Models. ECIR '04. Springer Verlag (2004) 311-326