# Endicott College at 2014 TREC Session Track

Henry Feild
Endicott College
Beverly, MA 01915
hfeild@endicott.edu

## 1 Overview

Endicott College submitted three runs to Task 1 of the 2014 TREC Session Track. All runs re-ranked the baseline runs provided by the track organizers. One of the runs made use of a click graph to re-rank results for RL1, RL2, and RL3. The other two used relevance models computed over snippets from the session, and boosted their RL3 run using click graph recommendations. In the absence of clicks (e.g., RL1 and clickless sessions in RL2 and RL3), two of the runs used pseudo relevance feedback over the session, while the other used the unmodified baseline ranking.

All runs used a similar pre-processing procedure, which we describe in Section 2. We then discuss our click graph re-ranking technique for the `ECxCGxPRF` run in Section 3 and the session relevance modeling technique for our `ECxSRMxOS` and `ECxSRMxPRF` runs in Section 4. We follow that with an analysis of the performance of our runs compared to each other, as well as the track minimums, medians, and maximums. Finally, we wrap up with some closing thoughts and future directions in Section 6.

## 2 Preprocessing

While the interaction history for each session includes the snippets for all displayed documents for queries other than `currentquery`, the baseline runs for `currentquery` provided by the Session Track coordinators were in the TREC run format and did not include snippets. Since the snippet relevance model re-ranking technique used in two of our runs requires snippets, we had to generate these.

To generate snippets, we used an index we created earlier consisting of the content plus the inlink, URL, and title fields of all non-spam documents in the ClueWeb2012 Category A collection. We split the corpus into 10 shards. We used Indri version 5.7 with Krovetz stemming and a short stoplist of 12 words: *a, at, i, you, we, it, the, of, he, she, there,* and *them.* Documents were considered spam if they were marked with a Waterloo spam score lower than 65 [1].[1] This resulted in 256,555,383 documents being white listed as non- spam. The final index consisted of 256,554,159 documents (we did not investigate what caused the 1,224 document discrepancy). As we planned to generate snippets, we included the collection and the core index. The index portion took up 842 GB of disk space, and collection 2.5 TB. The total combined took 3.3 TB of disk space. Each shard took between 22–37 hours to process with up to five shards being processed in parallel on a single machine with two 8-core processors (2GHz/core) and 64 GB of RAM with Ubuntu 12.04 server edition.

[1] http://www.mansci.uwaterloo.ca/ ms mucker/cw12spam/

For each baseline run, we removed all spam documents (i.e., documents not in our index), after which we kept only the top 100 results (or fewer if the list size was less than 100) and generated the snippets. We generated snippets by submitting the `currentquery` text to Indri and using the `workingSetDocno` parameter to specify only the results we kept from the previous step.

# 3 Click Graph

Craswell and Szummer [2] demonstrated the effectiveness of random walks on click graphs to improve ranking. For our `ECxCGxPRF` runs, as well as the RL3 runs for both `ECxSRMxOS` and `ECxSRMxPRF`, we relied on a simple click graph to provide document suggestions, and then used these to re-rank the top 50 documents in a run. A description of the graph generation process is below, followed by the particulars for different runs, including what was used to define "clicked" documents.

## 3.1 Generating a click graphs

The click graphs used in this work were generated using the following procedure. First, we generate a directed graph $G = \{V, E\}$, where $V$ is the union of the identifiers of all clicked documents in the input data set (in our case, the TREC Session data from 2013, 2014, or both) and $E$ is a set of weighted edge tuples: $(v_i, v_j, w_{ij})$. Weights are calculated in three stages as follows. For each session, consider the set of clicked documents $\{d_1, d_2, \ldots, d_n\}$, and their click frequencies $\{f_1, f_2, \ldots, f_n\}$ (in many sessions, certain documents are clicked several times). In stage 1 of the weight calculation, we generate the weight for each of the $n^2$ pairs of documents in each session. For any pair $(d_i, d_j) : i \neq j$, we create the weighted edges $(d_i, d_j, f_i \times f_j)$ and $(d_j, d_i, f_i \times f_j)$. This means that if document $d_i$ is clicked five times, $d_j$ is clicked once, and $d_k$ is clicked once, then the weight between $d_i$ and either of $d_j$ or $d_k$ will be five times stronger than between $d_j$ and $d_k$.

In stage 2, weights are summed across weighted edges between the same two documents, which generates an unnormalized set of edges of the form $(v_i, v_j, \hat{w}_{ij})$ where there is exactly one weight corresponding to each pair $(v_i, v_j)$.

In stage 3, we normalize the weights of all outgoing edges of each vertex such that they sum to one, giving us the final set of edges for $G$: $E = \{\ldots, (v_i, v_j, w_{ij}), \ldots\}$. Note that for a pair of documents $v_i$ and $v_j$, the edges $(v_i, v_j, w_{ij})$ and $(v_j, v_i, w_{ji})$ are not necessarily weighted the same as they were normalized independently.

## 3.2 Generating suggestions from a click graph

To generate suggestions from a click graph, we used a random walk with restarts [3]. The algorithm takes an initialization vector $\mathbf{u}$, which consists of starting weights for each vertex in the graph. The following formula is used to compute the suggestions:

$$\mathbf{v}_{t+1} = (1 - c)A\mathbf{v}_t + c\mathbf{u},$$

where $c$ is the restart factor, $\mathbf{v}_t$ is a vector containing the probabilities of being at each vertex in the graph after $t$ iterations, and $A$ is the transition matrix between vertices in the graph. Initially, $\mathbf{v}_0$ is set to $\mathbf{u}$. We continue to compute $\mathbf{v}_{t+1}$ for increasing values of $t$ until either it has converged, meaning the $\mathrm{L}_1$ distance between two iterations is less then some value $\epsilon$, i.e., $||\mathbf{v}_{t+1} - \mathbf{v}_t||_2 < \epsilon$, or $\rho$ iterations have been completed.

| Run ID | RL | Graph source | 'Click' definition |
|---|---|---|---|
| ECxCGxPRF | 1 | 2013 | Top document retrieved for currentquery. |
| ECxCGxPRF | 2 | 2013 | Real clicks or top document retrieved for each query for clickless sessions. |
| ECx* | 3 | 2013+2014 | Real clicks or top document retrieved for each query for clickless sessions. |

Table 1: For each run that used click graphs in some capacity, this table lists the data source used to generate the graph (either the 2013 session data, or the 2013+2014 session data), and what constituted a click when generating the initialization vector for the suggestion algorithm.

For all of our experiments, we used a restart factor of $c = 0.2$, a convergence distance of $\epsilon = 0.005$, and a max iteration cap of $\rho = 1000$. We used a custom implementation of the algorithm, available on GitHub.[2]

For use with the graphs described above, the weights in the initialization vector $\mathbf{u}$ are the click frequencies of the corresponding documents in the session under examination, that is, $\mathbf{u}[d_i] = f_i$.

### 3.3 Defining clicks

The algorithm to generate suggestions requires some definition of what it means for a document to be clicked. In RL1 runs, for example, we have no click information—only the currentquery is known. As listed in Table 1, we simulated clicks for sessions in the ECxCGxPRF RL1 runs as the top retrieved document from the baseline for the currentquery. For RL2 and RL3, we used real clicks in sessions with at least one click, and simulated clicks as the top retrieved baseline result for each query in the session for clickless sessions. The 'PRF' (pseudo relevance feedback) in the run tag denotes our use of top results as click surrogates.[3]

To be clear, we used only real clicks from the 2012 and 2013 datasets to generate the click graphs. Click surrogates were only used when creating the initialization vector $\mathbf{u}$ in the suggestion generation phase (Section 3.2).

## 4 Session Relevance Modeling

Jiang and He [4] demonstrated that using clicked summaries as explicit relevance feedback is a useful tool in session-based retrieval. Motivated by their work, two of our runs, ECxSRMxOS and ECxSRMxPRF, use session relevance modeling (SRM) to re-rank the baseline results. SRM considers a set of clicked summaries, $S$, from a session. These summaries are then used in the same way as the Relevance Model Method 1 proposed by Lavrenko and Croft [5], except we substitute clicked summaries in place of the top $k$ documents as the relevant set and we retrieve over result snippets in the top 100 baseline results.

All snippets in $S$ and the baseline result snippets were stemmed using the Porter stemmer and stopped with a 119-term stoplist.[4] We used Dirichlet smoothing with $\mu = 50$. Out of vocabulary words were given a default term frequency of 0.00000005.

[2] https://github.com/hafeild/term-query-graph

[3] This naming convention was violated for ECxSRMxOS RL3, in which the associated RL2 run was boosted with PRF suggestions as used in the other RL3 runs.

[4] http://www.textfixer.com/resources/common-english-words.txt

| | Mean<br>nDCG@10 | Queries<br>w/ nDCG@10=1 | Queries<br>w/ nDCG@10=0 |
|---|---|---|---|
| ECxCGxPRF R1 | .181 | 0 | 23 |
| R2 | .182 | 0 | 23 |
| R3 | .193 | 0 | 21 |
| ECxSRMxPRF R1 | .168 | 1 | 34 |
| R2 | .178 | 3 | 35 |
| R3 | .194 | 3 | 33 |
| ECxSRMxOS R1 | .179 | 0 | 23 |
| R2 | .195 | 3 | 26 |
| R3 | .205 | 3 | 24 |

Table 2: A few performance stats across the 100 judged sessions.

As with the click graph, we used multiple definitions of a click. For ECxSRMxOS RL1, we used the baseline ('OS' is short for 'original SERP', i.e., the baseline), post spam removal. For the ECxSRMxOS RL2, we used real clicks in sessions with at least one click, and the baseline in clickless sessions. RL3 was the RL2 run, boosted by click graph suggestions (which used a click surrogate for clickless sessions, as describe in Section 3.3).

In ECxSRMxPRF RL1, we used the top result from the baseline for currentquery in each session as a surrogate for a click. For the ECxSRMxPRF RL2 runs, we used real clicks when possible, and the top results retrieved per query in the session for clickless sessions. The RL3 run was the result of boosting the RL2 run with click graph suggestions.

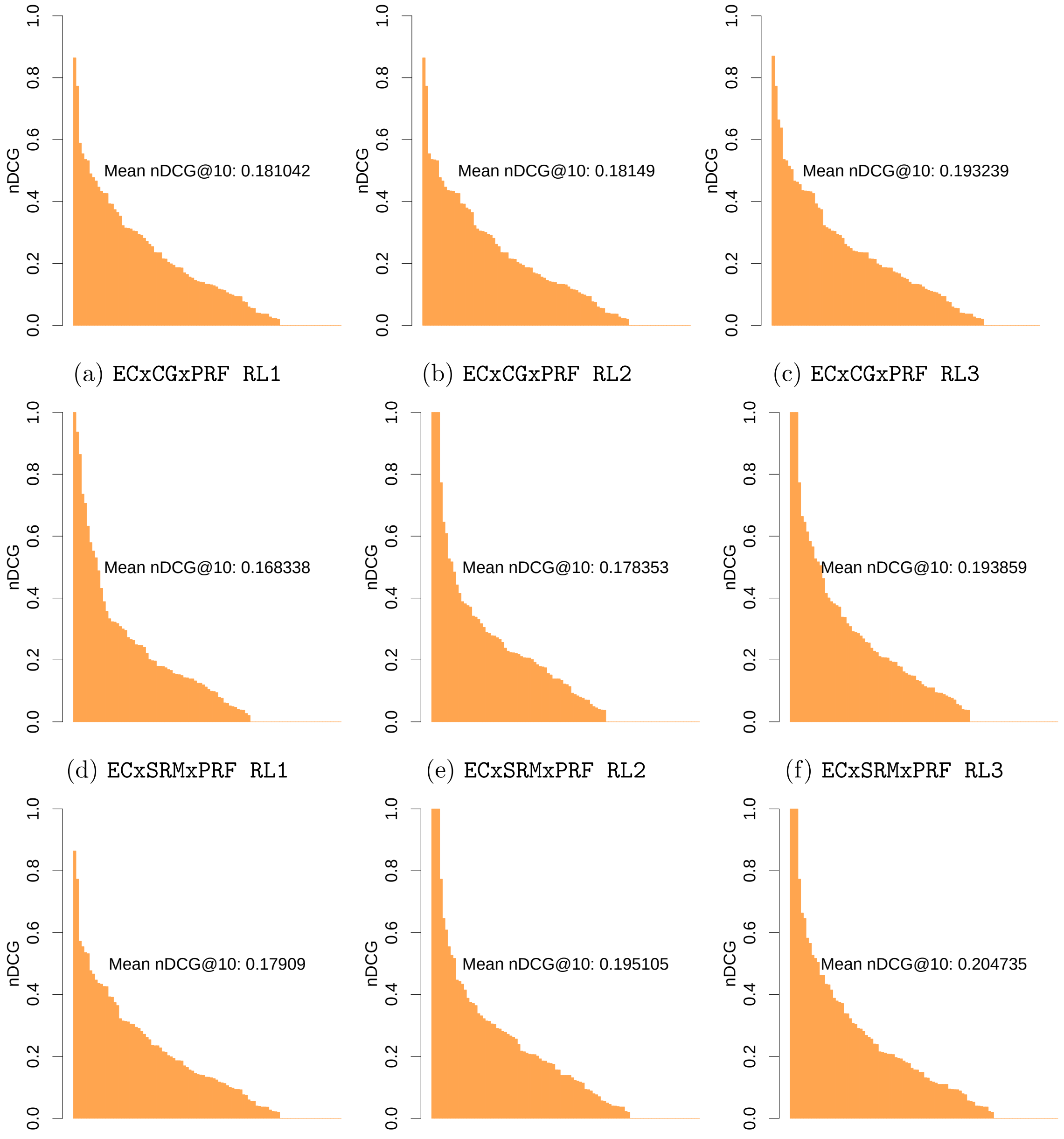
## 5 Performance Analysis

The results for the 2014 TREC Session Track are based on judgments made for the top ten documents retrieved for the first 100 sessions from each participant's top three priority runs. The official measure of the track is nDCG@10, which is the only measure we report on here. When we refer to sessions below, we specifically mean the currentquery of each session.

All three systems improved mean nDCG@10 from RL1 to RL2, and again from RL2 to RL3. Figure 1 shows the per session and mean nDCG@10 for all three run levels for each run group. Of note, the ECxCGxPRF runs (top row) never achieved an nDCG@10 above 0.88 for any session in any run level. ECxSRMxPRF was the only run group to achieve nDCG@10=1 for at least one session across all three run levels. ECxSRMxOS did so for only RL2 and RL3. All run groups saw the highest number of nDCG@10=0 sessions at RL2 and usually the lowest at RL3. Table 2 shows the exact counts, as well as a few other interesting statistics.

ECxSRMxOS RL2 and RL3 are the two best performing runs: highest mean nDCG@10, tied for highest number of nDCG@10=1 sessions, and almost the lowest number of nDCG@10=0 sessions (about a quarter of all sessions; a little worse than the ECxCGxPRF runs).

To understand how Endicott's three runs compare with runs submitted by other participants, consider Figure 2. In this figure, we take the difference in nDCG@10 for Endicott's RL3 runs and the minimum (left column), median (center column), and maximum (right column) across all participant runs (including Endicott's) by session. The difference reported on each graph shows the total difference for that run, or the area under the curve. For the median graphs, the difference is broken up between those where Endicott did better (above $y = 0$) and worse (below $y = 0$) than the median.



(a) `ECxCGxPRF RL1` (b) `ECxCGxPRF RL2` (c) `ECxCGxPRF RL3`

(d) `ECxSRMxPRF RL1` (e) `ECxSRMxPRF RL2` (f) `ECxSRMxPRF RL3`

(g) `ECxSRMxOS RL1` (h) `ECxSRMxOS RL2` (i) `ECxSRMxOS RL3`

Figure 1: nDCG@10 for the 100 judged sessions of RL1, RL2, and RL3 for each run group. Sessions are not necessarily aligned between graphs.

|  | **Run − Median** |  |  |
|---|---|---|---|
|  | $\mathbf{\Delta > 0}$ | $\mathbf{\Delta = 0}$ | $\mathbf{\Delta < 0}$ |
| `ECxCGxPRF RL3` | 39 | 23 | 38 |
| `ECxSRMxPRF RL3` | 37 | 21 | 42 |
| `ECxSRMxOS RL3` | 38 | 21 | 41 |

Table 3: Number of sessions where Endicott's runs are above, at, and below the median nDCG@10 across all participant runs.

In Figure 2 (b), (e), and (h), we see the Endicott runs have a higher mass above the median than below. Plots (c), (f), and (i), however, show that Endicott's runs were well below the maximum performance. Indeed, Table 3 shows that each run demonstrates sub-median performance on roughly as many or more sessions as above-median performance (roughly 40 of the 100 sessions).

## 6 Conclusions and Future Directions

Of the three run groups submitted by Endicott, the highest performing run, `ECxSRMxPRF RL3` used clicked document summaries to model relevance and re-rank spam-filtered baseline results, then boosted those results with recommendations from a click graph created over the 2013 and 2014 session data. While nDCG@10 was in general greater than the median performance across participant submissions, there is still much room for improvement. We have not conducted a failure analysis at this time, though such an analysis is crucial to understanding how the proposed methods may be improved. We leave such an analysis to future work.

## References

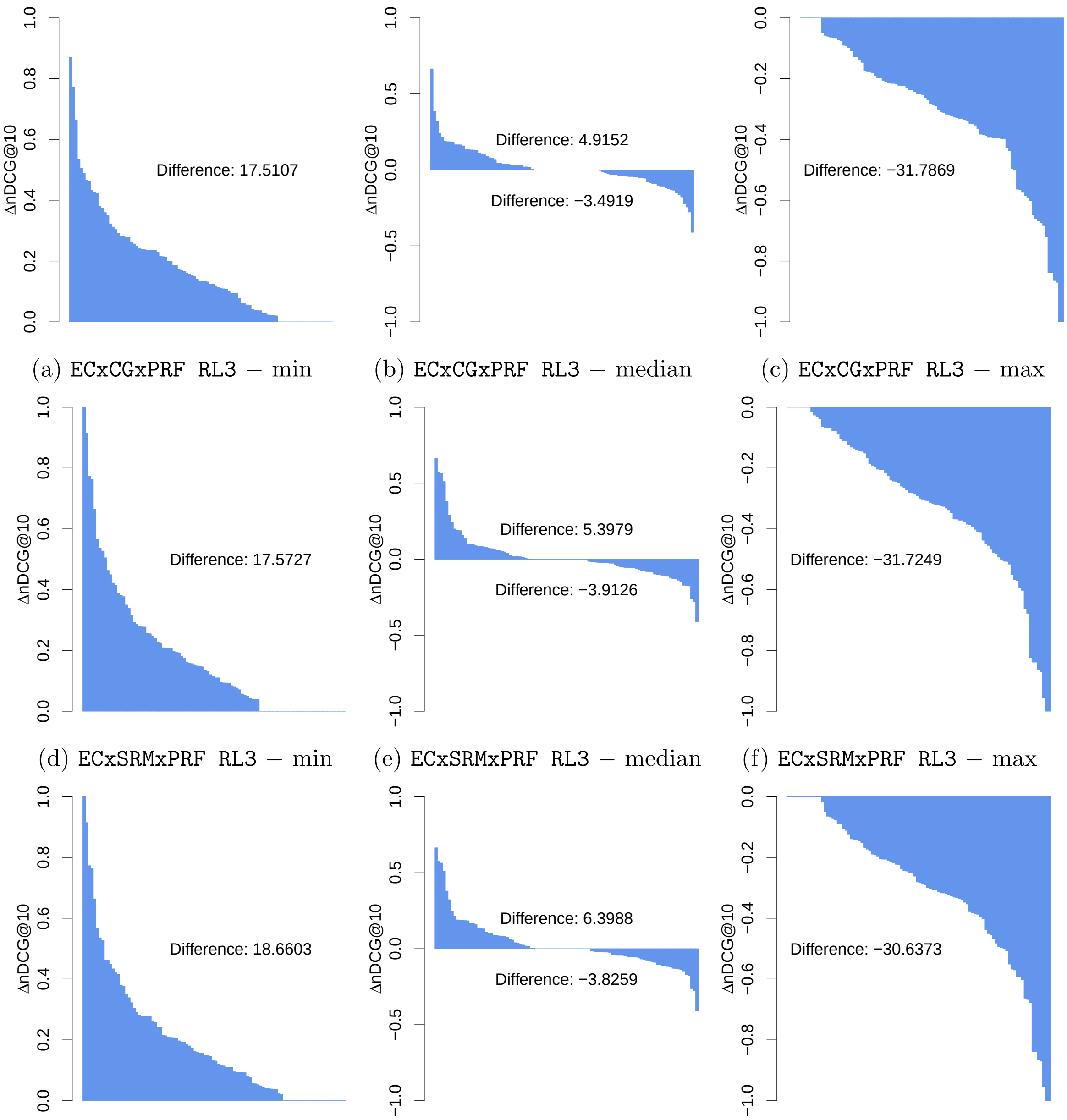
[1] G. Cormack, M. Smucker, and C. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval*, 14(5):441–465, 2011.

[2] N. Craswell and M. Szummer. Random walks on the click graph. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 239–246. ACM, 2007.

[3] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, pages 271–279. ACM, 2003.

[4] J. Jiang and D. He. Pitt at TREC 2013: Different Effects of Click-through and Past Queries on Whole-session Search Performance. In *Proceedings of the 22nd Text Retrieval Conference*, 2013.

[5] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127. ACM, 2001.



(a) `ECxCGxPRF RL3` — min

(b) `ECxCGxPRF RL3` — median

(c) `ECxCGxPRF RL3` — max

(d) `ECxSRMxPRF RL3` — min

(e) `ECxSRMxPRF RL3` — median

(f) `ECxSRMxPRF RL3` — max

(g) `ECxSRMxOS RL3` — min

(h) `ECxSRMxOS RL3` — median

(i) `ECxSRMxOS RL3` — max

Figure 2: Differences between RL3 nDCG@10 and the participant minimum, median, and maximum nDCG@10 by session. Sessions are not necessarily aligned between graphs.