Procrastination is the Thief of Time: Evaluating the Effectiveness of Proactive Search Systems

Procheta Sen
ADAPT Centre. School of Computing
Dublin City University
procheta.sen2@mail.dcu.ie

Debasis Ganguly
IBM Research Lab
Dublin Ireland
debasis.ganguly1@ie.ibm.com

Gareth Jones
ADAPT Centre. School of Computing
Dublin City University
Gareth.Jones@dcu.ie

ABSTRACT

Traditional search system users actively interact with the system to complete their search task. We believe that the next generation of search systems will see a shift towards proactive *understanding* of user intent based on analysis of user activities. Such a proactive system could start recommending documents that are likely to help users accomplish their tasks without requiring them to explicitly submit queries to the system. We propose a framework to evaluate such a search system. The key idea behind our proposed metric is to aggregate a correlation measure over a search session between the *expected outcome*, which in this case refers to the list of documents retrieved with a true user query, and the *predicted outcome*, which refers to the list of documents recommended by a proactive search system. Experiments on the AOL query log data show that the ranking of two sample proactive IR systems induced by our metric conforms to the expected ranking between these systems.

CCS CONCEPTS

• **Information systems → Query intent**;

KEYWORDS

Evaluation Metric, Proactive Search, Contextual Recommendation

ACM Reference format:
Procheta Sen, Debasis Ganguly, and Gareth Jones. 2018. Procrastination is the Thief of Time: Evaluating the Effectiveness of Proactive Search Systems . In *Proceedings of The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, July 8–12, 2018 (SIGIR '18),* 4 pages.
https://doi.org/10.1145/3209978.3210114

1 INTRODUCTION

Users typically interact with a search system to complete a task within a search session. Examples of such search tasks are learning or gathering knowledge on a topic, or a goal driven task, such as making arrangements for traveling to a conference etc. A task often consists of a number of granular sub-tasks, each with a corresponding information need. For each such information need, a user needs to submit a query to the search system, which then returns a ranked list of documents, which could potentially help the user with relevant information to complete the sub-task. A traditional search system involves considerable user effort in search task interactions, where the user need to manually execute a set of queries. Now, consider an alternative search system designed to minimize this user interaction. One of the useful features of such a system would be recommendation of a set of documents to the user that may be relevant to his next sub-task (information need). To the best of our knowledge, no reported research on proactive search systems (PSSs) considers methods to suggest relevant documents ahead of time without requiring the user to enter a query expressing their next information need.

Existing research on proactive systems include tracking eye movements over document titles to proactively find related documents [7], proactively suggesting attachments of an email from previous emails [9], and utilizing screen surveillance signals for proactive retrieval of relevant information [10]. A set of desirable characteristics of a proactive IR interface is described in [1]. Studies on related topics, such as search task classification [6] and query prediction [3, 4] can serve as useful inputs to the development of PSSs. Despite the presence of a considerable number of reported studies, there exists no standard definition of proactivity in IR. For our work, we define a IR system to be *proactive* if *given the context of a small number of queries towards the beginning of a search task, the system is able to recommend documents that are likely to be relevant for accomplishing their current task.* As the title of our paper suggests, unlike traditional search systems a PSS does not *procrastinate* possible IR actions until the user inputs a query.

In this paper, we develop a framework to evaluate the effectiveness of a PSS within a search session. The key idea is to aggregate a correlation measure over a search session between the *expected outcome*, which in this case refers to the list of documents retrieved with a true user query, and the *predicted outcome*, which refers to the list of documents recommended by a proactive system. A PSS evaluation metric needs to take into account of the fact that different search systems can start recommending documents at different points in time. For instance, one system could start suggesting after the first query of each session, whereas the other could wait to accumulate more contextual data (e.g. till the third query in a session) before it starts proactive document recommendation.

2 PROACTIVE SEARCH SYSTEM OUTLINE

In this section, we describe the outline of a PSS and methodologies that may be used for contextual recommendation in a PSS. Given a context of previously entered queries by a user within a

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210114

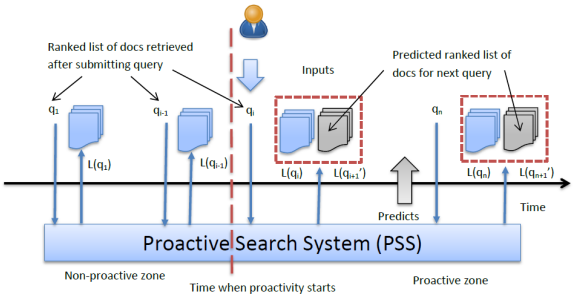

Figure 1: Schematic diagram of a proactive search system.

search session, say, $q_1, \ldots, q_{n-1}$, the objective of a PSS is to suggest documents retrieved in response to the query q_n without the user explicitly submitting q_n to the system. A simple approach towards developing a PSS is to make use of a query prediction system, which given a context of previous search queries within a session, $q_1, \ldots, q_{n-1}$, predicts the next query $\hat{q_n}$. Such a system is schematically described in Figure 1.

Without loss of generality, we assume that a PSS may start recommending documents after any length of context of queries executed within a search session as shown by the dotted vertical line in Figure 1. Before this point in time, a PSS behaves identically to a standard search system, i.e. given a query q_i, it returns a ranked list of documents $L(q_i)$. From the point of inception of proactivity, in addition to the ranked list for the current query $L(q_i)$, a PSS starts recommending a ranked list of documents, $L(q'_{i+1})$, based on, generally speaking, a set of predicted next queries. A PSS system can be considered to perform effectively if the documents in this predicted list overlap with those that the user would see after explicitly executing the next query in sequence, q_{i+1}. The objective and the challenge of a PSS is then to accurately estimate (*ahead of time*) the documents retrieved in response to the next information need of the user within the search session.

3 A PSS EVALUATION METRIC

In this section, we first introduce three desirable characteristics of an evaluation metric for a fair and meaningful comparison between PSSs before formally describing the metric.

3.1 Desirable Characteristics of the Metric

C_1: Rewards computed exclusively on similarities between predicted and groundtruth lists of documents. The first desirable characteristic of the PSS evaluation metric is that it should not attempt to evaluate the effectiveness of any intermediate step opaque to the system output. This is because it should be possible to evaluate PSSs that do not employ any intermediate step. Thus, the evaluation metric should only consider the predicted list of recommended documents and compare it with the list of documents that would have been retrieved if the user had actually executed the true query in the system.

C_2: Normalized cumulative rewards. The second characteristic of the metric is that the rewards of matching the expected list with the predicted list should be accumulated for each query within a search session that follows the recommendation inception point. For instance, the rewards should be accumulated over all queries to the right of the vertical line in Figure 1, i.e. queries $q_i, \ldots, q_n$. This characteristic ensures that a PSS is considered effective if it performs consistently well throughout the search session.

The metric should also consider the fact that different PSSs can have different inception points (no fixed value of i for the vertical line in Figure 1). Consequently, the accumulated rewards should be normalized with respect to the proactivity period, or more precisely, the number of queries for which a PSS suggests documents ahead of time within a search session. As an example, in Figure 1 this normalization value is $n - i + 1$.

C_3: Rewards weighted on context length of queries. The third important characteristic of a PSS metric is that it should be able to favour systems that start recommending early (i.e. capable of addressing the cold-start problem). This is because from a user experience point of view, such a system should be considered more proactive. In contrast, this metric should penalize a system which waits for a large number of queries to be executed by a user before starting to recommend documents proactively.

3.2 Formal Definition

We use the following notations in the description of our proposed metric. We denote the effectiveness measure of a PSS as $P(\pi, M, n)$, which is parameterized by the π-index and the number of recommended documents M over a search session of n queries.

#Queries in a session	n
Proactivity inception index (π-index)	π
#Documents recommended by a PSS	M

3.2.1 Reward function. The reward function outlined in Section 3.1 is a measure of how similar the predicted list of documents is to the groundtruth list of documents. The reward function, $r(k)$, is thus defined for a particular query q_k within a session, where $i \leq k \leq n$. Another parameter of the reward function is the number of top ranked documents, M, returned by a PSS based on which the similarity between the predicted and the groundtruth lists is computed. The reward function is presented in a general form in Equation 1, where ϕ denotes a function to measure the similarity between the predicted and the groundtruth ranked lists of documents (each of cardinality M) for the next query in sequence, q_{k+1}.

$$r(k, M) = \phi(L(q_{k+1}), L(q'_{k+1})). \tag{1}$$

This definition of the reward function is consistent with characteristic C_1. We propose two specific variants of the function ϕ. The first of the two is based on reciprocal of the highest rank at which a document from $L(q_{k+1})$ is seen in $L(q'_{k+1})$.

$$\begin{aligned} \phi_1(L(q_{k+1}), L(q'_{k+1})) &= \frac{1}{m}, \;\; m = \min_{j \in H} j \\ &= 0, \;\; \text{if } H = \{\}, \;\; \text{where } H = L_j(q'_{k+1}) \cap L(q_{k+1}). \end{aligned} \tag{2}$$

A problem with the reciprocal rank (RR) based reward function is that it considers each document in the reference list to be equally relevant. However, it is more intuitive to think that a PSS where the first predicted document is the 1^{st} in the reference list, is preferable to one in which the first predicted document is the 10^{th} item in the reference list. To capture the intuition of rank correlation, we

propose a second variant of the reward function that employs the Spearman's rank correlation coefficient (normalized in $[0, 1]$ so as to ensure positive rewards) between the predicted and the reference lists. This is shown in Equation 3.

$$\phi_2(L(q_{k+1}), L(q'_{k+1})) = \frac{1}{2}\big(1 + \rho(L(q_{k+1}), L(q'_{k+1}))\big). \quad (3)$$

3.2.2 Accumulating reward functions. The reward functions are accumulated over a session as shown in Equation 4.

$$P(\pi, M, n) = \frac{1}{n - \pi} \sum_{k=\pi}^{n} \frac{r(k, M)}{k}. \quad (4)$$

As per C_2, the total reward function is normalized by the total number of queries with proactivity retrieval, i.e. $n - \pi$. It is easy to see that as per C_3, the metric in Equation 4 discounts rewards that are accumulated towards the end of a session by the factor $\frac{1}{k}$. This way the metric tends to favour systems that become proactive towards the beginning of a session. According to the reward function used, we get two variants of the metric, which we name as **PREVAL-RR** (RR-based proactivity evaluation) and **PREVAL-ρ** (Spearman's coefficient-based proactivity evaluation), respectively.

3.2.3 Analysis for variations in π-index. In this section, we analytically show that it is possible for our metric PREVAL (Equation 4) to score a system B higher than A even if A starts recommending earlier than B. This analysis ensures fairness between two PSSs with different π-index values. For simplicity, we base our analysis on the RR-based reward function, i.e. PREVAL-RR. Let the two systems be $A(\pi_A, M, n)$ and $B(\pi_B, M, n)$, where $\pi_A < \pi_B$. Let the reward function of system A for query q_k be m_k^A and that of B for the same query be m_k^B. If system B outperforms A, then

$$P_B(\pi_B, M, n) > P_A(\pi_A, M, n), \quad (5)$$

which on substitution from Equation 4 gives

$$\sum_{k=\pi_B}^{n} \frac{r_B(k, M)}{k} > \sum_{k=\pi_A}^{n} \frac{r_A(k, M)}{k} \geq 0. \quad (6)$$

Substituting Equation 2 into 6 gives

$$\sum_{k=\pi_B}^{n} \frac{1}{k}\left(\frac{1}{m_k^B} - \frac{1}{m_k^A}\right) > \sum_{k=\pi_A}^{\pi_B - 1} \frac{1}{k m_k^A} \geq 0. \quad (7)$$

Now, with reference to Equation 7, we consider two scenarios with different ideal rankings between systems A and B.

Case-1. System A does not perform worse than B from the point B starts prediction (in which case the ideal rank is $A > B$). To see if this could happen, we substitute $m_k^A \leq m_k^B, \forall k = \pi_B, \ldots, n$ and get

$$0 \leq \sum_{k=\pi_A}^{\pi_B - 1} \frac{1}{k m_k^A} \leq 0, \quad (8)$$

which shows that $P_B(\pi_B, M, n) > P_A(\pi_A, M, n)$ is a contradiction. In other words, the metric indeed scores system A better. This is desirable because B's predictions were late and not better than A.

Case-2. System A performed well before B's recommendations after which B performed well and A poorly. The ideal rank in this case is $A < B$. To see what could happen in this scenario, let $m_k^A = 1 \forall k = \pi_A, \ldots, \pi_B - 1$ (A's recommendation was perfect initially), $m_k^A = \epsilon \; \forall k = \pi_B, \ldots, n$ (A fails to recommend well after B starts), and $m_k^B = 1 \; \forall k = \pi_B, \ldots, n$ (B recommends perfectly throughout). Substituting these values in Equation 7, and using the identity $\log(n) \approx \sum_{i=1}^{n} \frac{1}{i}$,

$$\sum_{k=\pi_B}^{n} \frac{1}{k}(1 - \epsilon) > \log(\pi_B - 1) \Rightarrow (1 - \epsilon)\log(\frac{n}{\pi_B}) > \log(\pi_B - 1)$$
$$\Rightarrow \epsilon < \log(\frac{n}{\pi_B(\pi_B - 1)}) \quad (9)$$

Equation 9 suggests an upper bound on A's effectiveness to satisfy the condition that B is scored better than A in terms of PREVAL. This ensures that the metric is fair because it can score B better than A if B (from its inception point which is later than A's) performs significantly better than A. Otherwise A is favored by PREVAL.

4 EMPIRICAL EVALUATION

We first describe three sample proactive IR systems with an intuitive preferred ranking order. The objective of the experiments is to compute the values of our proposed metric PREVAL (Equation 4) on these systems, and see if the ranking induced by our metric corresponds to the intuitive preference among the systems.

4.1 Sample proactive IR systems

While developing a PSS is not the objective of the paper, for the sake of comparisons between different systems, we describe two simple PSS approaches. We subsequently compute our proposed PSS effectiveness metrics under a number of different settings.

Relevance Model based PSS (RM-PSS). The first approach employs a relevance model [5] to obtain the predicted ranked list $L(q'_i)$ given the previous query q_{i-1} and the ranked list retrieved in response to it, namely $L(q_{i-1})$. The relevance model employs relevance feedback to improve retrieval effectiveness. In this case, we argue that the expanded query obtained from the estimated distribution $P(w|Q)$ (see [5] for details on the notations) is representative of a reformulated query, which is then used to retrieve the recommended documents $L(q'_i)$. We call this system RM-PSS (Relevance Model based PSS).

Query Prediction based PSS (QP-PSS). The second PSS approach employs a supervised query prediction algorithm to compute a set of most likely query reformulations given a current query. The probability of query reformulations is trained by constructing a query flow graph (QFG) from a large number of real-user queries in a query log [2]. In a QFG, each node represents a query and the weight of each edge (q_i, q_j) indicates the likelihood of q_j following q_i within a search session. This QP-PSS based system obtains a ranked list of k predicted next queries, $R_k(q_{i-1})$, from the current query q_{i-1}, sorted in descending order by the transition probabilities estimated from a QFG. It then retrieves documents with each predicted query $q \in R_k(q_{i-1})$ and outputs a weighted COMBSUM [8] of the lists. We experiment with two variants of COMBSUM weights, one with uniform weights, denoted as **QP-PSS-U**, and the other with weights set to probabilities of reformulation estimated from the QFG, denoted as **QP-PSS-W**. With respect to the three approaches described, the intuitive ranking between them should be **RM-PSS** < **QP-PSS-U** < **QP-PSS-W**.

Table 1: PREVAL metrics computed over the evaluation set.

Proactive System	PREVAL-RR	PREVAL-ρ
RM-PSS	0.0235	0.4955
QP-PSS-U	0.0503	0.4980
QP-PSS-W	0.0503	0.5004

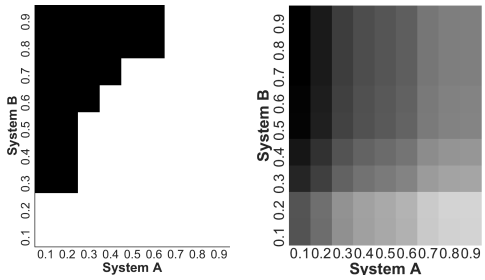

Figure 2: Comparison of PREVAL-RR values of RM-PSS (x-axis) and QP-PSS-W (y-axis), π-index values being shown alongside the axes. i) Left: A black cell indicates that $P_A > P_B$. ii) Right: Each cell plots the value of $\frac{P_A - P_B}{\max(P_A, P_B)} \in [-1, 1]$, darker shades denoting values towards 1 ($P_A > P_B$).

4.2 Experiment Settings and Results

We used a subset of the AOL query log data from the period of Mar'06 to Apr'06 for our experiments, which comprises over $6M$ queries. The first $5M$ queries (preprocessed using a Porter stemmer) were used as a training set to construct a QFG, which was then used in the QP-PSS approaches. A session length of 26 minutes, as prescribed in [6], was used as the adjacency criteria to construct the CFG. As our evaluation set to measure the relative performances of the three PSS systems, we use a random subsample of 50 query sessions. Each session of the evaluation set comprises at least 5 queries (after removing duplicate queries within a session). The average session length in our evaluation set is 8.12.

Table 1 shows the results of applying the two variants of the metric PREVAL on the test set of 50 query sessions. To compute the reward functions, M was set to 10 (see Equation 1). It can be seen that RM-PSS performs the worst in terms of both the RR and ρ-based PREVAL scores. This is to be expected since RM-PSS only uses the information from the current query and the top documents retrieved in response to it. Our metric is also able to rank QP-PSS-W better than QP-PSS-U. This is also expected since the latter considers the contribution from all predicted queries uniformly when constituting the final list of suggested documents. An interesting observation is that PREVAL-ρ is better able to distinguish between QP-PSS-W and QP-PSS-U (in terms of the relative differences between the scores) in comparison to PREVAL-RR. This shows that the rank correlation based reward acts as a better estimate for PSS effectiveness.

Figure 2 shows the relative differences between the PREVAL-RR values for systems A (RM-PSS) and B (QP-PSS-W), i.e., $\delta = \frac{P_A - P_B}{\max(P_A, P_B)}$, for a range of π-index values π_A and π_B. To report an average over sessions of varying lengths, we plot relative proportions of π_A and π_B values within a range of 0.1 to 0.9. A sample cell $(\pi_A, \pi_B) = (0.2, 0.6)$ in both plots of Figure 2 indicates that A started suggesting documents after 20% queries within a session were executed, whereas B waited until 60% of the queries had been executed before initiating proactive suggestions. Each of the $9 \times 9 = 81$ cells of Figure 2 represents an experimental observation. The left plot indicates whether RM-PSS is the winner (black cell). The right plot shows the relative differences between the PREVAL-RR values of RM-PSS and QP-PSS-W. If RM-PSS performs better, this relative difference is close to 1 (represented by darker shades), whereas lighter shades indicate that QP-PSS-W is better.

It can be seen that if QP-PSS-W starts predicting too late in comparison to RM-PSS (cells corresponding to the upper left region of the plots in Figure 2), RM-PSS is the winner. This happens because the rewards accumulated from a small number of proactive suggestions in these cases is not sufficient to outweigh the effectiveness of the early suggestions coming from RM-PSS. This in turn shows that the metric is able to prefer cold-start proactive systems. On the other hand, the system QP-PSS-W (being a more effective system) often wins in the central and the bottom-right part of the plots in Figure 2. This can be seen from the white regions in the left plot, and the lighter shades in the right plot of Figure 2, with the relative differences getting larger towards the bottom-right. This shows that the metric is not overly biased towards cold-start systems and can favour more effective systems with higher π-index values.

5 CONCLUDING REMARKS

Our analysis shows that the proposed metric PREVAL is able to fairly compare between two PSSs that become proactive at different times during search sessions and the the ranking of systems induced by PREVAL conforms to the intuitive expectation. We believe that our proposed PSS evaluation framework will help in developing effective proactive systems in the near future.

Acknowledgement. This work was supported by Science Foundation Ireland as part of the ADAPT Centre (Grant 13/RC/2106) (www.adaptcentre.ie) at Dublin City University.

REFERENCES

[1] D. Billsus, D. M. Hilbert, and D. Maynes-Aminzade. Improving proactive information systems. In *Proc. of IUI'05*, pages 159–166, 2005.
[2] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: Model and applications. In *Proc. of CIKM 2008*, pages 609–618, 2008.
[3] H. Feild and J. Allan. Task-aware query recommendation. In *Proc. of SIGIR 2013*, pages 83–92, 2013.
[4] W. Kong, R. Li, J. Luo, A. Zhang, Y. Chang, and J. Allan. Predicting search intent based on pre-search context. In *Proc. of SIGIR 2015*, pages 503–512, 2015.
[5] V. Lavrenko and B. W. Croft. Relevance based language models. In *Proc. of SIGIR '01*, pages 120–127, 2001.
[6] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei. Discovering tasks from search engine query logs. *ACM Trans. Inf. Syst. 2013*, pages 14:1–14:43, 2013.
[7] K. Puolamäki, J. Salojärvi, E. Savia, J. Simola, and S. Kaski. Combining eye movements and collaborative filtering for proactive information retrieval. In *Proc. of SIGIR 2005*, pages 146–153, 2005.
[8] J. A. Shaw, E. A. Fox, J. A. Shaw, and E. A. Fox. Combination of multiple searches. In *TREC-2*, pages 243–252, 1994.
[9] C. Van Gysel, B. Mitra, M. Venanzi, R. Rosemarin, G. Kukla, P. Grudzien, and N. Cancedda. Reply with: Proactive recommendation of email attachments. In *Proc. of CIKM'17*, pages 327–336, 2017.
[10] T. Vuong, G. Jacucci, and T. Ruotsalo. Watching inside the screen: Digital activity monitoring for task recognition and proactive information retrieval. *Proc. of IMWUT'17*, 1(3):109:1–109:23, 2017.