A Markovian Approach to Evaluate Session-based IR Systems

David van Dijk[1], Marco Ferrante[2][0000−0002−0894−4175], Nicola Ferro[2][0000−0001−9219−6239], and Evangelos Kanoulas[1][0000−0002−8312−0694]

University of Amsterdam, The Netherlands {d.v.van.dijk@hva.nl,
e.kanoulas@uva.nl}
University of Padua, Italy {ferrante, ferro}@dei.unipd.it

Abstract. We investigate a new approach for evaluating session-based information retrieval systems, based on Markov chains. In particular, we develop a new family of evaluation measures, inspired by random walks, which account for the probability of moving to the next and previous documents in a result list, to the next query in a session, and to the end of the session. We leverage this Markov chain to substitute what in existing measures is a fixed discount linked to the rank of a document or to the position of a query in a session with a stochastic average time to reach a document and the probability of actually reaching a given query. We experimentally compare our new family of measures with existing measures – namely, session DCG, Cube Test, and Expected Utility – over the TREC Dynamic Domain track, showing the flexibility of the proposed measures and the transparency in modeling the user dynamics.

Keywords: Information Retrieval · Evaluation · Sessions · Markov Chains

1 Introduction

Evaluation measures are an intrinsic part of experimental evaluation. Even if a growing attention is called in the field for developing stronger theoretical foundations [1,2,3,9,12], they are often formulated and justified in a somewhat informal and intuitive way rather then being based on well-founded mathematical models. Carterette [4] has made a post-hoc attempt to propose a unifying framework which explains modern evaluation measures based on three components: a browsing model, a model of document utility, and a utility accumulation model. According to this framework, measures such as RBP [21], DCG [13] or ERR [5] can be defined as expectations of the utility, total utility, and effort, respectively over a probabilistic space defined by the chance of a user to browse the next in rank document in a provided ranking.

When it comes to session search, defining an evaluation measures based on a rigorous mathematical model becomes an even more challenging task. Session search involves multiple iterations of searches in order for a user to accomplish a complex information need, with multiple queries being issued or reformulated and multiple runs of search results being returned by the search engine and examined by the user. The difficulty of defining session evaluation measure comes from the question of how to assess the value of a relevant document not only along a certain ranking but across rankings of different queries within a session, or, in other words, how to mathematically model the dynamics of a user across the entire search session. Session evaluation measures proposed in the literature, such as the session Discounted Cumulated Gain (sDCG) [14], the Expected Utility (EU) [28], the Expected Session measures (esM) [15], or the Cube Test (CT) [19], typically extend single ranking evaluation measures in an ad-hoc manner that results in a lack of a sound, clear, and extensible mathematical framework. In this paper, we focus on the following research question: *How can we mathematically model user dynamics over a multi-query session and inject them into an effectiveness measure?*

To answer this question, we represent queries in a session and documents within a ranked result list for a query as states in a Markov chain. We then define an event space of user actions when searching: (a) moving along a ranking of documents, (b) reformulating her query, and (c) abandoning the search session, and the probabilities of each one of these actions. Different instantiations of these probabilities give rise to different transition probabilities among the states of the Markov chain which allow us to model the different and perhaps complex user behaviors and paths in scanning the ranked result lists in a session.

Finally, we conduct a experimental evaluation of the Markov Session Measures (MsM) using three standard Text REtrieval Conference (TREC) collections developed by the Dynamic Domain Track (DDT) [26,27], based on which we show the flexibility of MsM in modeling a wide variety user dynamics, as well as how close MsM is to existing measures in terms of user dynamics.

2 Related Work

Järvelin et al. [14] extended the Discounted Cumulated Gain (DCG) measure to consider multi-query sessions. The measure – session Discounted Cumulated Gain (sDCG) – discounts documents that appear lower in the ranked list for a given query, as well as documents that appear in follow up query reformulations. sDCG underlies a deterministic user model with the user stepping down the ranked list until a fixed reformulation point and then moving to the next query in the session until all ranked lists in the session have been scanned. Luo et al. [19] proposed the Cube Test (CT) which is also based on a deterministic user model of browsing a ranked list up to a certain reformulation point and then continuing to browse the results of the next query. Departed from the work of Smucker and Clarke [22,23] who defined the Time-Biased Gain (TBG) measure, Luo et al. inject the time it takes users to read relevant documents as a discounting factor of the utility of a document. Differently from the aforementioned deterministic user models, both Yang and Lad [28] and Kanoulas et al. [15] took a probabilistic approach and defined a session measure as an expectation over a set of possible browsing paths. Yang and Lad introduced Expected

Utility (EU) and, to define the probability of a user following a certain path, they followed the Rank-Biased Precision (RBP) approach [21], replacing RBP's stopping condition with a reformulation condition. Kanoulas et al., instead, first defined a reformulation probability that allows for an early abandonment and then, for those queries that are being realized, they introduced a stepping-down probability, similar to RBP. Our approach differs from sDCG and CT by considering a probabilistic event space of user actions across the states of a Markov chain, which represent documents in the different ranked lists and positions within them. Further, it offers a solid mathematic framework that from Yang and Lad and Kanoulas et al. by avoiding the unreasonable assumptions their approaches make, but also offering the ability to extend the framework to more advanced user dynamics.

Markov-based approaches have been previously exploited in IR, for example: Markov chains have been used to generate query models [17], for query expansion [7,20,25], and for document ranking [8], or to address the placement problem in the case of two dimensional results [6]. Ferrante et al. [10] use Markov chains to define evaluation measures over a single ranked list. This work is an extension of their work to session retrieval. However, differently from their work that depends on the computation of an invariant distribution and which makes the assumption that there is no absorbing state, our work takes a random walk approach and assumes the presence of an absorbing state.

Finally, according to a conducted laboratory user study, Liu et al. [18] have recently suggested some desirable features of a session-based evaluation measure: (1) the most useful document in a query is the most important; (2) the weighting function between queries should be normalized; (3) the primacy effect is not suitable for session evaluation; (4) the recency effect has a stronger influence on user's session satisfaction. Our MsM measure addresses some of the requirements formulated by Liu et al.: (1) because it handles graded relevance and an higher gain can be assigned to the most useful document; (2) because modelling the whole session with a single Markov chain seamlessly normalizes scores across queries; (3) and (4) because by setting appropriate transition probabilities and discount functions, it is possible to smooth the effect of the first documents (primacy) and emphasise the importance of latter queries (recency).

3 The Model

Section 3.1 introduces our Markovian model of the user dynamics over a multi-query session. Section 3.2 exploits this Markovian model to define the Markov Session Measure (MsM) which can be used to evaluate session-based IR systems.

3.1 Multi-query session dynamics

For a given task, a user can generate a sequence of queries, each of which originates a ranked list of documents. Given D the whole document corpus and $N \in \mathbb{N}$ the length of a run, $D_j(N) = \{(d_{1,j}, \ldots, d_{N,j}) : d_{n,j} \in D, d_{n,j} \neq$

$d_{m,j}$ for any $m \neq n\}$ is the ordered set of documents retrieved by a system run for the j-th query. The sets $D_j(N)$ for $j \in \mathbb{N}$ may not be disjoint and the same document may appear in many queries. Without loss of generality we assume that every run has the same length N.

Let $k \in \mathbb{N}$ the number of queries in a session. The whole search session is defined as a matrix of documents, where columns are the the runs $D_1(N), \dots, D_k(N)$ corresponding to each query

$$\begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} & \dots & d_{1,k} \\ d_{2,1} & d_{2,2} & d_{2,3} & \dots & d_{2,k} \\ d_{3,1} & d_{3,2} & d_{3,3} & \dots & d_{3,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{N,1} & d_{N,2} & d_{N,3} & \dots & d_{N,k} \end{bmatrix}$$

The user moves among the documents according to some dynamics, that we assume to be Markovian, i.e. the user decides which document to visit only on the basis of the last document considered. Moreover, we assume that the user starts her search from the first document in the first query, i.e. first row and first column, as typically assumed by any evaluation measure. Then, she moves among the documents in the first column until she decides to change column, i.e. to reformulate the query, or to abandon the search session. In case of query reformulation, she passes to the next result list and, as before, she starts from from the first document of the subsequent column, i.e. the next query, and so on until she ends the search.

We define the sequence of positions of the documents visited by the user as a stochastic process, $(X_n = (X_n^1, X_n^2))_{n \geq 1}$, where $X_n = (i, j)$ means that the n-th document visited by the user is $d_{i,j}$, the i-th document of the j-th column. We assume that this process is a Markov chain on the state space $S = \{1, \dots, N\}^\infty \cup \{(F, j), j \in \mathbb{N}\}$ where $X_n = (F, j)$ represents the fact that the user ends his search after visiting $n-1$ documents and formulating j queries. The transition matrix of this Markov chain

$$p_{(i_n, j_n),(i_{n+1}, j_{n+1})} = \mathbb{P}[X_{n+1} = (i_{n+1}, j_{n+1}) | X_n = (i_n, j_n)] \ ,$$

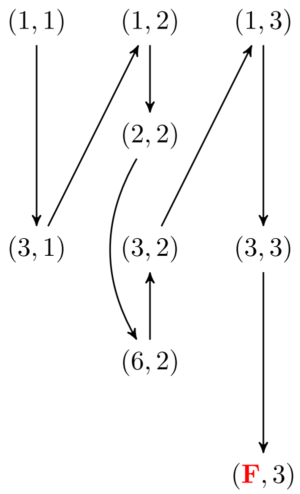
undergoes these constraints:

1. $p_{(i_n, j_n),(i_{n+1}, j_{n+1})} = 0 \ \text{ if } \ j_{n+1} \neq j_n, j_n + 1$, i.e. the user can either move within a column of documents or pass to the next one;
2. $p_{(i_n, j_n),(i_{n+1}, j_n + 1)} = 0 \ \text{ if } \ i_{n+1} \neq 1$, i.e. when the user leaves a column, she goes to the first document of the next one;
3. $p_{(i_n, j_n),(F, j_n)} > 0$ for any $i_n \neq F$;
4. $p_{(F, j_n),(F, j_n)} = 1$ for any j_n, i.e. the states (F, j)'s are all *absorbing*.

Example 1. Let us assume that the stochastic process $(X_n)_{n \geq 1}$ takes the following values:

$$X_1 = (1,1), X_2 = (3,1), X_3 = (1,2), X_4 = (2,2), X_5 = (6,2),$$
$$X_6 = (3,2), X_7 = (1,3), X_8 = (3,3), X_9 = (F,3).$$

This means that the user performed 3 queries and considered 8 documents before stopping, as shown in the following graph



In order to determine how many queries have been issued and how long the search lasted, we define the following sequence of stopping times. Recall that the *stopping time* for a Markov chain $(X_n)_{n\geq 1}$ is a random variable T with values in $\mathbb{N}\cup\{\infty\}$ such that for any $n\in\mathbb{N}$ the event $\{T=n\}$ depends only on $\{X_m, m\leq n\}$. The stopping time

$$H=\inf\{n\geq 1 : X_n^1=F\}$$

determines the number of steps done by the process, with the convention that $\inf\varnothing=\infty$. It allows us to define the (random) number K of queries performed during the search

$$K=X_H^2$$

K is the second component of the process $(X_n, n\in\mathbb{N})$ once absorbed in $(F,\cdot)$.

Then, we define the random times to leave any query as

$$\begin{aligned}
H^1 &:= \inf\{n\geq 1 : X_n^2=2\}\\
H^2 &:= \inf\{n\geq 1 : X_n^2=3\}\\
&\vdots\\
H^{K-1} &:= \inf\{n\geq 1 : X_n^2=K-1\}
\end{aligned}$$

Thanks to these stopping times, we are able to determine how many documents of any query have been visited by a user. Indeed, defined $H^0=1$, the user has considered H^1-H^0 documents of the first query, H^2-H^1 documents of the second query, H^3-H^2 documents of the third query and so on until the last query, where the number of documents visited is $H-H^{K-1}$. In the previous example, we have $H=9$, $K=3$, $H^1=3$, $H^2=7$, and the user has visited, respectively, 2, 4 and 2 documents in the three queries before stopping the search.

By means of these stopping times, we can define the events corresponding to the end of the search session in any given query. Indeed, if $H^1 = \infty$, it means that the user never passes to the second query and $(F, 1)$ is the unique absorbing state, $A^1 = \{\omega : H^1(\omega) = \infty\}$ corresponds to the event "the user visits just the documents in the first query". Analogously, for any $j > 1$, we can define the event $A^j = \{\omega : H^1(\omega) < \infty, \dots, H^{j-1}(\omega) < \infty, H^j(\omega) = \infty\}$ that the user ends search after considering the first j queries. The events $\{A^j, j \in \mathbb{N}\}$ form a partition of the underlying probability space.

In the following, these events are used to measure how "often" a random user visits each query during her search and to obtain, as a consequence, a weight to be assigned to any query. Moreover, to evaluate the effectiveness of the search within the queries actually visited by the user, we evaluate how far (stochastically) any state is, i.e. any document of any query, from the initial state $(1, 1)$ and discount its relevance proportionally to this "random" distance.

3.2 Evaluation of Multi-query Sessions

As previously discussed, evaluation measures typically apply a deterministic discount of the gain/utility of a document by a function of its rank position. We replace these deterministic discounts operating a two-step stochastic procedure:

- Given that the search generates k queries, we consider the probabilities that the search ends in $(F, 1)$, $(F, 2)$, $\dots$, (F, k), respectively;
- Given that the user does not end her search before the query j, i.e. she visits the documents of the query j, we compute the discount at each rank position of the j-th query according to the expected number of steps needed to reach that rank position starting from $(1, 1)$.

The user can stop her search after considering only the first run, or the first two runs, or the first three runs and so on. This is equivalent to considering that the Markov chain is absorbed in $(F, 1)$, or $(F, 2)$ and so on until (F, k). We are able to evaluate the absorption probabilities in any of these states $h = (h^1, \dots, h^k)$ starting from the probabilities of the events $A^1, \dots, A^{k-1}$ defined above. Indeed, we have $h^j = \mathbb{P}[A^j]$ for any $j < k$, and $h^k = 1 - h^1 - \dots - h^{k-1}$.

Let us define π_j as the probability that the user visits the query j before ending the search

$$\pi_j = \sum_{l=j}^{k} h^l = 1 - \sum_{l=1}^{j-1} h^l \ .$$

To evaluate the "expected distance" from state $(1, 1)$ for the documents in query j, we define the following family of stopping times for any $i \leq N$, since the search does not end before this query j

$$H^{(i,j)} = \inf\{n \geq 1 : X_n = (i, j)\} \, .$$

These stopping times allow us to evaluate how long it takes to reach the document at depth i in query j, and these values are used to perform the average inside the columns.

Thus, given that the search does not end before document (i, j), we define the weight at position (i, j) as

$$e(i, j) = \mathbb{E}_{(1,1)}[H^{(i,j)}] = \mathbb{E}[H^{(i,j)} | X_1 = (1, 1)] \ . \tag{1}$$

To evaluate the contribution of the j-th query to the multi-session search, we compute

$$E(j) = \sum_{i=1}^{N} \phi(e(i, j)) \ GT(d_{i,j}) \tag{2}$$

where $GT(d_{i,j}) \in \mathbb{N}_0$ is the gain corresponding to document $d_{i,j}$ (0 for not relevant documents) and the discount function ϕ is a positive, monotone real function. Choosing it decreasing we discount the relevance of the documents and queries far from the top (primacy according to Liu et al. [18]), while choosing it increasing we give more weight to the relevance of those documents and queries (recency according to Liu et al.). Examples of the function ϕ are: *reciprocal linear weight*, i.e. $\phi(x) = \frac{1}{x}$; *reciprocal logarithmic weight*, i.e. $\phi(x) = \frac{1}{1+\log_{10}(x)}$; and, *logarithmic weight*, i.e. $\phi(x) = 1 + \log_{10}(x)$.

Finally, the new *Markov Session Measure* (MsM) combines the contribution of the k queries in a search session as

$$MsM = \sum_{j=1}^{k} \pi_j E(j) \ . \tag{3}$$

Overall, MsM expresses the expectation of the stochastic time $E(j)$, i.e. number visited documents, it takes for a user to accumulate gain during the search, monotonically transformed by a weighting function ϕ which can put more emphasis either on the start or the end of the search, weighted by the probability π of actually continuing to query.

4 Experimental Setup

In this section, we evaluate the behavior of the proposed measure, answering the following research questions:

RQ1 How does MsM compare to existing session evaluation measures regarding the ranking of retrieval systems?

RQ2 Which factors of the user dynamics affect these correlations and to what extent?

Computation of the MsM Measure We developed an efficient way of computing the MsM measure, avoiding the most general and immediate approach of using a large block-diagonal matrix, where each sub-matrix would represent a single query in the session. For space reasons the pseudo-code of the algorithm is omitted here but it is available in the electronic appendix available as Online Resource 1. Moreover, to further ease the reproducibility of experiments, the source code of the actual implementation is available at: `https://github.com/ekanou/Markovian-Session-Measures`.

Data Collection To answer **RQ1** and **RQ2** we ran experiments on the TREC 2015, 2016, and 2017 Dynamic Domain Track (DDT) collection. The search tasks in DDT focus in domains of special interests, which usually produce complex and exploratory searches with multiple rounds of user and search engine interactions. The DDT collection consists of a set of topics, and multi-query sessions corresponding to each topic. In DDT retrieval systems were provided with the first query, they returned a ranked list of 5 documents, and based on passage annotations in these documents, a jig (user simulation) returned a follow-up query. IR systems had the chance to decide when to stop providing users with ranked lists of documents.

Session Evaluation Measures We compare MsM to the normalized [24] versions of session DCG (sDCG) [14], Expected Utility (EU) [28], and Cube Test (CT) [19]. Since we are not dealing with diversity, we simplify them by using a gain function that ignores subtopic relevance.

Model Instantiations. As an exemplification for experimentation purposes, we consider two user's models, with two different set of assumptions. The first model, called **Random-Walk model**, assumes a user who after considering a document she decides, according to constant probabilities, to proceed to the next document (p), to the previous document (q), to stop her search (s), or to reformulate a new query (r). From the transition matrix point of view, this model is determined by the following assumptions:

$$\begin{array}{l}
p_{(i,j),(i+1,j)} = p \ \text{ if } \ i \neq F \\
p_{(i,j),(i-1,j)} = q \ \text{ if } \ i \neq 1 \\
p_{(i,j),(1,j+1)} = r \ \text{ if } \ i \neq F \\
p_{(i,j),(F,j)} = s \ \text{ if } \ i \neq F \\
p_{(F,j),(F,j)} = 1 \ \text{ for any } \ j
\end{array}$$

where $p + q + r + s = 1$ and $p > 0$, $q \geq 0$, $r > 0$ and $s > 0$.

The second one, called **Forward model**, is a special case of the first one, inspired by the RBP philosophy, where the backward probability, q, is set to 0, i.e. it assumes that the user moves only forward in the ranked list.

Experiments To answer **RQ1** we experimented with both the Forward and the Random-Walk model, using reciprocal log and linear weight, introduced earlier, while the probabilities p, r, and s were set to values on a grid in $[0, 1)^3$ with a step of 0.05, under the constraint that $p + r + s = 1$. Given that the results of the Forward and Random-walk model were highly correlated, and due to space limitations, we present results only of the Forward model.

To answer **RQ2** we experiment with both the Forward and Random-Walk model, while we factorize user types by three characteristics: (a) *patience*, in terms of the total number of documents they are willing to examine, (b) *browsing pattern* in terms of whether they prefer to scan the ranked list or reformulate, and

(c) *decisiveness* in terms of deciding whether a document is relevant once they observe it, or moving back to it and re-examining it after they have examined more documents. We control patience by setting the stopping probability, s, to three distinct values, 0.01, 0.1, and 0.3; the first type of user will on average view around 50 documents, the second around 9, and the third around 3, before they quit their search. We control the browsing pattern by setting the probabilities of walking down the ranked list, p, and reformulating, r to a set of values, such that the user either demonstrates a bigger willingness to scan the ranked list, to reformulate, or to have a balanced behaviour. Last, we control decisiveness by either not allowing the user to walk backwards, hence setting the backwards probability, q, to 0, or allowing the user to do so, by setting q to 0.1. [1]

5 Results and Analysis

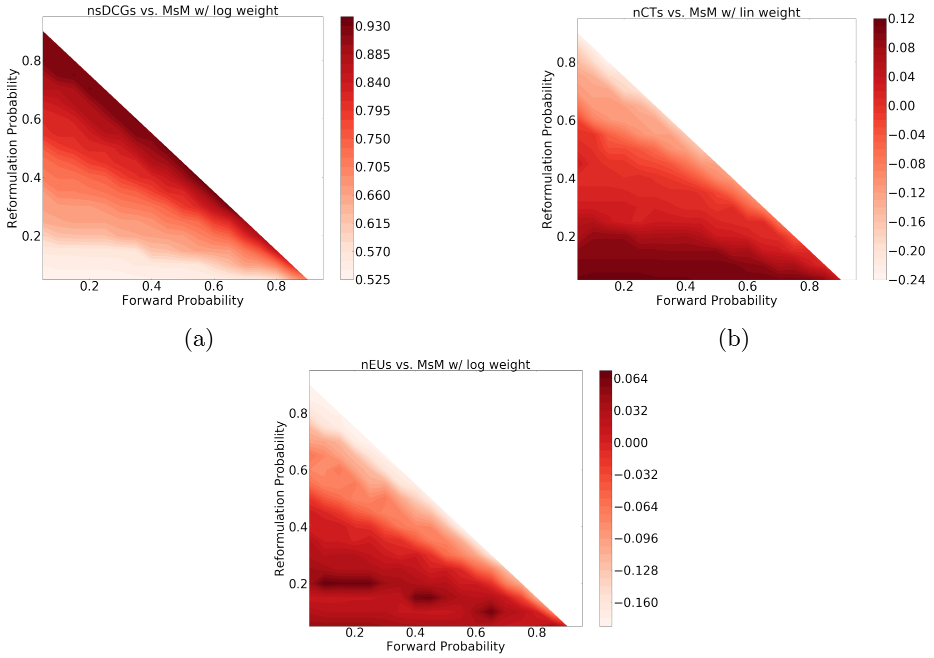
5.1 RQ1 – Correlation Analysis

To answer **RQ1**, we conduct a correlation analysis using Kendall's τ [16] among the rankings of systems produced by the different evaluation measures. Ferro [11] has shown that, even if the absolute correlation values are different, removing or not the lower quartile runs produces the same ranking of measures in terms of correlation; similarly, it was shown that both τ and AP correlation τ_{ap} [29] produce the same ranking of measures in terms of correlation. Therefore, we focus only on Kendall's τ without removing lower quartile systems.

Figure 1 presents the average Kendall's τ correlation between different instantiations of MsM measure and sDCG, EU and CT, respectively, on rankings of systems in DDT. The x-axis in all three plots corresponds to the forward probability, p, while the y-axis to the reformulation probability, r. The stopping probability, s, can be infered, given that $p + r + s = 1$. The colorbar shows the actual correlation values.

Figure 1a corresponds to the correlation with sDCG. It can be observed that the highest correlation is achieved along the secondary diagonal, i.e. when the stopping probability is 0.05, with the maximum value obtained when p is 0.55, r is 0.40 and s is 0.05. This shows that the browsing model of sDCG penalizes documents both lower in the ranking and further in the session queries. Figure 1b corresponds to the correlation with CT. It can be observed that the highest correlation is achieved along the x-axis, i.e. when the reformulation probability is 0.05, with the maximum value obtained when p is 0.10, r is 0.05, and s is 0.85. As a reminder CT does not penalize documents that appear lower in the ranking; it only penalizes documents that appear further in the session, with a reciprocal linear weight of the index of the query in the session. This is captured by the plot: the high forward probability essentially dictates little penalization within a ranking, while the low reformulation probability dictates a high penalization across queries in the session. The overall low correlation

[1] Advanced user dynamics that condition probabilities on the relevance of the viewed document, similar to ERR, are also possible with MsM but are left as future work.



(a)

(b)

(c)

Fig. 1: Average Kendall's τ correlation.

(0.12) however also designates that the penalization model of CT can be hardly modeled in a probabilistic manner. Figure 1c corresponds to the correlation with EU. The highest correlation is achieved when p is 0.10, r is 0.20 and s is 0.70. The plot demonstrates a pattern of high correlations that is in between the high correlation patterns of sDCG and CT. The high correlation at low reformulation probabilities also shows that EU expects a user to move forward a ranked list and reformulate only at the end of it. In conclusion, to some extent, the MsM measure provides some insights on the implicit user models of existing measures, even if some of the assumptions made in those measures do not always allow high correlation scores.

5.2 RQ2 – Analysis of Variance

To answer **RQ2**, we conduct an Analysis of Variance (ANOVA) of the different factors that may influence the correlation between MsM and existing session evaluation measures. For space reasons, we report this analysis in the case of sDCG and EU, being possible to draw similar conclusions also in the case of CT.

Table 1 shows the three-way ANOVA for analysing the factors in the MsM measures which influence the correlation with nsDCGs. The Track factor rep-

Table 1: Analysis of the factors influencing correlation with sDCG.

Source	**SS**	**DF**	**MS**	**F**	**p**	$\hat{\omega}^2_{\langle fact \rangle}$
Track	0.1526	2	0.0763	5.3294	0.0083	0.1382
Patience	0.2712	2	0.1356	9.4704	0.0003	0.2388
Browsing	0.4330	2	0.2165	15.1245	$<$ e-4	0.3435
Weight	0.0014	1	0.0014	0.1004	0.7528	–
Error	0.6582	46	0.0143			
Total	1.5167	53				

Table 2: Analysis of the factors influencing correlation with EU.

Source	**SS**	**DF**	**MS**	**F**	**p**	$\hat{\omega}^2_{\langle fact \rangle}$
Track	0.0084	2	0.0042	0.3962	0.6752	–
Patience	0.0255	2	0.0127	1.2022	0.3098	–
Browsing	0.9185	2	0.4593	43.3024	$<$ e-4	0.6104
Weight	0.0057	1	0.0057	0.5333	0.4689	–
Error	0.4879	46	0.0106			
Total	1.4459	53				

resents the effect of one of the three tracks (DD 2015, 2016, and 2017); the Patience factor represents the effect of the patience of the user in scanning the list (impatient, balanced, patient); the Browsing factor represents the attitude to walk down the list or reformulate new queries (down, balanced, reformulate); the Weight factor represents the type of discount (linear, log). The ANOVA analysis shows that the Track, Patience, and Browsing factors are statistically significant while the Weight one is not; we also conducted an ANOVA analysis (not reported here for space reason) to test the interaction among these factors but none of them is significant. The Tukey Honestly Significant Difference (HSD) test shows that the `impatient` user is significantly different from the `balanced` and `patient` ones, which are not significantly different from each other, being the `impatient` user the lowest one in terms of correlation and the `patient` the highest one. The Tukey HSD test also shows that the `balanced` browsing pattern is significantly different from the `down` and `reformulate` ones, which are not significantly different from each other, being the `reformulate` strategy the lowest one in terms of correlation and the `balanced` the highest one. The Strength of Association (SOA) ω^2 shows that the Track factor is a medium-size effect while the Patience and Browsing factors are large-size effects, being the browsing pattern the most prominent one. Overall, this analysis suggests that the most prominent motivations of similarity between MsM measures and sDCG are a balanced browsing pattern and a balanced/patient user, which is the user model actually implemented in sDCG.

Table 2 shows the three-way ANOVA for analysing the factors in the MsM measures which influence the correlation with EU. The ANOVA analysis shows that only the Browsing factor is statistically significant while all the others are not; we also conducted an ANOVA analysis (not reported here for space reason) to test the interaction among these factors but none of them is significant. The Tukey HSD test shows that all the browsing patterns are significantly different, being the `balanced` strategy the lowest one in terms of correlation and the `down` the highest one. The SOA ω^2 shows that the Browsing factor is a large-size effect. Overall, this analysis suggests that the most prominent motivation of similarity between MsM measures and EU is a down browsing pattern.

Overall, the ANOVA analysis also highlights that the Track factor, when significant, is not the most influencing one and this supports the previous observation about a consistent behaviour of the measures across the tracks and reporting the correlation values averages across tracks.

6 Conclusions and Future Work

We considered the problem of evaluating multi-query sessions. Differently from past attempts we provided a mathematical formulation of the user dynamics on the basis of a Markov chain that allows for a strong theoretical underpinning of the deduced measure. The measure proposed provides a flexible but at the same time mathematically sound and intuitive parametrization on the basis of the expected user behavior. We experimented with different variations of the measure each making its own assumption regarding (a) the chance of the user to return to an already seen document in a ranked list; (b) the patience of the user to move down in a ranked list as opposed to reformulating her query; and, (c) the patience of the user in the overall use of the information retrieval system. We showed that the produced measures can indeed capture different user behaviors, and through a correlation analysis we attempted to provide a better understanding of existing session measures and the implicit assumptions in their user models.

What we present in this work is a rather flexible framework to construct session evaluation measures of interest. A number of future directions could be explored: (a) identifying the right parameters that will reduce the proposed MsM to existing session measures, providing a theoretical underpinning of those measures and better expandability; (b) injecting more advanced user dynamics in the MsM by e.g. modeling transition probabilities as conditional probabilities on the relevance of the visited documents; (c) learning parameters using query logs or leveraging user studies; and, (d) expanding the discrete Markov chain to a continuous-time Markov chain to naturally incorporate time in the measure.

Acknowledgements. This research was supported by the NWO Innovational Research Incentives Scheme Vidi (016.Vidi.189.039). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

1. Allan, J., Arguello, J., Azzopardi, L., Bailey, P., Baldwin, T., Balog, K., Bast, H., Belkin, N., Berberich, K., von Billerbeck, B., Callan, J., Capra, R., Carman, M., Carterette, B., Clarke, C.L.A., Collins-Thompson, K., Craswell, N., Croft, W.B., Culpepper, J.S., Dalton, J., Demartini, G., Diaz, F., Dietz, L., Dumais, S., Eickhoff, C., Ferro, N., Fuhr, N., Geva, S., Hauff, C., Hawking, D., Joho, H., Jones, G.J.F., Kamps, J., Kando, N., Kelly, D., Kim, J., Kiseleva, J., Liu, Y., Lu, X., Mizzaro, S., Moffat, A., Nie, J.Y., Olteanu, A., Ounis, I., Radlinski, F., de Rijke, M., Sanderson, M., Scholer, F., Sitbon, L., Smucker, M.D., Soboroff, I., Spina, D., Suel, T., Thom, J., Thomas, P., Trotman, A., Voorhees, E.M., de Vries, A.P., Yilmaz, E., Zuccon, G.: Research Frontiers in Information Retrieval – Report from the Third Strategic Workshop on Information Retrieval in Lorne (SWIRL 2018). SIGIR Forum 52(1), 34–90 (June 2018)
2. Amigó, E., Fang, H., Mizzaro, S., Zhai, C.: Report on the SIGIR 2017 Workshop on Axiomatic Thinking for Information Retrieval and Related Tasks (ATIR). SIGIR Forum 51(3), 99–106 (December 2017)
3. Busin, L., Mizzaro, S.: Axiometrics: An Axiomatic Approach to Information Retrieval Effectiveness Metrics. In: Kurland, O., Metzler, D., Lioma, C., Larsen, B., Ingwersen, P. (eds.) Proc. 4th International Conference on the Theory of Information Retrieval (ICTIR 2013). pp. 22–29. ACM Press, New York, USA (2013)
4. Carterette, B.: System effectiveness, user models, and user utility: A conceptual framework for investigation. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 903–912. SIGIR '11, ACM, New York, NY, USA (2011), `http://doi.acm.org/10.1145/2009916.2010037`
5. Chapelle, O., Metzler, D., Zhang, Y., Grinspan, P.: Expected Reciprocal Rank for Graded Relevance. In: Cheung, D.W.L., Song, I.Y., Chu, W.W., Hu, X., Lin, J.J. (eds.) Proc. 18th International Conference on Information and Knowledge Management (CIKM 2009). pp. 621–630. ACM Press, New York, USA (2009)
6. Chierichetti, F., Kumar, R., Raghavan, P.: Optimizing Two-Dimensional Search Results Presentation. In: King, I., Nejdl, W., Li, H. (eds.) Proc. 4th ACM International Conference on Web Searching and Data Mining (WSDM 2011). pp. 257–266. ACM Press, New York, USA (2011)
7. Collins-Thompson, K., Callan, J.: Query Expansion Using Random Walk Models. In: Herzog, O., Schek, H.J., Fuhr, N., Chowdhury, A., Teiken, W. (eds.) Proc. 14th International Conference on Information and Knowledge Management (CIKM 2005). pp. 704–711. ACM Press, New York, USA (2005)
8. Daniłowicz, C., Baliński, J.: Document ranking based upon Markov chains. Information Processing & Management 37(4), 623—637 (July 2001)
9. Ferrante, M., Ferro, N., Pontarollo, S.: A General Theory of IR Evaluation Measures. IEEE Transactions on Knowledge and Data Engineering (TKDE) (2018)
10. Ferrante, M., Ferro, N., Maistro, M.: Injecting user models and time into precision via markov chains. In: Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 597–606. SIGIR '14, ACM, New York, NY, USA (2014), `http://doi.acm.org/10.1145/2600428.2609637`
11. Ferro, N.: What Does Affect the Correlation Among Evaluation Measures? ACM Transactions on Information Systems (TOIS) 36(2), 19:1–19:40 (September 2017)

12. Fuhr, N.: Salton Award Lecture: Information Retrieval As Engineering Science. SIGIR Forum 46(2), 19–28 (December 2012)
13. Järvelin, K., Kekäläinen, J.: Cumulated Gain-Based Evaluation of IR Techniques. ACM Transactions on Information Systems (TOIS) 20(4), 422–446 (October 2002)
14. Järvelin, K., Price, S.L., Delcambre, L.M.L., Nielsen, M.L.: Discounted cumulated gain based evaluation of multiple-query ir sessions. In: Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval. pp. 4–15. ECIR'08, Springer-Verlag, Berlin, Heidelberg (2008), http://dl.acm.org/citation.cfm?id=1793274.1793280
15. Kanoulas, E., Carterette, B., Clough, P.D., Sanderson, M.: Evaluating multi-query sessions. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1053–1062. SIGIR '11, ACM, New York, NY, USA (2011), http://doi.acm.org/10.1145/2009916.2010056
16. Kendall, M.G.: Rank correlation methods. Griffin, Oxford, England (1948)
17. Lafferty, J., Zhai, C.: Document Language Models, Query Models, and Risk Minimization for Information Retrieval. In: Kraft, D.H., Croft, W.B., Harper, D.J., Zobel, J. (eds.) Proc. 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001). pp. 111–119. ACM Press, New York, USA (2001)
18. Liu, M., Liu, Y., Mao, J., Luo, C., Ma, S.: Towards Designing Better Session Search Evaluation Metrics. In: Collins-Thompson, K., Mei, Q., Davison, B., Liu, Y., Yilmaz, E. (eds.) Proc. 41th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018). pp. 1121–1124. ACM Press, New York, USA (2018)
19. Luo, J., Wing, C., Yang, H., Hearst, M.: The water filling model and the cube test: Multi-dimensional evaluation for professional search. In: Proceedings of the 22Nd ACM International Conference on Information and Knowledge Management. pp. 709–714. CIKM '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2505515.2523648
20. Maxwell, K.T., Croft, W.B.: Compact Query Term Selection Using Topically Related Text. In: Jones, G.J.F., Sheridan, P., Kelly, D., de Rijke, M., Sakai, T. (eds.) Proc. 36th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2013). pp. 583–592. ACM Press, New York, USA (2013)
21. Moffat, A., Zobel, J.: Rank-biased precision for measurement of retrieval effectiveness. ACM Trans. Inf. Syst. 27(1), 2:1–2:27 (Dec 2008), http://doi.acm.org/10.1145/1416950.1416952
22. Smucker, M.D., Clarke, C.L.A.: Stochastic simulation of time-biased gain. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management. pp. 2040–2044. CIKM '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2396761.2398568
23. Smucker, M.D., Clarke, C.L.: Time-based calibration of effectiveness measures. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 95–104. SIGIR '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2348283.2348300
24. Tang, Z., Yang, G.H.: Investigating per topic upper bound for session search evaluation. In: Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval. pp. 185–192. ICTIR '17, ACM, New York, NY, USA (2017), http://doi.acm.org/10.1145/3121050.3121069

25. Yan, X., Gao, G., Su, X., Wei, H., Zhang, X., Lu, Q.: Hidden Markov Model for Term Weighting in Verbose Queries. In: Catarci, T., Forner, P., Hiemstra, D., Peñas, A., Santucci, G. (eds.) Information Access Evaluation. Multilinguality, Multimodality, and Visual Analytics. Proceedings of the Third International Conference of the CLEF Initiative (CLEF 2012). pp. 82–87. Lecture Notes in Computer Science (LNCS) 7488, Springer, Heidelberg, Germany (2012)
26. Yang, G.H., Soboroff, I.: TREC 2015 dynamic domain track overview. In: Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 15-18, 2016 (2015)
27. Yang, G.H., Soboroff, I.: TREC 2016 dynamic domain track overview. In: Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016 (2016)
28. Yang, Y., Lad, A.: Modeling expected utility of multi-session information distillation. In: Proceedings of the 2Nd International Conference on Theory of Information Retrieval: Advances in Information Retrieval Theory. pp. 164–175. ICTIR '09, Springer-Verlag, Berlin, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-04417-5_15
29. Yilmaz, E., Aslam, J.A., Robertson, S.E.: A New Rank Correlation Coefficient for Information Retrieval. In: Chua, T.S., Leong, M.K., Oard, D.W., Sebastiani, F. (eds.) Proc. 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008). pp. 587–594. ACM Press, New York, USA (2008)

A Markovian Approach to Evaluate Session-based IR Systems

Electronic Appendix

David van Dijk[1], Marco Ferrante[2][0000−0002−0894−4175], Nicola Ferro[2][0000−0001−9219−6239], and Evangelos Kanoulas[1][0000−0002−8312−0694]

University of Amsterdam, The Netherlands {d.v.van.dijk@hva.nl, e.kanoulas@uva.nl}
University of Padua, Italy {ferrante, ferro}@dei.unipd.it

A Pseudocode for the Computation of the MsM Measure

Algorithm 1 stems from the observation that, since the random variables in eq. (1) can take the value ∞ with positive probability, their expectation is equal to ∞. Nevertheless, in order to measure the "stochastic distance" of the document at position (i, j) from that in $(1, 1)$, we can compute the expectation under the restriction that the search does not end before reaching the target document. In practice, we can equal to 0 the probability to reach the state F from the states above (i, j) and normalize the other elements to still have a stochastic matrix. Therefore, in the algorithm, we can use a single smaller transition matrix P, representing the dynamic within a generic query, with $N + 2$ states: the first N states represents the N retrieved documents within the query, the $F = N + 1$ state is the end of the search session, and the $Q = N + 2$ is the reformulation to the next query. We then, appropriately, select rows and columns from this transition matrix P, rescale them into a $\hat{P}$ matrix ensuring it is stochastic, and perform the different computations, as discussed in the pseudo-code.

Algorithm 1: Pseudo-code for computing the *MsM* measure (It continues on the next page.)

```
Input: run a $N \times K$ integer matrix where rows are documents, columns are
       queries, and each cell contains a natural number representing the
       relevance of a document, 0 for not relevant. $i$ is the document index, $j$
       is the query index.
Input: $p$, $q$, $r$, and $s$, the transition probabilities of, respectively: moving
       forward to the next document within a query; moving backward to the
       previous document within a query; moving to the next query, i.e.
       jumping to the first document of the next query; and, stopping a
       search session. We assume $p + q + r + s = 1$, $p > 0$, $q \geq 0$, $r > 0$, and
       $s > 0$.
Output: the MsM measure for the input run.

Data: For the first document of a generic query, there is no backward
      transition probability, so we have to rescale the transition probabilities:
      $p_1 = \frac{p}{p+s+r}$, $s_1 = \frac{s}{p+s+r}$, $r_1 = \frac{r}{p+s+r}$
Data: For the last document of a generic query, there is no forward transition
      probability, so we have to rescale the transition probabilities:
      $q_N = \frac{q}{q+s+r}$, $s_N = \frac{s}{q+s+r}$, $r_N = \frac{r}{q+s+r}$
Data: The transition matrix $P$ of a generic query is a $(N+2) \times (N+2)$
      stochastic matrix where:
  – the leading $N \times N$ principal sub-matrix contains the backward and forward
    transitions between the documents within a query, paying attention to the
    re-scaled transition probabilities for the first and last document;
  – the additional absorbing state $F = N + 1$ represents the end of a search session;
  – the additional absorbing state $Q = N + 2$ represents the jump to the next query.
```

$$P = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ N-1 \\ N \\ F \\ Q \end{array} \begin{array}{c} \begin{array}{cccccccc|cc} 1 & 2 & 3 & 0 & \cdots & N-2 & N-1 & N & F & Q \end{array} \\ \left(\begin{array}{cccccccc|cc} 0 & p_1 & 0 & 0 & \cdots & 0 & 0 & 0 & s_1 & r_1 \\ q & 0 & p & 0 & \cdots & 0 & 0 & 0 & s & r \\ 0 & q & 0 & p & \cdots & 0 & 0 & 0 & s & r \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & q & 0 & p & s & r \\ 0 & 0 & 0 & 0 & \cdots & 0 & q_N & 0 & s_N & r_N \\ \hline 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 1 \end{array}\right) \end{array}$$

```
Data: $e$ is a $N \times 1$ vector, representing the average time to go from document
      1 to documents $1, \ldots, N$, respectively.
Data: $e_Q$ is a scalar, representing the average time to move to the next query.
Data: $h_F$ is a scalar, representing the probability to go from document 1 to
      state $F$, i.e. end of search session, in the first $K - 1$ queries.
Data: $h_{FK}$ is a scalar, representing the probability to go from document 1 to
      state $F$, i.e. end of search session, in the $K$-th query.
Data: $h$ is a $1 \times K$ vector, representing the probability to go from document 1
      of the first query to the state $(F, 1), \ldots, (F, K)$, respectively.
Data: $\pi$ is a $1 \times K$ vector, representing the probability of ending the search
      session in query $1, \ldots, K$, respectively.
Data: $g$ is a $1 \times K$ vector, representing the total gain of each query.

/* the average time to go from document 1 to itself is 0          */
$e(1) \leftarrow 0$;
```

Algorithm 1: Pseudo-code for computing the MsM measure. (Continued from the previous page and it continues on the next page.)

```
/* $\hat{P}_1$ is the same matrix as $P$ but where the $F$ and $Q$ state rows
   and columns have been removed since, to compute the average time
   to move from document 1 to document $i$, you must assume that
   neither the search session has ended nor you have moved to the
   next query                                                        */
$\hat{P}_1 \leftarrow P(1:N, 1:N)$;
ensure $\hat{P}_1$ is a stochastic matrix, i.e. normalize each row so that it sums up to
  1;
for $i \leftarrow 1$ to $N-1$ do
    /* compute the average time to go from each document to document
       $i+1$ ($\mathbf{I}_i$ is the $i \times i$ identity matrix)    */
    $tmp \leftarrow \Big(\mathbf{I}_i - \hat{P}_1(1:i, 1:i)\Big)^{-1} \begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}$;
    /* we are interested only in the average time to go from
       document 1 to document $i+1$                                  */
    $e(i+1) \leftarrow tmp(1)$;
end
/* $\hat{P}_2$ is the same matrix as $P$ but where the $F$ state rows and
   columns have been removed since, to compute the average time to
   move to the next query, you must assume that the search session
   has not ended                                                     */
$\hat{P}_2 \leftarrow P([1:N\ Q], [1:N\ Q])$;
ensure $\hat{P}_2$ is a stochastic matrix, i.e. normalize each row so that it sums up to
  1;
/* compute the average time to go from each document to the next
   query                                                             */
$tmp \leftarrow \Big(\mathbf{I}_N - \hat{P}_2(1:N, 1:N)\Big)^{-1} \begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}$;
/* we are interested only in the average time to go from document 1
   to the next query                                                 */
$e_Q \leftarrow tmp(1)$;
/* compute the probability to go from document $i$ to state $F$ in the
   first $K-1$ queries                                               */
$tmp \leftarrow \Big(\mathbf{I}_N - P(1:N, 1:N)\Big)^{-1} P(1:N, F)$;
/* we are interested only in the probability to go from document 1
   to state $F$ in the first $K-1$ queries                           */
$h_F \leftarrow tmp(1)$;
/* $\hat{P}_3$ is the same matrix as $P$ but where the $Q$ state rows and
   columns have been removed since, to compute the probability to
   go from document $i$ to state $F$ in the $K$-th query, there is not
   next query                                                        */
$\hat{P}_3 \leftarrow P(1:F, 1:F)$;
ensure $\hat{P}_3$ is a stochastic matrix, i.e. normalize each row so that it sums up to
  1;
```

Algorithm 1: Pseudo-code for computing the MsM measure. (Continued from the previous page, last page.)

```
/* compute the probability to go from document $i$ to state $F$ in the
   $K$-th query                                                        */
$tmp \leftarrow \left(\mathbf{I}_N - \hat{P}_3(1:N, 1:N)\right)^{-1} \hat{P}_3(1:N, F)$;
/* we are interested only in the probability to go from document $1$
   to state $F$ in the $K$-th query                                    */
$h_{FK} \leftarrow tmp(1)$;

/* compute the probability to go from document $1$ (of the first
   query) to state $F$ of the $j$-th query                             */
for $j \leftarrow 1$ to $K - 1$ do
    $h(j) \leftarrow (1 - h_F)^{j-1} \cdot h_F$;
end
$h(K) \leftarrow (1 - h_F)^{K-1} \cdot h_{FK}$;

/* compute the probability of ending the search session in query $j$
   */
$\pi(1) \leftarrow 1$;
for $j \leftarrow 2$ to $K$ do
    $\pi(j) \leftarrow 1 - \sum_{l=1}^{j-1} h(l)$;
end

/* discount each retrieved document by the average time to get to
   it                                                                  */
for $i \leftarrow 1$ to $N$ do
    for $j \leftarrow 1$ to $K$ do
        $E(i, j) \leftarrow \frac{run(i,j)}{1+e(i)+e_Q}$;

        /* In case of logarithmic discount, use instead
           $E(i, j) \leftarrow \frac{run(i,j)}{1+\log_{10}(1+e(i)+e_Q)}$;

                                                                       */
    end
end

/* compute the total gain for each query                               */
for $j \leftarrow 1$ to $K$ do
    $g(j) \leftarrow \pi(j) \sum_{i=1}^{N} E(i)$;
end

/* compute the $MsM$ measure                                           */
$MsM \leftarrow \sum_{j=1}^{K} g(j)$;
```