Investigating per Topic Upper Bound for Session Search Evaluation

Zhiwen Tang
Department of Computer Science
Georgetown University
zt79@georgetown.edu

Grace Hui Yang
Department of Computer Science
Georgetown University
huiyang@cs.georgetown.edu

ABSTRACT

Session search is a complex Information Retrieval (IR) task. As a result, its evaluation is also complex. A great number of factors need to be considered in the evaluation of session search. They include document relevance, document novelty, aspect-related novelty discounting, and user's efforts in examining the documents. Due to increased complexity, most existing session search evaluation metrics are NP-hard. Consequently, the optimal value, i.e. the upper bound, of a metric highly varies with the actual search topics. In Cranfield-like settings such as the Text REtrieval Conference (TREC), scores for systems are usually averaged across all search topics. With undetermined upper bound values, however, it could be unfair to compare IR systems across different topics. This paper addresses the problem by investigating the actual per topic upper bounds of existing session search metrics. Through decomposing the metrics, we derive the upper bounds via mathematical optimization. We show that after being normalized by the bounds, the NP-hard session search metrics are then able to provide robust comparison across various search topics. The new normalized metrics are experimented on official runs submitted to the TREC 2016 Dynamic Domain (DD) Track.

KEYWORDS

Session Search; Evaluation; Normalization

1 INTRODUCTION

Session search is a complex search task which involves multiple iterations of searches in order to accomplish a complex information need. In the process, multiple queries are issued or reformulated and multiple runs of search results are returned by the search engine and examined by the user. Session search systems learn the users' search intents from the interactions to satisfy the complex information need of the entire session. The complexity of session search results comes from many factors, including the relevance of documents, the cost of reviewing documents, how to discount the relevance score of a document that is ranked lower in a list or is returned later at a later iteration. The challenge of evaluating session search lies in how to evaluate the search engine effectiveness throughout the entire course of the search.

Most metrics measure the amount of information a user gains during a search session. The gain is usually represented as the sum of relevance of the documents that have been returned so far. The relevance scores are usually assigned by third party annotators as ground truth scores. They can be graded or binary scores. Generally, as shown in previous research, it is believed that there is a positive correlation between a third party annotated relevance score and a user's satisfaction level [1, 6]. In this paper, instead of a user study, we will mainly focus on discussing evaluation in the Cranfield-like settings which rely on third party annotations.

Discounting document relevance is a common technique used in IR evaluation. There are two main types of discounting methods. First, discounting is done based on ranking orders. The idea is that the relevance scores of the lower ranked documents are discounted (as in DCG [7]) because it is assumed that users are less likely to read those documents hence less gains come from them. Second, discounting is done based on content redundancy. The relevance scores of documents that repeat on topics that have appeared in earlier documents are discounted (as in α-nDCG [4]), too.

Some IR evaluation metrics also take into account users' efforts (sometimes also known as *cost*). The cost can be regarded as a user's effort, both mentally and/or physically, spent in the session. It is usually simplified as the time spent in completing the entire task/session [12, 16], or the aggregated lengths of documents a user has read [19]. As more user efforts are put into search activities, the user's satisfaction level is expected to decrease. Kokubu et al. has found a positive correlation between user satisfaction and the reciprocal rank of relevant results in QA systems [10], which implies that a user is more satisfied if less documents are needed to examine in order to find the answers. The cost of a document may be discounted as well (as in Expected Utility [19]).

Most session search metrics handle all of the above factors and usually combine them into a single formula. The complexity of these metrics is quite high and most of them are NP-Hard. Consequently, the optimal value, i.e. the upper bound, of a metric highly varies with the actual search topics. That is to say, the bounds are not only decided by the mathematical definition of each metric, but also affected by the ground truth data of each topic. For instance, some search topics might be easy (with high optimal value for those topics) because many relevant documents are available and many search systems can achieve pretty high evaluation metric scores on those topics; while other topics might just be difficult for all search systems (with low optimal values for those topics). In Cranfield-like settings such as the Text REtrieval Conference (TREC), however, evaluation metric scores for systems are usually averaged across

ICTIR'17, October 1–4, 2017, Amsterdam, The Netherlands
© 2017 ACM. ISBN 978-1-4503-4490-6/17/10…$15.00
DOI: http://dx.doi.org/10.1145/3121050.3121069

Table 1: Document Relevance Scores in Ground Truth

topic	document
1 (2 subtopics)	d1(1.1, 1) d2(1.2, 3)
2 (4 subtopics)	d1(2.1, 4) d2(2.2, 4) d3(2.2, 2) d4(2.3, 4) d5(2.4, 4)

Table 2: The Importance of Normalization

systems	CT-topic 1	CT-topic 2	CT avg	normalized-CT avg
optimal	4	17	/	/
system 1	1	16	8.5	0.596
system 2	3	14	8.5	0.787

all search topics without considering the per topic upper bound a metric could achieve. Neglecting the differences in those bounds could be unfair when evaluating the systems across different topics.

A toy example is shown in Tables 1 and 2. Suppose two systems are evaluated on two topics with two and four subtopics respectively. In Table 1, $d1(1.1, 1)$ means document $d1$ is relevant to subtopic 1.1 with a relevance score of 1 (scaled from 1 to 5). Assume that each system returns five documents. *System 1* found $d1$ for *topic*1 and $d1, d3, d4, d5$ for *topic 2*; *System 2* found $d2$ for *topic 1* and $d1, d2, d4, d5$ for *topic 2*. Other documents returned are irrelevant. Suppose that Cube Test [12] is used to evaluate session search effectiveness. We also assume that equal amount of time is needed to read each document and the discounting factor is 0.5. The optimal scores and the actual scores are shown in Table 2. As we can see here, the two systems' averaged CT scores are the same (8.5) thus it is hard to tell which system is better. However, the raw CT scores do not reflect the actual performance of the two systems. *Topic 2* has a high upper bound (17), which suggests an easy search topic. Therefore, the higher CT score from *system 1* (16) does not support that it is a better system than *system 2* which does an impressive job on the more difficult topic (*topic 1* with an upper bound of 4). The issue can be resolved by proper normalizing the raw scores by the per topic upper bounds, as shown in the last column in Table 2. The example demonstrates that knowing the per topic metric bounds is very important in fairly evaluating IR systems.

In this paper, we investigate existing session search evaluation metrics and focus on the following Research Question (**RQ**): *What is the best possible optimal metric value that a system could achieve?* To answer this question, we first take apart existing session search metrics into components. We then analyze the rationale behind each component and the ways used to combine them. Based on the component-based analysis, we compute the optimal score or the bounds of these metrics, which are then used for score normalization later. The new normalized metrics are experimented on the official runs submitted to the TREC 2016 Dynamic Domain (DD) Track. Our results show that these NP-hard session search metrics are then able to provide robust comparison across different topics.

This paper is organized as follows. Literature review is in section 2. Existing session search metrics are analyzed in sections 3 and section 4. The optimization method is detailed in section 5 and the experiments are shown in section 6. Section 7 concludes the paper.

2 RELATED WORK

2.1 Session Search Evaluation

Different from ad-hoc retrieval where a single query is asked and a single ranked list of documents is returned for that query, session search involves multiple ranked lists of documents generated from multiple runs of queries. To the best of our knowledge, the following metrics have been proposed to evaluate the effectiveness of session search results. They are Session-based DCG (sDCG) [8], Cube Test (CT) [12] and Expected Utility (EU) [19].

Relevance is a critical element in IR evaluation. sDCG [8] is a session search evaluation metric mainly about relevance. It extends the Discounted Cumulative Gain (DCG) [7]. Besides being discounted based on ranking position in the same list, documents ranked at a later iteration also get discounted because they are assumed less likely to be read. In this metric, discounted relevance is the main factor being considered.

Novelty of documents is another important factor in IR evaluation. IR systems that return more novel results should be rewarded more. Metrics use nuggets (in EU [19]), subtopics (in CT [12]), or intents to refer to the similar idea of "aspects" that compose into a complete search topic for a session. If a document is related to a nugget/subtopic that is previously found, then its relevance score should be discounted.

Recent research has shown that another dimension, the effort spent by the user, should also be incorporated into IR evaluation[21]. Metrics measure the user effort include CT [12], EU [19] and Time-bias gain [16]. The first two are session search evaluation metrics. CT represents the effort as the time spent reviewing the documents. EU represents it as the total lengths of documents having been read. Time-bias gain calculates the expected time spent reading the documents.

Many session search metrics assumes that the user will read all the document returned in every iteration from the beginning to the end. However, it is probably not true. Users may choose to stop early or read the documents in a different order. Kanoulas et al. addressed this problem by incorporating the user's reviewing paths into IR evaluation [9]. EU [19] also takes the reviewing path into consideration. Since users may not read all the documents, the utility of the search session may differ for different users. EU uses a uniformly distributed user model to calculate the final scores.

2.2 Score Normalization and Optimization

Table 2 demonstrates that topic difficulty level could affect fairness in IR evaluation. It has been a while since this issue was recognized. As an early attempt, Robertson proposed to use Geometric Mean of Average Precision (GMAP) [13], which can be seen as the arithmetic mean of the logarithms of Average Precisions (AP). GMAP reduces the variance among high AP scores and enlarges the variance among low AP scores. However, the physical meaning of the logarithm function remains unclear.

Another approach was proposed by Webber et al. to first standardize the raw scores, then map the standardized scores into the range of [0, 1] by applying the cumulative density function of a standard normal distribution [17]. Their method provides the opportunity of comparing the scores across different topics and collections. Sakai used a similar method [15], where a linear transformation is applied to the standardized scores. Lee et al. proposed to use Generalized Adaptive-Weight Mean (GAWM) to aggregate the scores from different topics [11]. Based on the Fixed Point Theorem, GAWM assigns more weights to topics on which the search systems' effectiveness has higher variance.

In various ways, these existing methods are able to average scores across different topics. However, what is missing is that they all provide little information about how to make use of the averaged scores to guide an IR system to do better. Here we propose to inform the search systems not only the averaged scores, but also how far away they are from the best they can do via providing the knowledge of per topic upper bounds of the metrics.

The work most relevant to our paper is [3]. It computes the optimal values of a few IR metrics, including S-recall, S-precision [22] and α-DCG [4], which are not used for session search though. In [3], Carterette found that optimizing metrics considering document novelty is usually NP-Complete [3]. Given different search topics, the performance of search systems varies and there is no general guideline for optimizing novelty and relevance at the same time. Our paper is along the same line of research of [3], with an emphasis on session search metrics.

3 EXISTING SESSION EVALUATION METRICS

The complexity of the session search task poses challenges to its evaluation. This section presents existing session search metrics, sDCG [8], Cube Test [12] and Expected Utility [19]. The following notations are used in the rest of the paper.

For a search session:

- i: the index of a search iteration in a session with total L iterations, $i = 1, ..., L$.
- $list_i$: the returned document list at the i^{th} iteration.
- j: the position of a document within an iteration, $j = 1, ..., |list_i|$.
- k: the id of a document in the entire corpus, $k = 1, ..., n$.
- c: a subtopic/nugget of the search topic.
- θ_c: the importance/relevance of a subtopic or nugget.

For document $d_{i,j}$, which is at the j^{th} position in the i^{th} iteration:

- $rel_c(i, j)$ is the relevance score of $d_{i,j}$ regarding c if we consider subtopic/nugget level relevance. Otherwise if we only consider document level relevance, $rel(i, j)$ is the relevance score of $d_{i,j}$.
- $cost(i, j)$: the cost or user effort of examining $d_{i,j}$.

Dynamic parameters during the session:

- $n(c, i, j - 1)$: the number of documents that are relevant to c returned before the j^{th} document in the i^{th} iteration.
- γ, bq, b : novelty, iteration, and within-iteration position discounting factors.

The definitions of sDCG, Cube Test and Expected Utility are shown in formulas 1, 2 and 3, respectively.

$$sDCG = \sum_{i=1}^{L} \sum_{j=1}^{|list_i|} \frac{rel(i, j)}{(1 + \log_b j) * (1 + \log_{bq} i)} \tag{1}$$

$$CT = \frac{\sum_{i=1}^{L} \sum_{j=1}^{|list_i|} \sum_c \theta_c * rel_c(i, j) * \gamma^{n(c,i,j-1)}}{\sum_{i=1}^{L} \sum_{j=1}^{|list_i|} cost(i, j)} \tag{2}$$

For simplification, without loss of generality, the cap for maximum relevant scores of CT is neglected here. We also assume that the amount of time needed for reviewing each document in CT is equal.

$$EU = \sum_{\omega} P(\omega) \left(\sum_{(i,j)\in\omega} \left(\sum_{c \in d_{i,j}} \theta_c * \gamma^{n(c,i,j-1)} \right) - a * cost(i, j) \right) \tag{3}$$

Expected Utility assumes that a user only reviews a subset of documents being returned, which is denoted as ω. $P(\omega)$ is the probability that ω are reviewed. a is a coefficient to adjust the relationship between the gain and cost. The cost of reviewing each document is measured by their document lengths.

As we can see, all these metrics attempt to handle multiple aspects of session search, such as relevance and novelty of documents and time spent reviewing them. Mixing various factors into a single metric has two consequences. First, it is difficult to understand them. Second, most metrics here are NP-hard thus they would be unreliable to be directly used as optimization goals in supervised ranking algorithms such as in learning to rank. In the following sections, we decompose these metrics into a few simple components and further analyze them.

4 DECONSTRUCTING THE METRICS

We observe that there are common components shared by almost all the session search metrics. They are *gain*, *cost*, *ranking discount* and *novelty discount*.

Gain: the gain of each document is its raw relevance score. *Gain* represents the amount of useful information a user can learn from the document.

Cost: the cost of each document is its length or time spent examining it. *Cost* represents the effort the user needs to spend on that document.

Both *Gain* and *Cost* are inherent attributes of a document. It means they are irrelevant to the environment or context in which the document is present. Such context includes being at a specific position among a list and the content of other documents being examined before.

When considering the context, the raw *Gain* or *Cost* of a document may be adjusted to reflect the influences of the context. For instance, they are usually being discounted when a document appears at a lower ranking position, or the document contains redundant information compared to an early document. We thus have two types of discounts.

Ranking discount: Discounting that is based on the original ranking position of a document. It is irrelevant to the document's own content. The rationale behind the ranking discount is that the lower a document ranks, the less likely the user will read it, the less the expected gain or expected cost comes out of this document. Decaying functions like logarithmic reduction factor $\frac{1}{1+\log_b x}$ are commonly used in ranking discount.

Novelty discount: Nuggets and subtopics both measure a user's knowledge coverage. If a document is related to a subtopic/nugget that the user read before, then it contributes less novel information about this subtopic/nugget, for which its value will be discounted. Decaying functions like exponential discounting function γ^x are commonly used in novelty discount. Novelty discount can be seen as a general form of ranking discount, where the ranking order is one within each subtopic or nugget.

Let us denote *Gain*, *Cost*, *Ranking discount* and *Novelty discount* by A, B, C, and D:

A = raw gain of each document B = ranking discount

C = novelty discount D = raw cost of each document

We propose to view metrics shown in formula 1, 2 and 3 as combinations of A, B, C, and D and apply a component-based analysis.

Specifically, sDCG does not take novelty into account, thus it only has components $A+B$. The definition of sDCG can be rewritten as

$$sDCG = Discounted\ Gain = \sum_{d} rank\_discount_d * gain_d$$

Ranking discount is not considered in CT, thus CT can be seen as $A + C + D$. Its definition can be rewritten as

$$CT = \frac{Discounted\ Gain}{Cost} = \frac{\sum_d \sum_c novelty\_discount_{d,c} * gain_{d,c}}{\sum_d cost_d}$$

EU takes all the components into consideration. It can be regarded as $A + B + C + D$:

$$EU = Discounted\ Gain - Discounted\ Cost$$
$$= \sum_{d} \sum_{c} novelty\_discount_{d,c} * rank\_discount_d * gain_{d,c}$$
$$- \sum_{d} rank\_discount_d * cost_d$$

An interesting question arose out of our abstraction is how to combine the *(Discounted) Gain* and *(Discounted) Cost* when both are present. CT divides *Gain* by *Cost* while EU subtracts *Cost* from *Gain*. Which one is more appropriate, subtraction or division?

Subtracting *Cost* from *Gain* is like calculating a "net gain". The assumption here is that *Gain* and *Cost* are of the same nature and can be directly added or subtracted. Is this true? In session search, *Cost* is the amount of time needed to review the documents or the aggregated length of reviewed documents, for which its unit can be seconds or words. *Gain* is the information obtained from the reviewed documents and its unit is still unknown. Based on dimensional analysis [2], if two things do not belong to the same dimension, it is probably incorrect to add or subtract them with each other (eg. adding area into length). Even if both measure the same thing, it may still not be ideal to sum them up unless the exchange rate is fixed (eg. 1 foot = 0.3048 meters).

When combining measurements from two different dimensions, in this case gain and cost, we think using division is more appropriate. No matter whether *Gain* and *Cost* measure in the same dimension or not, the result of division can still be seen as a measurement of the rate to achieve that gain. An effective IR system should return more *Gain* with the same *Cost* or provide the same *Gain* with less *Cost*. CT can be regarded as a speed function measuring how fast an IR system can satisfy an information need.

5 FINDING THE UPPER BOUND VALUES

This section focuses on computing the bound of the metrics after decomposing them into components. For a given topic, the bound of a metric is one of the topic's inherent propertites and is very useful in identifying where to make improvement to a search system. For example, if a system receives a very low score on a topic, it might be because of the poor retrieval model, or be because of a very difficult search topic. Without knowing the optimal score on the topic, it is difficult to separate the two cases. Moreover, without knowing the bound of every topic, it might also be unfair to average the scores across topics.

As an evaluation scheme, suppose we know the ground truth of a given search topic, which consists of the relevance scores and reviewing cost of each document, the importance of every subtopic and the number of documents returned at each iteration. The research problem here is to calculate the optimal scores (bounds) for these NP-hard metrics for a given topic.

After knowing the bounds, **normalizing the scores** would be straightforward. Normalization requires not only the upper bound, but also the lower bound. In this paper, for a given metric, the normalized score of a search system is

$$score_A = \sum_{t} \frac{raw\_score(t,A) - lower\_bound(t)}{upper\_bound(t) - lower\_bound(t)} \quad (4)$$

where A is the system, t is the topic and $raw\_score(t,A)$ is the raw metric value of system A on topic t. Note that for sDCG and CT, the lower bound is always zero, we thus will only focus on calculating their upper bounds. However, the lower bound of EU could be negative so both upper and lower bounds will be studied.

5.1 Optimization Methods

Components A (gain) and D (cost) become constants once ground truth data is provided. The challenge of computing the optimal scores/upper bounds lies in components B (ranking discount) and C (novelty discount). As discussed before, novelty discount could be seen as a more general form of ranking discount. Both components share the property that the lower (later) a document ranks, the more discount it receives. It implies that we can use the same optimization framework for both types of discounts.

For a single document ranked list, its discounted gain can be expressed as $\sum_j rel(j) * discount(j)$, where $rel(j)$ is the relevance score of the j^{th} document in this list. Since $discount(j)$ is only related to the ranking position j and not relevant to the document's relevance or reviewing cost, computing the optimal score or the bound of the ranked list is equivalent to finding the best permutation of documents that optimizes the metric score. The ranked list can be optimized based on the *rearrangement inequality* [5].

The *rearrangement inequality* states that

$$x_1 y_n + ... + x_n y_1 \le x_{\sigma(1)} y_1 + ... + x_{\sigma(n)} y_n \le x_1 y_1 + ... + x_n y_n$$

for all the real numbers $x_1 \le ... \le x_n$ and $y_1 \le ... \le y_n$. Moreover, $x_{\sigma(1)}, ...x_{\sigma(n)}$ is a permutation of $x_1, ..., x_n$. In our case, $rel(j)$ can be seen as x_i and $discount(j)$ can be regarded as y_i. The position that has the larger $discount(j)$ should be reserved for a document that has higher rel_j, if we would like to calculate the max value.

This method is also referred as *Probability Ranking Principle* in IR [14], which states that the overall effectiveness of an IR system can be achieved the best by ranking the documents by their usefulness in descending order. The method can give a feasible optimal score if only one ranking order needs to be optimized. However, when multiple ranked lists are required to be optimized simultaneously,

ALGORITHM 1: optimal score on a single ranking list

```
Input:
$|list_i|$ where $i = 1, 2...L$,
raw (relevance/cost) scores: $r_k$ of document $d_k$, $k = 1, 2, ..., n$,
discount function: $discount(i, j)$
optimization direction: $IsMaximize = true$ or $false$
Output: optimal score on a single ranked list
$POS = \{(1, 1), ..., (1, |list_1|), (2, 1), ..., (L, |list_L|)\}$
// Set of all possible document ranking positions
$SP = Queue(sort(POS)$ by $discount(i, j)$ in ascending order)
if IsMaximize then
    $D = Queue(sort(d_k)$ by $r_k$ in ascending order)
end
else
    $D = Queue(sort(d_k)$ by $r_k$ in descending order)
end
$opt = 0$
while SP is not empty and D is not empty do
    $(i, j) = SP.deQueue()$
    $d_k = D.deQueue()$
    $opt+ = r_k * discount(i, j)$
end
return opt
```

e.g. the optimization of CT requires the optimization of the ranked list within each subtopic, there may not exist a ranking order that can optimize all lists. Nonetheless, optimizing each required ranking list independently can approximate an overall bound.

Greedy algorithms have been proposed to approximate the optimal score in novelty-related metrics [3, 4], where multiple ranked lists are required to be optimized simultaneously. However, the focus of a greedy algorithm is to produce an ideal document list, which is not required for score normalization. It is also shown that the results yielded by a greedy algorithm can be far below the optimal score on certain topics [3]. From this point of view, optimization based on *rearrangement inequality* is more proper because it is able to lead to a tighter bound and is very efficient especially when the number of relevant documents is large.

Our optimization algorithm is shown in Algorithm 1. It handles both minimization and maximization based on different settings. In the algorithm, all the possible slots are first ranked based on the discount value it will receive. When maximization is needed, a document with higher raw score will be put at a position with higher $discount(i, j)$ value. When minimization is needed, the document with higher raw score will be assigned to a position with lower $discount(i, j)$ value. The algorithm forms the basis for optimizing of the session search metrics in the rest of this section.

5.2 sDCG

sDCG is essentially discounted cumulated gain for a search session. Computing the optimal sDCG score can be expressed as

$$maximize \sum_{i=1}^{L} \sum_{j=1}^{|list_i|} \frac{rel(i, j)}{(1 + \log_b j) * (1 + \log_{bq} i)} \tag{5}$$

Since subtopics are not considered in sDCG, only one document ranking list needs to be optimized. Algorithm 1 can be directly used by setting $IsMaximize = true$ and $discount(i, j) = \frac{1}{(1+\log_b j)*(1+\log_{bq} i)}$.

Different from the normalization method proposed in [8], there is no duplicated result in our ideal ranked lists. Our optimal sDCG score represents the optimal performance a system can achieve in the session. The range of sDCG score of any system on a topic is $[0, opt]$ when the session length L is fixed.

5.3 Cube Test

Computing the optimal score of CT is not as intuitive as that of sDCG. In fact, computing the optimal score of Cube Test is NP-Hard even for a special case of CT, which is defined as

$$maximize\ CT = \frac{\sum_c \sum_d novelty\_discount_{d,c} * gain_{d,c}}{\sum_d cost_d} \tag{6}$$

where $cost_d = 1$, $novelty\_discount_{d,c}$ is boolean and is set to 1 iff. document d is the first relevant document found on subtopic c.

In the special case, the discounted cumulated gain can be considered as the number of subtopics found by the IR system and the aggregated cost is the number of documents returned. The optimal CT score in this special case should have the highest number of subtopics found with the minimum number of documents returned.

The optimization problem can be transformed into the *Minimum Edge Dominating Set* problem in graph theory, which is NP-Hard [20]. An *edge dominating set* is a subset of edges satisfying the property that every edge that is not in this subset is adjacent to at least one edge in this subset. The *minimum edge dominating set* is one such subset of edges that has the smallest size.

Here each document is considered as an edge and each subtopic is considered as a vertex. Then, documents (edges) in the *minimum edge dominating set* are the ones that need to be returned so as to achieve the optimal performance in the special form of CT. Because the special CT is NP-Hard, computing the optimal value of general CT is also NP-Hard. Nonetheless, CT's upper bound can still be computed based on *rearrangement inequality*. The optimal CT can be achieved by

$$\begin{aligned} &maximize \sum_{i=1}^{L} \sum_{j=1}^{|list_i|} rel_c(i, j) * \gamma^{\sum_{l=1}^{i-1} |list_l| + (j-1)} \quad \forall c \\ &minimize \sum_{i=1}^{L} \sum_{j=1}^{|list_i|} cost(i, j) \end{aligned} \tag{7}$$

Formula 7 requires to maximize a number of objective functions and minimize one objective function. The number of objective functions need to be maximized equals to the number of subtopics, i.e., $\#(c)$. The upper bound of CT can be derived by optimizing each of these target functions independently using Algorithm 1 and then combine them according to formula 2. For the maximization target functions, the input contains the raw relevance scores of all the documents regarding a given subtopic, and $discount(i, j) = \gamma^{\sum_{l=1}^{i-1} |list_l| + (j-1)}$. For the minimization target function, the input contains the raw cost scores of all the documents, and

$discount(i, j) = 1$. The approximated upper bound may not be feasible in real situations. However, it provides good approximations of the real bound.

5.4 Expected Utility

EU assumes that the user will scan the returned document in a top-down fashion with a probability p of stopping at some document in the current iteration. Once the user stops reviewing, (s)he will start the next search iteration. In order to make the computation tractable, Yang et al. [19] approximated the computation of EU by formula 8.

$$EU = \frac{1}{1-\gamma}\left(\sum_{c} \theta_c \left(1 - \gamma^{\sum_{\omega} P(\omega) n(c,\omega)}\right)\right) - a \sum_{\omega} P(\omega) len(\omega) \tag{8}$$

where $n(c, \omega)$ is the number of appearances of nugget c in the reviewed document subset ω and $len(\omega)$ is the total length of documents in ω.

The computation of $\sum_{\omega} P(\omega)n(c, \omega)$ and $\sum_{\omega} P(\omega)len(\omega)$ can be further expanded. $\sum_{\omega} P(\omega)n(c, \omega) = \sum_{i=1}^{L} \sum_{h_i=1}^{|list_i|} P(h_i) \sum_{j=1}^{h_l} rel_c(i, j)$ $= \sum_{i=1}^{L} \sum_{j=1}^{|list_i|} rel_c(i, j) \sum_{h_i=j}^{|list_i|} P(h_i) \sum_{i=1}^{L} \sum_{j=1}^{|list_i|} rel_c(i, j)(1-p)^{j-1}$, where h_i is the position where the user stops in the i^{th} iteration of the session. Similar transformation can also be applied to $\sum_{\omega} P(\omega)len(\omega) = \sum_{i=1}^{L} \sum_{j=1}^{|list_i|} cost(i, j)(1 - p)^{j-1}$.

In EU, each nugget c has its own graded importance score θ_c. The relevance between a document and a nugget $rel_c(i, j)$ is binary (a document does or does not contain a nugget). The cost of a document, $cost(i, j)$, is its length.

The maximization of formula 8 is also NP-Hard. It is achieved by

$$\begin{aligned} maximize & \sum_{i=1}^{L} \sum_{j=1}^{|list_i|} rel_c(i, j) * (1 - p)^{j-1} \quad \forall c \\ minimize & \sum_{i=1}^{L} \sum_{j=1}^{|list_i|} cost(i, j) * (1 - p)^{j-1} \end{aligned} \tag{9}$$

Using similar methods as in CT, the upper bound of EU can be obtained by maximizing #(c) target functions and minimizing 1 target function independently with Algorithm 1. Note that the lower bound of EU is negative. It is because an IR system may return documents that are all irrelevant but still requires user's effort to read them, which leads to a negative EU score. The lower bound of EU can be derived by

$$maximize \sum_{i=1}^{L} \sum_{j=1}^{|list_i|} cost(i, j)(1 - p)^{j-1} \tag{10}$$

which can also be approximated using Algorithm 1.

6 EXPERIMENTS

In this section, we experiment on the dataset and the official runs submitted to the TREC 2016 Dynamic Domain (DD) Track [18]. The dataset contains documents in various formats, including html pages and tweets. Each search topic contains several subtopics addressing different aspects of the topic. Every document may be relevant to multiple subtopics with different relevant grades. Subtopics within a topic are assigned identical weights. In total, there are 53 topics and 242 subtopics, with an average of 4.57 subtopics per topic. The ground truth data was created by NIST assessors. It contains 14, 597 relevant documents, with an average of 291.47 relevant documents per topic[1]. Table 3 shows an example topic from TREC 2016 DD Track.

Table 3: Topic sample in TREC-DD 2016

Topic/Subtopic id	Topic/Subtopic name
DD16-1	US Military Crisis Response
– DD16-1.1	West African mission
– DD16-1.2	Key Personnel
– DD16-1.3	Personnel safety protocols

We have conducted two sets of experiments. The first mainly studies the upper bounds (or the bound size, if the lower bound of the metric is negative) of the session search metrics. The second studies the influence of score normalization.

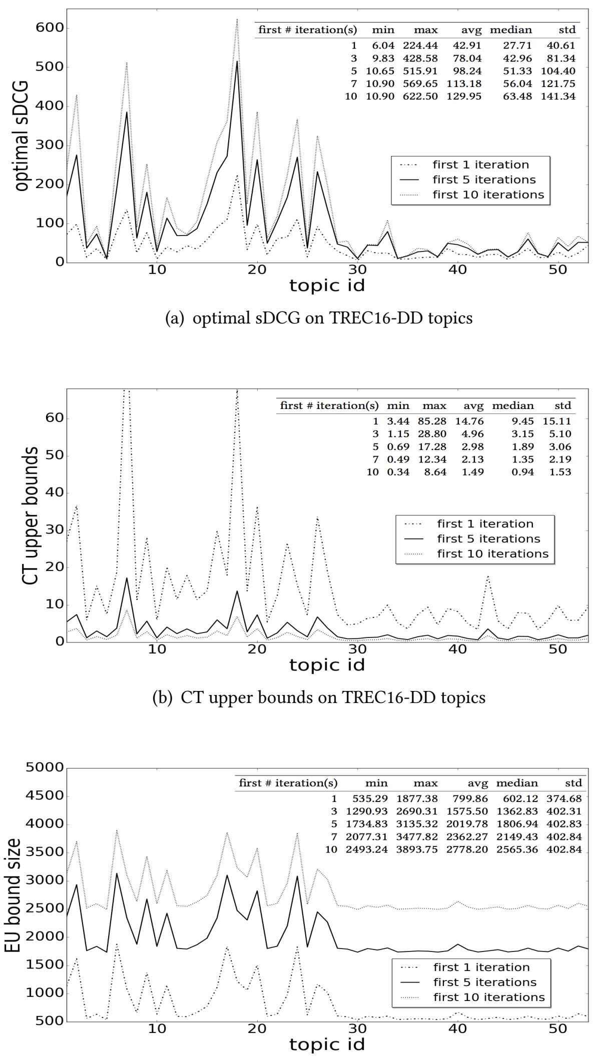
6.1 Plotting the Bounds

Figures 1(a), 1(b) and 1(c) plot the upper bounds/ bound sizes on different topics for sDCG, Cube Test and Expected Utility respectively at different session lengths. Within each figure, we also show the statistics for the corresponding metric in a table. The statistics include the min, max, average, median and standard deviation of the bounds. The bounds are computed as in Section 5. The runs we studied are the 21 official runs submitted to TREC 2016 DD Track.

We observe that differences among the bounds of a metric across different topics are huge and non-negligible for all three session search metrics. This conclusion is true regardless of how many search iterations having been conducted. It suggests that without proper score normalization, it would be unfair to compare across different topics. A metric score averaged for all topics would be biased towards topics that have higher bounds.

For a particular search topic, as more search iterations are conducted, we observe that the metrics' upper bound/bound size changes. They change differently for the three metrics. The optimal sDCG score increases as the search goes on with more iterations. It is because sDCG is essentially a gain function and it ignores the cost of search. Therefore, more search iterations would always increase chances of getting more relevant documents. On the other hand, the upper bound of CT decreases as the search keeps going. It suggests that the rate of gaining relevant information is decreasing as more iterations are used. It makes sense since once a user has learned relevant information in the initial runs, (s)he will not be too surprised for more relevant information at the later iterations thus the gains from those later iterations decrease. However, this observation might also only be related to the TREC 2016 DD dataset and tasks, where the search topics are mostly factual and informational. For navigational or learning-intensive search topics, we might be able to observe a different learning rate. Lastly, we observe that the bound size of EU enlarges as the session length increases. It comes from the fact that EU is a "net gain" function, more iterations mean more possible gain as well as more possible cost.

[1] Some documents are relevant to multiple topics



first # iteration(s)	min	max	avg	median	std
1	6.04	224.44	42.91	27.71	40.61
3	9.83	428.58	78.04	42.96	81.34
5	10.65	515.91	98.24	51.33	104.40
7	10.90	569.65	113.18	56.04	121.75
10	10.90	622.50	129.95	63.48	141.34

(a) optimal sDCG on TREC16-DD topics

first # iteration(s)	min	max	avg	median	std
1	3.44	85.28	14.76	9.45	15.11
3	1.15	28.80	4.96	3.15	5.10
5	0.69	17.28	2.98	1.89	3.06
7	0.49	12.34	2.13	1.35	2.19
10	0.34	8.64	1.49	0.94	1.53

(b) CT upper bounds on TREC16-DD topics

first # iteration(s)	min	max	avg	median	std
1	535.29	1877.38	799.86	602.12	374.68
3	1290.93	2690.31	1575.50	1362.83	402.31
5	1734.83	3135.32	2019.78	1806.94	402.83
7	2077.31	3477.82	2362.27	2149.43	402.84
10	2493.24	3893.75	2778.20	2565.36	402.84

(c) EU bound size on TREC16-DD topics

Figure 1: Upper bounds/ bound sizes on TREC16-DD topics

Moreover, as more iterations conducted in a session, the differences of the metric bounds across topics also change in various ways. For sDCG, the differences among topics become bigger. As a result, the optimal sDCG on different topics become more polarized as the number of iterations increases. For CT, this difference reduces a bit which indicates the system performance gets similar among the topics when more iterations are used. But the differences are still huge and cannot be neglected. For EU, the differences among topics remain relatively the same as the session develops.

Regardless of the changes on the upper bounds/bound sizes as more iterations are conducted, the difference of the optimal value a metric would produce for different topics is large and should not be ignored. We recommend taking per topic bounds into account for fairer evaluation.

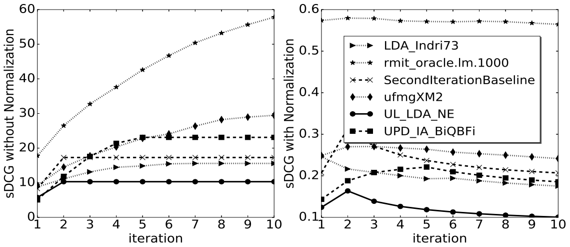


Figure 2: sDCG on TREC16-DD

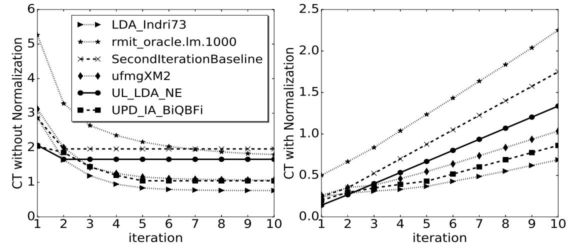
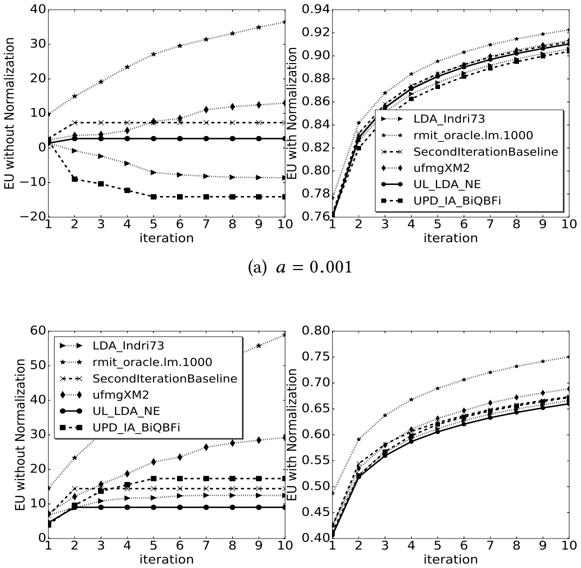


Figure 3: CT on TREC16-DD



(a) $a = 0.001$

(b) $a = 0.0001$

Figure 4: EU on TREC16-DD

6.2 Normalizing Effect

Twenty-one runs from six groups are submitted to TREC 2016 DD Track. We select six representative runs, one from each team, to examine the effects of score normalization. Figures 2, 3 and 4 plot the raw scores without and with normalization. The scores are averaged across all topics for selected runs.

Figure 2 compares the raw (left) and the normalized (right) sDCG scores. The raw sDCG scores are non-decreasing as the number of iterations increases. It is because sDCG is essentially cumulated gains. However, the normalized sDCG scores could increase, decrease or remain as a constant as the number of iterations increases. It suggests that some systems actually have been close to the optimal sDCG score at certain points whereas others do not perform that well. We notice that systems such as $LDA\_Indri73$ and UL_LDA_NE, whose raw sDCG values increase (non-decrease) in Figure 2 (left) but decreases after being normalized (Figure 2 (right)). It suggests that even though the raw gains keep increasing, they increase at a much slower rate than that of the optimal sDCG increases.

Figure 3 compares the raw (left) and the normalized (right) CT scores. Most raw CT scores are decreasing, suggesting that as more iterations are involved, the rate of getting relevant information by those systems reduces. Some raw CT scores remain the same because their systems stop the search after a certain number of iterations. Based on the definition of CT in formula 2, after a system stops, its gain and cost in the following iterations are all zero therefore its CT score won't be affected. On the other hand, all the normalized CT scores increase as the session develops. It is yielded from the sharply declining upper bounds of CT. We even see some normalized CT scores increase so much that they become greater than 1. The normalized CT scores actually imply that, in session search, choosing the right time to stop the search can help an IR systems maintain its efficiency so as to improve users' satisfaction.

Figure 4 compares the raw (left) and the normalized (right) EU scores. As the number of iterations increases, the raw EU scores may increase (Figure 4(a)) or decrease (Figure 4(b)) while the normalized EU scores always increase. We realize that whether the raw EU score would increase or decrease is highly influenced by the choice of the parameter a in formula 3. With different parameter settings, the raw EU curves for the same run could be completely different. It confirms our conclusion that it is not appropriate to directly add or subtract gain by cost. On the other hand, the shape of the normalized EU score is not sensitive with different parameter settings. As a result, the normalization based on the metric bounds yields a more robust metric. Meanwhile, the increase of normalized EU score suggests an enlarged gap between the raw score and the metric's lower bound while the change on the difference between the raw score and upper bound may vary. Nonetheless, the enlarged gap suggests that all the systems have at least moved away from the worst case (the lower bounds).

7 CONCLUSIONS AND DISCUSSION

Session search brings rich interactions between the user and the search system. The evaluation of session search encompasses many factors in search, such as relevance, novelty, and user's effort. As more factors are included into the evaluation, many metrics for session search evaluation become NP-Hard. In this paper, we deconstruct those metrics, compute the optimal scores and use them for score normalization. Through experimenting on the TREC 2016 DD Track, we observe that (i) the variation of bounds of session search metrics on different topics is big and cannot be ignored; (ii) Using the bounds for normalization of those complex session metrics can bring in more fairness into evaluating across topics and yield more reliable evaluation. In addition, the upper bounds of the metrics could potentially be used as optimizing criteria for search systems to decide when to stop the search. Suppose a system has completed k iterations, by giving the upper bound of the first $k + 1$ iterations and computing its actual score in the first k iterations, the system could potentially make a better judgment on whether to continue or stop searching.

8 ACKNOWLEDGEMENTS

This research was supported by NSF grant IIS-145374 and DARPA FA8750-14-2-0226. Any opinions, findings, conclusions, or recommendations expressed in this paper are of the authors, and do not necessarily reflect those of the sponsor.

REFERENCES

[1] Azzah Al-Maskari and Mark Sanderson. 2010. A review of factors influencing user satisfaction in information retrieval. *Journal of the American Society for Information Science and Technology* 61, 5 (2010), 859–868.

[2] Jean Baptiste Joseph Baron Fourier. 1878. *The analytical theory of heat.* The University Press.

[3] Ben Carterette. 2009. An analysis of NP-completeness in novelty and diversity ranking. In *ICTIR'2009.* Springer, 200–211.

[4] Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *SIGIR'2008.* ACM, 659–666.

[5] Godfrey Harold Hardy, John Edensor Littlewood, and George Pólya. 1952. *Inequalities.* Cambridge university press.

[6] Scott B Huffman and Michael Hochster. 2007. How well does result relevance predict session satisfaction?. In *SIGIR'2007.* ACM, 567–574.

[7] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[8] Kalervo Järvelin, Susan L Price, Lois ML Delcambre, and Marianne Lykke Nielsen. 2008. Discounted cumulated gain based evaluation of multiple-query IR sessions. In *ECIR'2008.* Springer, 4–15.

[9] Evangelos Kanoulas, Ben Carterette, Paul D Clough, and Mark Sanderson. 2011. Evaluating multi-query sessions. In *SIGIR'2011.* ACM, 1053–1062.

[10] Tomoharu Kokubu, Tetsuya Sakai, Yoshimi Saito, Hideki Tsutsui, Toshihiko Manabe, Makoto Koyama, and Hiroko Fujii. 2005. The Relationship between Answer Ranking and User Satisfaction in a Question Answering System.. In *NTCIR'2005.*

[11] Chung Tong Lee, Vishwa Vinay, Eduarda Mendes Rodrigues, Gabriella Kazai, Nataša Milic-Frayling, and Aleksandar Ignjatovic. 2009. Measuring system performance and topic discernment using generalized adaptive-weight mean. In *CIKM'2009.* ACM, 2033–2036.

[12] Jiyun Luo, Christopher Wing, Hui Yang, and Marti Hearst. 2013. The water filling model and the cube test: multi-dimensional evaluation for professional search. In *CIKM'2013.* ACM, 709–714.

[13] Stephen Robertson. 2006. On GMAP: and other transformations. In *CIKM'2006.* ACM, 78–83.

[14] Stephen E Robertson. 1977. The probability ranking principle in IR. *Journal of documentation* 33, 4 (1977), 294–304.

[15] Tetsuya Sakai. 2016. Simple and Effective Approach to Score Standardisation. In *ICTIR'2016.* ACM, 95–104.

[16] Mark D Smucker and Charles LA Clarke. 2012. Time-based calibration of effectiveness measures. In *SIGIR'2012.* ACM, 95–104.

[17] William Webber, Alistair Moffat, and Justin Zobel. 2008. Score standardization for inter-collection comparison of retrieval systems. In *SIGIR'2008.* ACM, 51–58.

[18] Grace Hui Yang and Ian Soboroff. 2016. TREC 2016 Dynamic Domain Track Overview. (2016).

[19] Yiming Yang and Abhimanyu Lad. 2009. Modeling expected utility of multi-session information distillation. In *ICTIR'2009.* Springer, 164–175.

[20] Mihalis Yannakakis and Fanica Gavril. 1980. Edge dominating sets in graphs. *SIAM J. Appl. Math.* 38, 3 (1980), 364–372.

[21] Emine Yilmaz, Manisha Verma, Nick Craswell, Filip Radlinski, and Peter Bailey. 2014. Relevance and effort: an analysis of document utility. In *CIKM'2014.* ACM.

[22] Cheng Xiang Zhai, William W Cohen, and John Lafferty. 2003. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR'2003.* ACM, 10–17.