Different Effects of Click-through and Past Queries on Whole-session Search Performance

Jiepu Jiang
School of Computer Science,
University of Massachusetts Amherst
jpjiang@cs.umass.edu

Daqing He
School of Information Sciences,
University of Pittsburgh
dah44@pitt.edu

ABSTRACT

Past search queries and click-through information within a search session have been heavily exploited to improve search performance. However, it remains unclear how do these two data source contribute to whole-session search performance due to the lack of reliable evaluation approaches. For example, as pointed out last year (Jiang, He, & Han, 2012), using past search queries as relevance feedback information can make search results of the current query similar to previous queries' results. Such issues cannot be disclosed by ad hoc search evaluation metrics such as nDCG@10.

Therefore, in this paper, we focus on analyzing the effects of past queries and click-through information on whole-session search performance. We adopted alternative evaluation approaches other than the TREC official ones. We found that: past queries may seemingly enhance nDCG@10 by retrieving previously returned results, which is difficult to result in real improvements of whole-session search performance; in comparison, click-through can enhance search performance without sacrificing search novelty, consequently leading to improved search performance across the whole session.

Keywords

Search session; TREC; evaluation; relevance feedback.

1. RETRIEVAL METHODS

The methods adopted by the PITT group in 2011 and 2012 (Jiang, Han, Wu, & He, 2011; Jiang, He, & Han, 2012) are variants of the context-sensitive relevance feedback (Shen, Tan, & Zhai, 2005). This method adopts the framework of the KL-Divergence language model (Lafferty & Zhai, 2001; Zhai & Lafferty, 2001). The query language models are estimated based on the current search query, past search queries, and click-through documents.

Four different query model estimation methods were proposed by Shen et al. (Shen et al., 2005). In our experiments, we adopt the "FixInt" method because both Shen et al. and we found that it has better performance than other methods introduced in (Shen et al., 2005). Let q_k be the kth query in the current search session, FixInt estimates query model θ_k as Eq(1): $P(w|q_k)$ is the current query's MLE model; $P(w|H_c)$ and $P(w|H_q)$ are, respectively, the relevance feedback models using click-through documents and past queries. Eq(2) and Eq(3) show details of H_c and H_q: C_i is the concatenation of all clicked documents' summaries returned by q_i; $P(w|q_i)$ is the ith query's MLE model.

$$P(w|\theta_k) = \alpha P(w|q_k) + (1-\alpha)\left[\beta P(w|H_c) + (1-\beta)P(w|H_q)\right] \quad (1)$$

$$P(w|H_c) = \frac{1}{k-1}\sum_{i=1}^{k-1} P(w|C_i) \quad (2)$$

$$P(w|H_q) = \frac{1}{k-1}\sum_{i=1}^{k-1} P(w|q_i) \quad (3)$$

2. EVALUATION

We use TREC session track 2011 & 2012 datasets for evaluation. Each dataset provides static search sessions and whole-session relevance judgments. A static search session is the search history of a real user in an interactive search system, including the users' search queries, click-through, and other information. For each static session, session-level relevance judgments are provided in the datasets: annotators judged documents regarding whether they are relevant to the search session (instead of individual queries).

The past TREC session tracks evaluated participant systems based on nDCG@10 of the last queries in each session. In comparison, we adopted the following experiment procedure to study the whole-session search performance. Let $\{q_1, q_2, \dots, q_n\}$ be a static search session in the dataset. We iteratively produce results for q_1, q_2, until q_n using FixInt. For each q_i, we employ the past search queries and click-through (if any) to produce search results. For q_1, FixInt downgrades to query likelihood model.

After generating results for each query in the static session, we calculate the following measures:

(1) **nDCG@10** (macro-average). Let $\{S_1, S_2, \dots, S_m\}$ be m static search sessions in a dataset, and $\{R_{i1}, R_{i2}, \dots, R_{in}\}$ be the results of the n queries in session S_i. We calculate the macro-average nDCG@10 of the dataset (referred to as nDCG@10) as follows:

$$\frac{1}{m}\cdot\sum_{i=1}^{m}\left(\frac{1}{n-1}\cdot\sum_{j=2}^{n} nDCG@10\left(R_{ij}\right)\right)$$

Note that we do not count the first query of each session because for the first query there is no search history information available (and therefore it contributes nothing to the evaluation of click-through and past queries).

(2) **inDCG@10** (macro-average). We proposed the inDCG@10 metric in (Jiang, He, Han, Yue, & Ni, 2012). It discounts the utility of documents in current search results if the documents have been retrieved in previous searches. Then, it calculates the nDCG@10 value based on the discounted utility of documents (referred to as inDCG@10). Similarly, we calculate inDCG@10 starting at the second query of each session and then compute the macro-average value across all the sessions of a dataset. We set different parameter values for inDCG@10: inDCG@10_88 (p=0.8, β=0.8), inDCG@10_85 (p=0.8, β=0.5), inDCG@10_58 (p=0.5, β=0.8), and inDCG@10_55 (p=0.5, β=0.5).

(3) **nsDCG@10** (normalized session DCG). For a static search session, nsDCG@10 concatenates the top 10 results of each query and evaluates the performance of the concatenated list of results. Please refer to (Kanoulas, Carterette, Clough, & Sanderson, 2010) for details. The same parameters have been adopted in this study.

(4) **Instance recall (instRec)**. This is a variant of a major metric adopted in the early TREC interactive tracks (Over, 2001). In

previous TREC interactive tracks, annotators identified relevant instances of each topic and marked up the occurrences of instances in documents. The metric "instance recall" was originally calculated as the proportion of relevant instances covered by the search results over all the identified instances. Here we calculate a similar measure by considering each single relevant document as an instance.

Let $\{D_i\}$ be the top 10 results of q_i, and D_R be the set of judged relevant documents. We concatenate the top 10 results of each query in the session as a whole set of retrieved documents (D_F). Then, we calculate instance recall (instRec) of the session as the proportion of D_R covered by D_F.

$$D_F = \bigcup_{i=1}^{n} \{D_i\} \qquad \text{instRec} = \frac{|D_F \cap D_R|}{|D_R|}$$

(5) **Instance recall gain (instRecGain)**. Apparently, instance recall will never decline with more queries being issued. Thus, we can calculate the contribution of each query's results to the overall instance recall of the session (referred to as instance recall gain). Let $\{D_i\}$ be the top 10 results of q_i. We calculate the instRecGain of q_k's results $\{D_k\}$ as follows. Then, we calculate the macro-average value of instance recall gain over all sessions.

$$\text{instRecGain}\left(D_k\right) = \text{instRec}\left(\bigcup_{i=1}^{k} \{D_i\}\right) - \text{instRec}\left(\bigcup_{i=1}^{k-1} \{D_i\}\right)$$

(6) **Jaccard similarity**. For a static session, we calculate for each unique pair of queries the Jaccard similarity of the pair of queries' top 10 results. Then, we calculate the macro-average value for each unique pair of queries across all search sessions. Although jaccard similarity is not a metric of search performance, it can help us analyze the novelty of search results.

3. RESULTS

Figure 1A and 1B show results of FixInt on TREC session track 2011 dataset using only click-through documents and past queries. Figure 2A and 2B show results on TREC session track 2012 dataset.

Our results indicate that:

(1) As we suspected in (Jiang, He, & Han, 2012), past queries can lead to serious decline of search novelty by making the results of the current query similar to previous queries' results. As shown in Figure 1B and 2B, in both datasets, the average Jaccard similarity of top 10 results can be increased from around 0.3 (α = 1.0, using sorely the user query) to around 0.8 (α = 0, using sorely the past queries). Whenever we increase the weight of past queries, there will be increase of the jaccard similarity.

In both datasets, nDCG@10 reaches the peak value when α = 0.5. At this moment, although we achieve 10% – 20% increase in nDCG@10, there is also 0.1 – 0.2 increase of jaccard similarity. In addition, instance recall dropped from 0.1079 (α = 1.0) to 0.0923 (α = 0.5) in 2011 dataset, and from 0.0881 (α = 1.0) to 0.0823 (α = 0.5) in 2012 dataset. This makes it difficult to assess whether there is a true improvement, because the improvement of nDCG@10 may come from one or two previously retrieved relevant documents.

Overall we found that past queries have no apparent effect on improving instance recall and instance recall gain. As shown in Figure 1B and 2B, instRec can at most can be increased from 0.1079 (α = 1.0) to 0.1104 (α = 0.9) in 2011 dataset, and from 0.0881 (α = 1.0) to 0.0896 (α = 0.8) in 2012 dataset.

(2) In comparison, our results suggest that click-through is a valuable relevance feedback information for improving search performance without sacrificing novelty. As shown in Figure 1A and 2A, for FixInt using click-through documents, the jaccard similarity of results will at most increase by 0.06 (comparing to an increase of around 0.5 when using past-queries).

In both datasets, click-through can at most increase nDCG@10 by 10% – 20%. In addition, instRec can also be increased by 10%, from 0.1079 (α = 1.0) to 0.1169 (α = 0.8) in 2011 dataset and from 0.0881 (α = 1.0) to 0.1007 (α = 0.5) in 2012 dataset. This indicates that click-through may increase the ranking of relevant documents without bringing previously retrieved ones to the top, which result in performance improvement all over the search session (indicated from instRec).

(3) Table 1 shows the correlation (Pearson's *r*) of various metrics on different parameter values of α and β. Results indicate that there are large disagreements between nDCG@10 & nsDCG@10 and instRec & instRecGain. However, inDCG@10_58 and inDCG@10_55 have satisfactory correlations with both nDCG@10 & nsDCG@10 and instRec & instRecGain. This suggests that inDCG@10 may be a less risky measure in model evaluation and optimization comparing to nDCG@10 and instRec.

Table 1. Pearson's correlation of metrics on different parameter values of α and β.

	TREC 2011		TREC 2012	
	nDCG@10	**instRec**	**nDCG@10**	**instRec**
nDCG@10	1.000	-0.235	1.000	0.245
inDCG@10_88	-0.013	0.956	0.496	0.952
inDCG@10_85	0.227	0.874	0.703	0.852
inDCG@10_58	0.483	0.719	0.773	0.793
inDCG@10_55	0.686	0.530	0.875	0.675
nsDCG@10	0.985	-0.244	0.994	0.204
instRec	-0.235	1.000	0.245	1.000
instRecGain	-0.226	0.979	0.228	0.992
avgJaccard	0.413	-0.957	0.180	-0.890

4. SUBMITTED RUNS

Based on our observations, we submitted three groups of runs:

(1) FixInt28: a FixInt run optimizing macro-average nDCG@10 (α = 0.2, β = 0.8). FixInt28N further applied the ranking discount method we proposed last year to FixInt28.

(2) FixInt58: a FixInt run optimizing instance recall (α = 0.5, β = 0.8). FixInt58N further applied the ranking discount method we proposed last year to FixInt58.

(3) KM1 and KM1N: new retrieval models aiming at whole-session relevance. Unfortunately, we did not notice there is a limit of four runs per group and therefore these two runs have not been evaluated in TREC 2013.

5. CONCLUSIONS

We studied the effects of past queries and click-through on whole-session search performance using alternative evaluation approaches other than TREC official ones. Our results indicate that it is risky to utilize past queries because we may easily run into the problem of making results similar to previously retrieved ones. In comparison, click-through documents seem to be a more valuable resource to achieve both query-level search performance and whole-session search performance. Therefore, we advocate more careful explanation of performance in studying session search methods.

At the time of writing this notebook paper, we did not get access to the qrels. Therefore, we did not further analyze this year's results at this moment. This will be analyzed in the final version.

6. REFERENCES

[1] Jiang, J., Han, S., Wu, J., & He, D. (2011). Pitt at TREC 2011 session track. In *Proceedings of the 20th Text REtrieval Conference, (TREC 2011)*.

[2] Jiang, J., He, D., & Han, S. (2012). On Duplicate Results in a Search Session. In *Proceedings of the 21st Text REtrieval Conference, (TREC 2012)*.

[3] Jiang, J., He, D., Han, S., Yue, Z., & Ni, C. (2012). Contextual evaluation of query reformulations in a search session by user simulation. In *Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM '12)* (p. 2635). New York, New York, USA: ACM Press. doi:10.1145/2396761.2398710

[4] Kanoulas, E., Carterette, B., Clough, P., & Sanderson, M. (2010). Session track overview. In *The 19th Text REtrieval Conference Notebook Proceedings (TREC 2010)*.

[5] Lafferty, J., & Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 111–119). New York, NY, USA: ACM. doi:10.1145/383952.383970

[6] Over, P. (2001). The TREC interactive track: an annotated bibliography. *Information Processing & Management*, *37*(3), 369–381.

[7] Shen, X., Tan, B., & Zhai, C. (2005). Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '05* (p. 43). New York, New York, USA: ACM Press. doi:10.1145/1076034.1076045

[8] Zhai, C., & Lafferty, J. (2001). Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the tenth international conference on Information and knowledge management* (pp. 403–410).

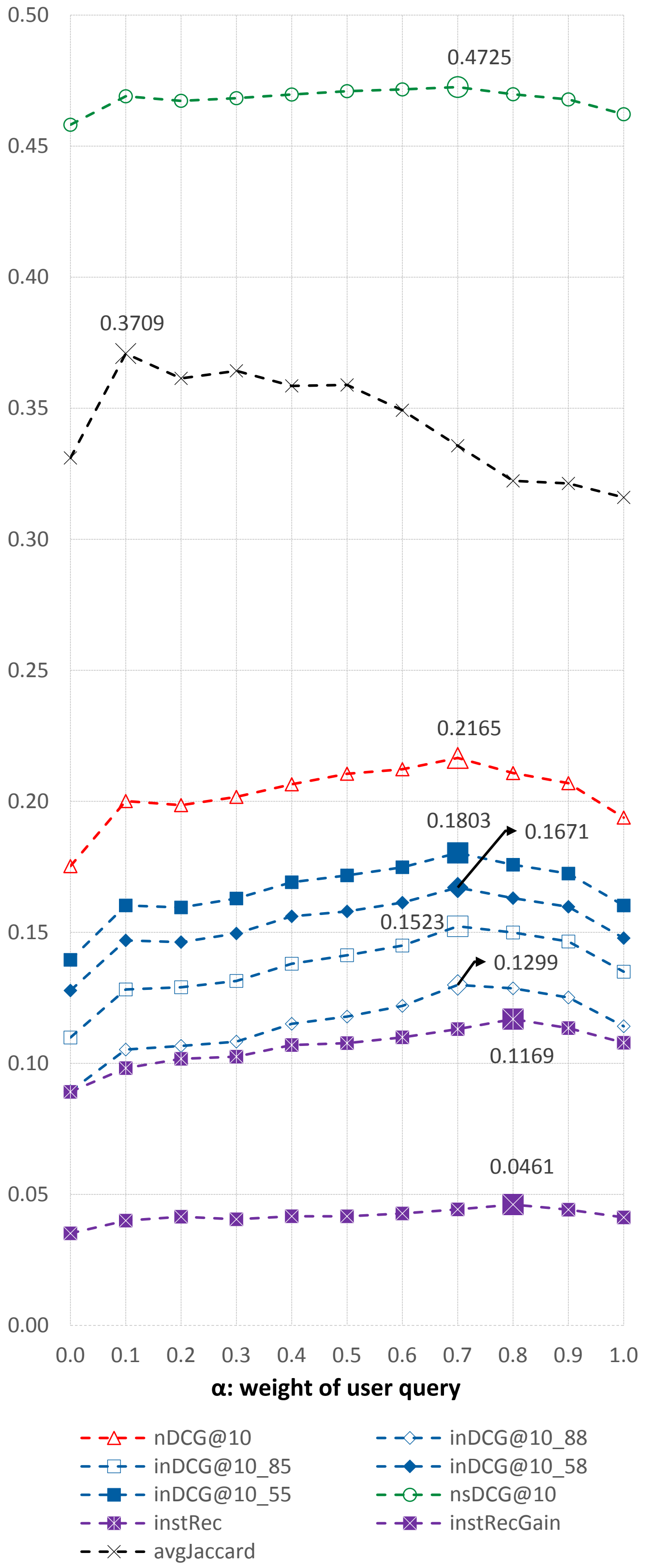

Figure 1A. Values of metrics under different parameter values on TREC session track 2011 dataset (FixInt using only click-through information, tuning the weight of the user query (α) from 0 to 1.0).

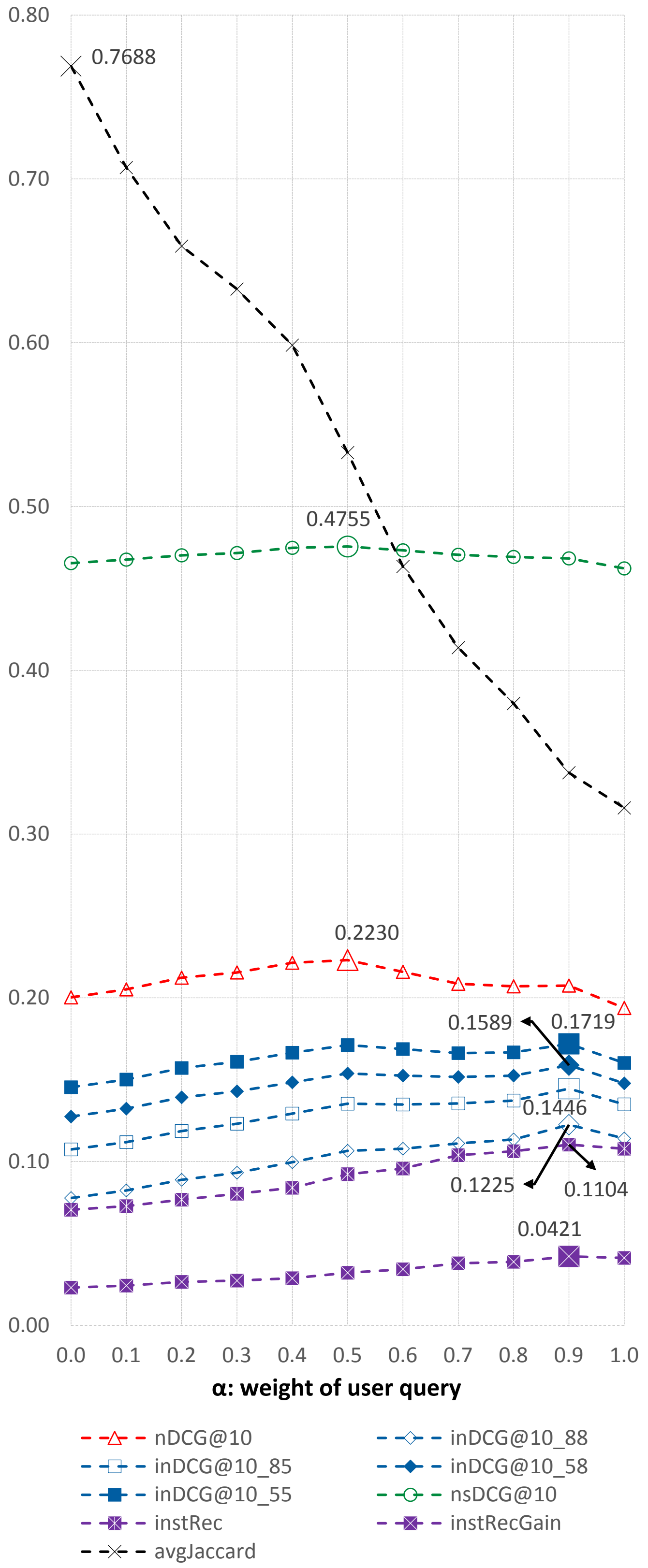

Figure 1B. Values of metrics under different parameter values on TREC session track 2011 dataset (FixInt using only past queries, tuning the weight of the user query (α) from 0 to 1.0).

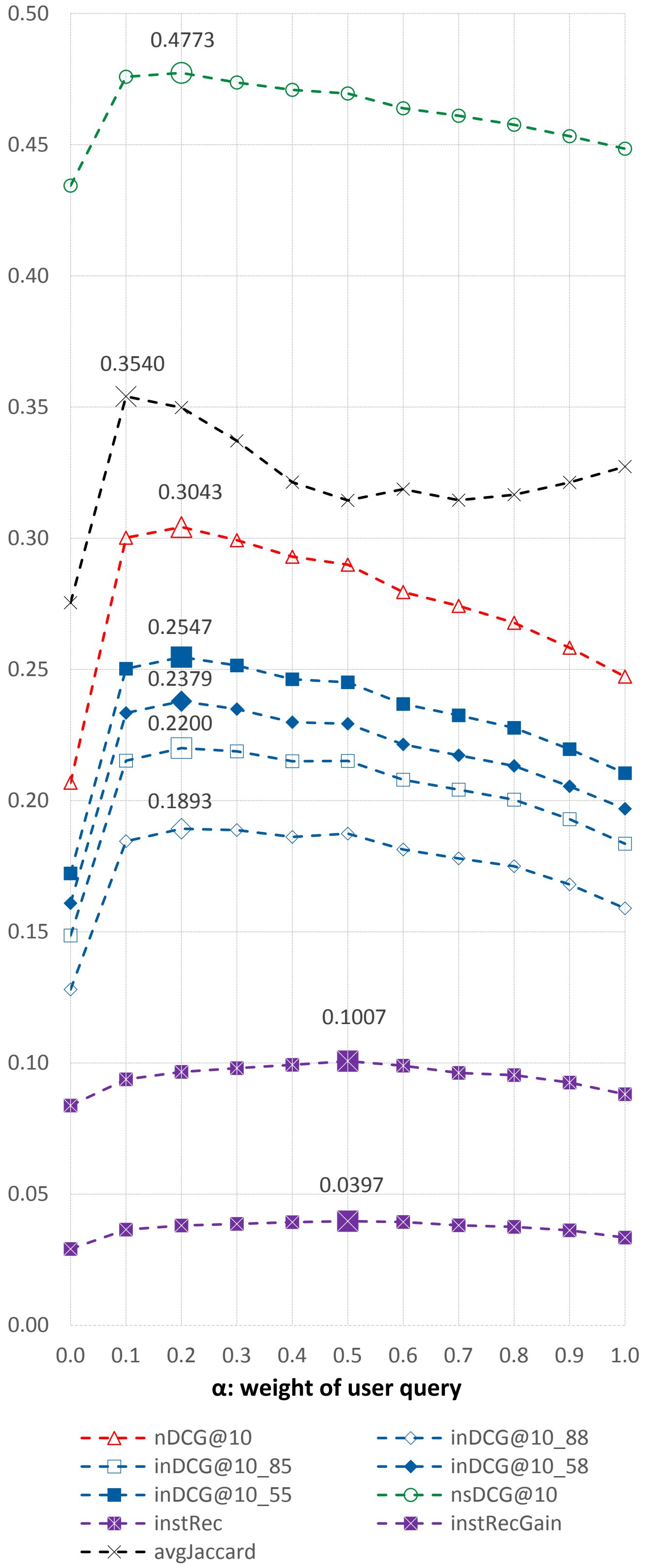

Figure 2A. Values of metrics under different parameter values on TREC session track 2012 dataset (FixInt using only click-through information, tuning the weight of the user query (α) from 0 to 1.0).

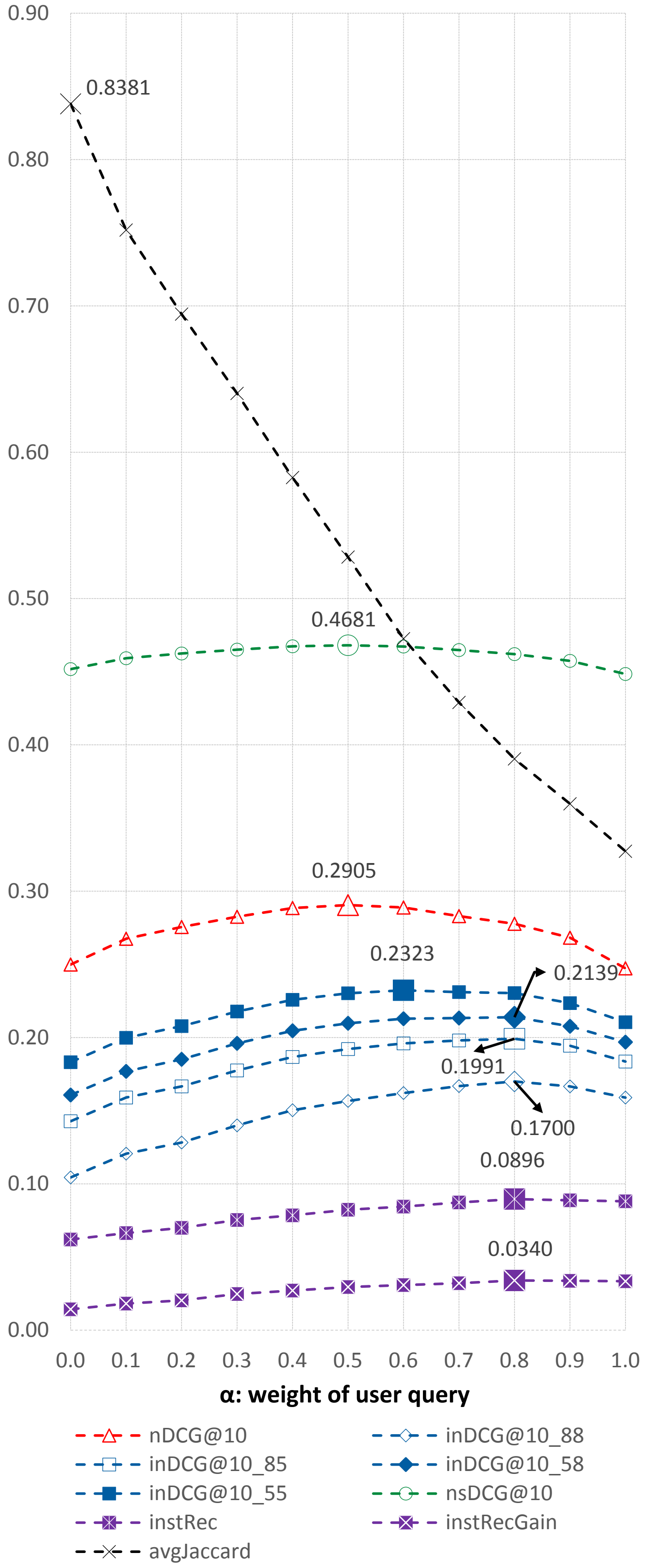

Figure 2B. Values of metrics under different parameter values on TREC session track 2012 dataset (FixInt using only past queries, tuning the weight of the user query (α) from 0 to 1.0).