Information Processing and Management 60 (2023) 103404

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Joint upper & expected value normalization for evaluation of retrieval systems: A case study with Learning-to-Rank methods

Dongji Feng, Shubhra Kanti Karmaker (Santu) *

Department of Computer Science and Software Engineering, Auburn University, Auburn, 36830, AL, USA

ARTICLE INFO

Keywords:
Information retrieval evaluation
Upper expected value
Normalization
Learning to Rank

ABSTRACT

While original IR evaluation metrics are normalized in terms of their upper bounds based on an ideal ranked list, a corresponding expected value normalization for them has not yet been studied. We present a framework with both upper and expected value normalization, where the expected value is estimated from a randomized ranking of the corresponding documents present in the evaluation set. We next conducted two case studies by instantiating the new framework for two popular IR evaluation metrics (e.g., $nDCG$, MAP) and then comparing them against the traditional metrics.

Experiments on two Learning-to-Rank (LETOR) benchmark data sets, MSLR-WEB30K (includes 30K queries and 3771K documents) and MQ2007 (includes 1700 queries and 60K documents), with eight LETOR methods (pairwise & listwise), demonstrate the following properties of the new expected value normalized metric: (1) Statistically significant differences (between two methods) in terms of original metric no longer remain statistically significant in terms of Upper Expected(UE) normalized version and vice-versa, especially for *uninformative* query-sets. (2) When compared against the original metric, our proposed UE normalized metrics demonstrate an average of 23% and 19% increase in terms of Discriminatory Power on MSLR-WEB30K and MQ2007 data sets, respectively. We found similar improvements in terms of consistency as well; for example, UE-normalized MAP decreases the swap rate by 28% while comparing across different data sets and 26% across different query sets within the same data set. These findings suggest that the IR community should consider UE normalization seriously when computing *nDCG* and *MAP* and more in-depth study of UE normalization for general IR evaluation is warranted.

1. Introduction

Empirical evaluation is a key challenge for any information retrieval (IR) system. The success of an IR system largely depends on the user's satisfaction, thus an accurate evaluation metric is crucial for measuring the perceived utility of a retrieval system by real users. While original $nDCG$ (Järvelin & Kekäläinen, 2002), MAP (Caragea et al., 2009) etc. metrics are normalized in terms of their query-specific upper bounds based on an ideal ranked list, a corresponding query-specific expected value normalization for them has not yet been studied. For instance, the normalization term in *nDCG* computation is the *Ideal DCG* at cut-off k, which converts the metric into the range between 0 and 1. On the other hand, MAP is normalized by the maximum possible *Sum of Precision* (SP) scores at cut-off k. Thus, *Ideal DCG* and *Sum of Precision* (SP) scores essentially serve as the query-specific upper-bound normalization factor for metric *nDCG* and *MAP*, respectively.

* Corresponding author.
E-mail address: sks0086@auburn.edu (S.K. Karmaker).
URL: https://karmake2.github.io/ (S.K. Karmaker).

https://doi.org/10.1016/j.ipm.2023.103404
Received 22 January 2023; Received in revised form 6 April 2023; Accepted 28 April 2023

Interestingly, the above two popular metrics do not include a similar query-specific expected value normalization factor (the current widely used assumption for expected value is **zero** across all queries). However, each query is different in terms of its difficulty (informative/uninformative/distractive), user's intent (exploratory/navigational), distribution of relevance labels of its associated documents (hard/easy) and user's perceived utility at different cut-off k, essentially implying different expected values for each of them. Therefore, an accurate estimation of an evaluation metric should not only involve an upper-bound normalization (e.g., Ideal DCG, SP, etc.), but also a proper query-specific expected value normalization.

Consider the case of re-ranking where an initial filtering has already been performed given a query and as expected, a large number of associated documents in the filtered set are highly relevant. In this case, even just a random ranking of those documents will yield a high accuracy as most of the documents are highly relevant anyway. This means that even if a ranker does not learn anything meaningful and merely ranks documents randomly, it can still achieve a very high score in terms of the original metric. In other words, the *expected* value of the original metric, in this case, is very high because of the skewed relevance label distribution of the associated documents and this factor should be accounted for while measuring the ranker's quality. In summary, a proper expected value normalization is essential for IR evaluation metrics to accurately measure the quality of a ranker as well as for a fairer comparison across multiple ranking methods.

What does query-specific expected value normalization mean for an IR evaluation metric? How can we come up with a more realistic expected value for each query and include it with the original IR metric computation? One way to address this issue is to introduce a penalty term inside the formula of different IR evaluation metrics which will penalize queries with high expected values of the same metric. In other words, given a query, we propose to use the expected value of the particular evaluation metric as a query-specific expected value of the same metric for that query, which can yield customized expectations for different queries and thus, ensure fairer treatment across all queries with different difficulty levels.

With the observation that both *nDCG* and *MAP* metrics only involve query-specific upper-bound normalization (e.g., normalization with ideal DCG for *nDCG* computation, while MAP is normalized by the maximum possible *Sum of Precision*); none of them include a query-specific expected value normalization. In this paper, we proposed a new general framework for IR evaluation with both upper and expected value normalization and instantiated the new framework for two popular IR evaluation metrics: *nDCG* and *MAP* by computing a more reasonable(non-zero) expected value. Specifically, we introduce two different variants of the framework, i.e., V_1, V_2, which are essentially two different ways to introduce a penalty in terms of normalization with a query-specific upper and expected value of the metric (see Section 5 for more details). We then show how we can compute a more realistic query-specific expected value for the two metrics by computing its expectation for each query in case of a randomized ranking of the corresponding documents, and then, use this expected value as a penalty term while computing the new metric. *The intuition here is that an intelligent ranking method should perform at least as good as a random-ranking algorithm, which naturally inspired us to use the expectation in case of random ranking as our expected value*. Finally, for each metric, we also theoretically prove the correctness of the expected value (Derivation details can be found in each case-study section).

Next, we investigated the implications of upper expected value normalization on the original IR metric. How it may impact IR evaluation in general and more importantly, which metric is better? Why should we care? To answer these questions, we have conducted extensive experiments on two popular Learning-to-Rank (LETOR) data-sets with eight LETOR methods including RankNet (Burges et al., 2005), RankBoost (Freund, Iyer, Schapire, & Singer, 2003), AdaRank (Xu & Li, 2007), Random Forest (Leo, 2001), LambdaMART (Burges, 2010), CoordinateAscent (Metzler & Croft, 2007), ListNet (Cao, Qin, Liu, Tsai, & Li, 2007) and L2 regularized Logistic Regression (Fan, Chang, Hsieh, Wang, & Lin, 2008; Lin, Weng, & Keerthi, 2008). Experimental results demonstrate that a significant portion of the queries in popular benchmark data-sets produced a high expected value normalization factor, verifying that expected value normalization can indeed alter the relative ranking of multiple competing methods (confirmed by Kendall's τ tests Sakai, 2006, 2016) and thus, should not be ignored. At the same time, for a number of closely performing LETOR method-pairs, statistically significant differences in terms of original metric no longer remain statistically significant in terms of expected value normalized metric and vice-versa, especially for *uninformative* query-sets (see Section 4 for a concrete definition), suggesting expected value normalization yields different conclusions than the original metric.

Next, we compare the original metric against the UE normalized version from two perspectives: *Distinguishability* and *Consistency*. In case of discriminative power, we followed Sakai (2006), Sakai et al. (2011) to use student's t-test as well as computed "Percentage Absolute Differences" to quantify distinguishability and found that UE normalized version can better distinguish between two closely performing LETOR methods in case of *uninformative* queries. For consistency, we performed swap rate tests and found that MSP^{UE} provides better performance in terms of *Consistency* while DCG^{UE} does not compromise in terms of *Consistency*.

These findings suggest that the community should rethink about IR evaluation and consider expected value normalization seriously. In summary, we make the following contributions to the paper:

1. We propose an extension of traditional IR evaluation metrics which includes an expected value normalization term, and systematically perform two case-studies by showing how expected value normalization can be materialized for *nDCG* and *MAP*.
2. We propose two different variants of the proposed UE normalized version for two popular IR evaluation metrics.
3. We show how we can compute a more realistic query-specific expected value for two IR evaluation metrics by computing its expectation for each query in case of a randomized ranking of the document collection and also theoretically prove its correctness.
4. We conducted extensive experiments to understand the implications of the expected value normalized metric and compared our proposed metric against the original metric from two important perspectives: *Distinguishability* and *Consistency*.
5. Our proposed framework is very general and can be easily extended to other IR evaluation metrics or evaluation metrics in other domain.

The rest of the paper is organized as follows: Section 2 reviews related works from the past literature. Section 3 provides essential background about our two experimental metric computations for expected value normalization. Section 4 states our research objectives. In Section 5, we first present the framework with query-specific upper and expected value normalization. Section 6 presents the experiment details and results. Finally, Section 7 concludes our paper with discussions and possible future directions.

2. Related work

Traditional IR evaluation metric: Many metrics have been introduced for IR system evaluation in recent years. Two most frequent and basic metrics for the performance evaluation of IR system are *precision* and *recall*, especially for extraction tasks (Karmaker Santu, Sondhi, & Zhai, 2016; Sarkar & Karmaker Santu, 2022). Novel extensions such as Rank-Biased Precision (Amigó, Mizzaro, & Spina, 2022; Moffat & Zobel, 2008) are proposed for solving long (deep) ranking results limitation. Other popular metrics such as MAP (Mean Average Precision) and Normalized Discounted Cumulative Gain ($nDCG$) are also widely used as offline evaluation standards. Different metrics have different hyper-parameters for users to choose from based on their own preferences.

nDCG: *nDCG* is the normalized version of **D**iscounted **C**umulative **G**ain (*DCG*), where the normalization term is essentially a *query-specific* upper-bound (i.e., normalization with *Ideal DCG*), which converts the metric into the range between 0 and 1 (Järvelin & Kekäläinen, 2002). The benefit of *nDCG* is it can be applied to multi-level relevance judgments and also sensitive to small changes in a ranked list. Many researchers have investigated its properties (see, e.g., Ravikumar, Tewari, & Yang, 2011; Wang, Wang, Li, He, & Liu, 2013; Yilmaz, Kanoulas, & Aslam, 2008). The fact that the general concept of $nDCG$ can be implemented in a variety of ways was recognized in the previous work (Kanoulas & Aslam, 2009), where the authors scrutinized how to choose from a variety of discounting functions and different ways of designing the gain function to optimize the efficiency or stability of $nDCG$ (Karmaker Santu, Sondhi, & Zhai, 2017). Previous research has also shown that with different gain functions, $nDCG$ may lead to different results and the discounting coefficients do make a difference in evaluation results as compared to using uniform weights (Voorhees, 2001). Regarding $nDCG$ cutoff-depths, Sakai and others (Sakai, 2007) have researched the reliability of $nDCG$ by establishing that it is highly correlated with average precision if the cutoff-depth k is big enough. According to a recent research (Santu, Sondhi, & Zhai, 2020), conventional $nDCG$ score results in a significant variance in response to the k value and urged for query-specific customization of $nDCG$ to acquire more trustworthy conclusions. Additionally, Gienapp, Stein, Hagen, and Potthast (2020) proposed a measure to explicitly reflect a system's divergence by comparing the query-level $nDCG$ with a randomized ranked $nDCG$, which they called $RNDCG$.

MAP: Average precision (AP) is another popular indicator for evaluating ranked output in IR experiments for a number of reasons as it is already known to be stable (Buckley & Voorhees, 2017) and highly informative measure (Aslam, Yilmaz, & Pavlu, 2005). Whereas Mean Average Precision (*MAP*) (Caragea et al., 2009) is the average AP of each class which can reflect the overall performance among multiple topics. However, the assumption behind *MAP* is retrieved documents can be considered as either relevant or non-relevant to user's information need, which is not accurate. Previous researchers have studied the properties of *MAP* in terms of different relevance judgments. Yilmaz and Aslam (2006), for instance, proposed different variants of AP for addressing incomplete and imperfect relevance judgments, where they consider the document collection is dynamic, as in the case of web retrieval, and they use an expectation of random sample from the depth-100 pool. Furthermore, Robertson, Kanoulas, and Yilmaz (2010) proposed an extended Average Precision named Graded Average Precision (GAP) which can tackle the cases of multi-graded relevance.

Query Specific Customization for General IR Evaluation: Previous work has explored how to incorporate query-specific customization for IR evaluation metrics in general. Recently, Chen, Zhang, and Sakai (2022) proposed a framework for query-level evaluation metrics by incorporating the anchoring effect into the user model and achieved better correlation with user satisfaction. Chen et al. (2021) proposed query reformulation aware metric as query reformulating behaviors may reflect user's search intents. Kuzi, Labhishetty, Karmaker Santu, Joshi, and Zhai (2019) presented a Best-Feature Calibration (BFC) strategy for analyzing learning to rank models and used this strategy to examine the benefit of query-level adaptive training, which demonstrated the importance of query-specific parameters in IR evaluation once again. Moffat, Thomas, and Scholer (2013) followed by Bailey, Moffat, Scholer, and Thomas (2015) argued that user behavior varies on a per-topic basis depending on the nature of the underlying information need, and hence that it is natural to expect that evaluation parameterization should also be variable. Billerbeck et al. studied the optimal number of top-ranked documents that should be used for extraction of terms for expanding a query (Billerbeck & Zobel, 2004). Such work has shown the need to employ a ranking function for each individual query. Egghe (2008) demonstrated precision, recall, fallout and miss as a function of the number of retrieved documents and their mutual interrelations.

IR Evaluation with Variable Parameterization: Query specific customization can be viewed as a special case of variable parameterization for IR evaluation metrics, which has been explored previously. Roitero, Maddalena, Mizzaro, and Scholer (2021) studied the effect of the choice of relevance scales on the evaluation of IR system. Webber, Moffat, and Zobel (2010) explored the role that the metric evaluation depth k plays in affecting metric values and system-versus-system performances for two popular families of IR evaluation metrics: i.e., recall-based and utility-based metrics. Study by Jiang and Allan (2016) showed that the adaptive effort metrics can better indicate user's search experience compared with conventional metrics. Yilmaz, Shokouhi, Craswell, and Robertson (2010) showed users are more likely to click on relevant results and also examined the differences between searcher's effort (dwell

time) and assessor's effort (judging time) on results, and features predicting such effort (Yilmaz, Verma, Craswell, Radlinski, & Bailey, 2014). Sakai and Robertson (2008) modeled a user population to assess the appropriateness of different evaluation metrics.

Distinction from prior work: Our work completely differs from the previous effort as our goal is to investigate the impact of expected value normalization on the prominent evaluation metrics. To the best of our knowledge, there has never been a systematic study of query-specific expected value normalization for IR evaluation metrics. Furthermore, our work is groundbreaking in that it proposes a generic upper expected value normalization framework and effectively applies it to two prominent evaluation metrics. We additionally compute an expectation over a randomized ranked list to estimate a more realistic expected value and also give the derivation. Our research clearly articulates the effects of such expected value normalization on two popular evaluation metrics and lays the foundation for future research in this direction.

3. Revisiting original metrics

In this section, we provide some essential background about $nDCG$ and MAP computation and also provide our motivation of expected value normalization for the two metrics.

3.1. Computation of standard nDCG

The principle behind Normalized Discounted Cumulative Gain ($nDCG$) is that documents appearing lower in a search result list should contribute less than similarly relevant documents that appear higher in the results (Järvelin & Kekäläinen, 2002). This is accomplished by introducing a penalty term that penalizes the gain value logarithmically proportional to the position of the result (Wang et al., 2013). Mathematically:

$$DCG@k = \sum_{i=1}^{k} \frac{2^{R_i} - 1}{\log_b(i+1)} \tag{1}$$

Here, i denotes the position of a document in the search ranked list and R_i is the relevance label of the ith document in the list, cutoff k means DCG accumulated at a particular rank position k, the discounting coefficient is to use a log based discounting factor b to unevenly penalize each position of the search result. $nDCG@k$ is $DCG@k$ divided by maximum achievable $DCG@k$, also called Ideal DCG($IDCG@k$), which is computed from the ideal ranking of the documents with respect to the query.

$$nDCG@k = \frac{DCG@k}{IDCG@k} \tag{2}$$

3.2. Computation of standard MAP

For our second case study, we selected another popular evaluation metric called Mean Average Precision (MAP). In the field of information retrieval, precision is the fraction of retrieved documents that are relevant to the query. The formula is given by: $Prec = TP/(TP + FP)$, where, TP and FP stands for *True Positive* and *False Positive*, respectively. Precision at cutoff k is the precision calculated by only considering the subset of retrieved documents from rank 1 through k. However, the original precision metric is not sensitive to the relative order of the ranked documents, hence, we do not consider it for our exploration.

A related popular metric, which is sensitive to the relative order of the ranked documents, is *Average Precision*, which computes the sum of precision scores at each rank where the corresponding retrieved document is relevant to the query.

$$AP@k = \frac{1}{k} \sum_{i=1}^{k} Prec(i) \cdot R_i \tag{3}$$

Here, R_i is an indicator variable that says whether ith item is relevant ($R_i = 1$) or non-relevant ($R_i = 0$). From Formula (3), we can see $AP@k$ is already normalized by the maximum possible *Sum of Precision* (SP), which is k in this case by assuming a precision value of 1.0 for every position from 1 to k. Thus, $AP@k$ is already upper-bound normalized version of $SP@k$, like $nDCG@k$ is for $DCG@k$. Finally, Mean Average Precision (MAP) of a set of queries is defined by the following formula, where, $|Q|$ is the number of queries in the set and $AP(q)$ is the average precision (AP) for a given query q.

$$MAP = \frac{\sum_{q=1}^{|Q|} AP(q)}{|Q|}$$

In summary, AP is essentially an upper-bound normalized version of *Sum of Precision* (SP), as defined below:

Sum of Precision (SP): *SP computes the summation of the precision scores at all ranks (from 1 to rank k), where the retrieved document is relevant to the query without any upper or expected value bound normalization.*

$$SP@k = \sum_{i=1}^{k} Prec(i) \cdot R_i \tag{4}$$

4. Research objectives

A closer look into the formula of conventional *nDCG* and *MAP* shows that the two metrics incorporate only a query-specific upper-bound normalization (i.e., *IDCG* is actually an upper-bound normalization term). However, as mentioned in Section 1, each query is different in terms of difficulty (hard/easy), informativeness (informative/uninformative/ distractive), user's intent (exploratory/navigational); as such, they have different expected values of different evaluation metrics. Thus, an accurate estimation of average $nDCG$ and MAP should include different expected values for different queries.

Our research objectives stem from this critical observation discussed above. More specifically, how can we develop a more realistic expected value for each query and include it in the original metric computation? What is the effect of query-specific expected value normalization on the IR evaluation metric? These are the research questions we systematically study in this paper. In other words, The main objective of our work is to relax the incorrect assumption of uniform expected values (of $nDCG$ and MAP) across all queries while evaluating IR systems. We propose that an accurate evaluation metric should customize for each query and normalize with respect to both query-specific upper and expected values. A follow-up question that arises immediately is the following: How can we estimate a realistic expected value of an IR evaluation metric? While the original implementation of the above two metrics assumes *zero* as the expected value, previous work proposed to use the worst possible ranking score as the expected value (Gienapp, Fröbe, Hagen, & Potthast, 2020) to achieve a standardized range; we argue that this expected value can be further constrained by using the score of a randomly ranked list for each query. *The justification behind this choice is that a reasonable ranking function should be at least as good as the method that ranks documents merely randomly and should be penalized in cases where it performs worse than random.*

To better motivate UE normalization, we first define the following types of queries, which we will use throughout the rest of the paper:

1. **Informative Queries:** These are queries where a *reasonable* ranking method performs significantly better than a pure random ranking system. Essentially, these are queries which contain the "right" keywords to find out the most relevant documents according to the user's information need. Therefore, the actual evaluation metric scores are much higher than the expected value (the lower triangle region of the plot 1).
   Ideal Queries: These are special cases of *Informative* queries where the difference between actual evaluation metric score and random ranked metric score (expected value) is the largest.
2. **Uninformative Queries:** These are queries where a *reasonable* ranking method performs close to a pure random ranking system. In other words, these are queries which does not offer much value in finding out the most relevant documents. Therefore, the actual evaluation metric scores are similar to the expected value (region around the diagonal line). There are two special cases for Uninformative queries as defined below:

   (a) **Hard Queries**: Hard queries are special cases of *Uninformative* queries, where both *reasonable* ranking methods, as well as pure random ranking systems, demonstrate poor performance. This usually happens in cases where there are no/very few relevant documents in the entire corpus.
   (b) **Easy Queries:** Easy queries are special cases of *Uninformative* queries, where both *reasonable* ranking methods, as well as pure random ranking systems, demonstrate very high performance. This usually happens in cases where there are a lot of relevant documents in the corpus (for example, in case of re-ranking in multi-stage ranking systems Asadi & Lin, 2013; Clarke, Culpepper, & Moffat, 2016; Tonellotto, Macdonald, & Ounis, 2013) and there is little room for improving beyond random ranking.

Fig. 1 shows an illustration of different types of queries with different combinations of evaluation metric expected value and actual metric score. As apparent from Fig. 1, the proposed UE normalization is expected to have a large penalty on uninformative queries including special cases like hard queries (lack of relevant document scenarios) and easy queries (re-ranking scenarios). On the other hand, expected value normalization will have minimal impact in case of *Ideal* queries as the expected value tends to zero and the actual metric score is very high. However, as demonstrated by our experiments, real-world queries are not *Ideal* always and hence, a proper expected value normalization is necessary while computing $nDCG$ and MAP scores because (1) It better captures the difficulty as well as variations across different queries. (2) It makes comparisons and averaging across different queries fairer.

5. IR evaluation with joint upper & expected value normalization

Assume that $A@k$ is the standard evaluation metric and k is the cutoff rank. Before introducing the generic IR evaluation framework with both upper & expected value (UE) normalization, we first define the following terms.

- **IUB[A@k]**: Given a particular query and an associated collection of documents (each with a distinct relevance labels), $IUB[A@k]$ (**I**deal **U**pper **B**ound for $A@k$) is the value that $A@k$ assumes in case of ***perfect*** *ranking* of the document collection.
- **REB[A@k]**: Given a particular query and an associated collection of documents (each with a distinct relevance label), $REB[A@k]$ (**R**andomized **E**xpected **B**ound for $A@k$) is the value that $A@k$ assumes in case of ***random*** ranking ($E[A@k]$) of the document collection.
- **Upper-Bound Normalization**: Given a particular query and an evaluation metric $A@k$, Upper-bound normalization of the metric is defined as $[A@k]^U = \frac{A@k}{IUB[A@k]}$.

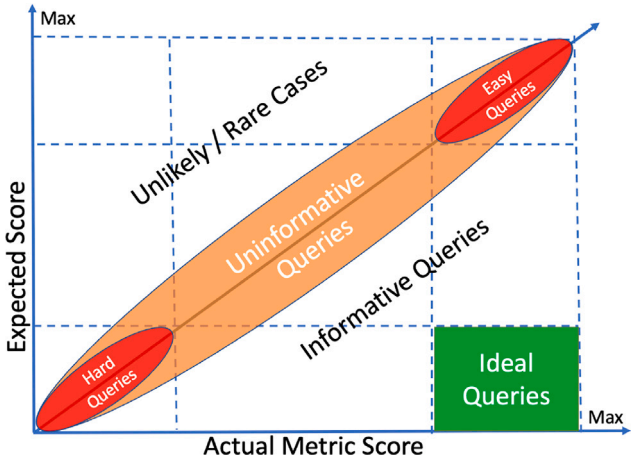

Fig. 1. Query types with different expected values of evaluation metric.

Now, we introduce two different variations of Joint Upper & Expected Value Normalization, which is denoted by, $[A@k]^{UE}$. We call the two versions as V_1, V_2.

$$[A@k]^{UE}_{V_1} = \left(\frac{A@k}{IUB[A@k]}\right)\left(\frac{A@k}{(A@k + REB[A@k])}\right) \tag{5}$$

$$[A@k]^{UE}_{V_2} = \begin{cases} \frac{A@k - REB[A@k]}{IUB[A@k] - REB[A@k]}, & \text{if } A \geq REB \\ \frac{A@k - REB[A@k]}{REB[A@k]}, & \text{otherwise} \end{cases} \tag{6}$$

In the first Eq. (5), we introduce a linear penalty term for Upper Expected Value Normalization while in the second Eq. (6) we introduce a non-linear penalty term. The intuition of the above two Equations is that we want to penalize methods for queries where it performs close to a random ranking method, i.e., the difference between $A@k$ and $REB[A@k]$ is minimal (the uninformative queries): $|A@k - REB[A@k]| \equiv 0$. Even if a ranker achieves high $A@k$ in this case, it does not necessarily mean it is an "intelligent" ranker as the "vanilla" random ranking method can achieve similar performance as well. So, the reward for the method in this case should be discounted. Therefore, to truly distinguish between an "intelligent" and "vanilla" ranking method, it is important to penalize the traditional metric with a more realistic expected value, e.g., score w.r.t. a randomly ranked collection. In other words, for a ranking algorithm to claim a high $A@k$ score, it must perform significantly better than the random ranking baseline.

5.1. Range of expected value normalized metric

It should be noted that V_1 and V_2 are just two different ways to introduce the penalty for higher REB and obviously, more variants are possible while the basic idea remains the same. As can be seen from Eq. (5), V_1 includes an additional multiplicative term that penalizes the original metric with the REB term in the denominator and the range of the metric is still bounded between 0 and 1. V_2 (Eq. (6)) works as follows: instead of range $[0, 1]$, it extends the range from negative to positive real numbers yielding negative numbers for a ranking method which performs worse than the random ranking baseline. **In summary**, for Eq. (5), the range is still $[0, 1]$; while for Eq. (6), the range of the metric is extended from -1 to $+1$ where, $+1$ means perfect ranking, 0 means randomized ranking and -1 means all irrelevant results.

6. Case studies

6.1. Data set

We used two LETOR benchmark data-sets, i.e., "MSLR-WEB30K" (Qin, Liu, Ding, Xu, & Li, 2010) and "MQ2007" (Qin & Liu, 2013) for our experiments. The first and second data-set includes 30,000 and 1700 queries respectively and have widely been used as benchmarks for LETOR tasks (Ganjisaffar, Caruana, & Lopes, 2011; Jia, Wang, Guo, & Wang, 2021; Keshvari, Ensan, & Yazdi, 2022; Shukla, Lease, & Tewari, 2012).

In these data-sets, each row corresponds to a query-document pair. The first column represents the relevance label of the pair, the second column is the query id, and the rest of columns represent features. The relevance scores are represented by an integer scale between 0 to 4 for "MSLR-WEB30K" and between 0 to 2 for "MQ2007", where 0 means non-relevant and 4(2) means highly relevant. The larger the value of relevance label, the more relevant the query-document pair is. Features related to each query-document pair

Table 1
Data sets statistics.

Data set	# Documents	# Queries	# Features
MSLR-WEB30K	~3771 K	31531	136
MQ2007	65323	1692	46

Table 2
Popular learning to rank algorithms.

Algorithm	Short form	Algorithm	Short form
RankNet (Burges et al., 2005)	RNet	LambdaMART (Burges, 2010)	LMART
RankBoost (Freund et al., 2003)	RBoost	CoordinateAscent (Metzler & Croft, 2007)	CA
AdaRank (Xu & Li, 2007)	ARank	ListNet (Cao et al., 2007)	LNet
Random Forest (Leo, 2001)	RF	Logistic Regression (Fan et al., 2008)	L2LR

are represented by a 136 dimensional feature vector for "MSLR-WEB30K" and 46 dimensional feature vector for "MQ2007" data-set (Santu et al., 2020). For more details on how the features were constructed, see Qin and Liu (2013) and Qin, Liu, Ding, et al. (2010). Table 1 shows the number of queries, documents, and features for each data-set (Keshvari et al., 2022). The documents in MQ2007 are retrieved from 25 million pages in the Gov2 web page collection (Qin, Liu, Xu, & Li, 2010) for queries in the million Query track of TREC 2008 while MSLR-web30K is created from a retired labeling set of the Bing search engine.

Both two data-sets come with five folds, where each fold has a test, train, and validation set. We used the train set of each fold for training the models and report the average results across test sets of all folds.

We randomly sampled 10,000 queries from the "MSLR-WEB30K" and 1000 queries from "MQ2007" individually. For "MSLR-WEB30K", the average number of documents associated with each query was 119.06; while for "MQ2007", the number was 41.47. We kept all the features available (136 for "MSLR-WEB30K" and 46 for "MQ2007") for all experiments conducted in this paper.

6.2. Learning to Rank (LETOR) methods

Table 2 contains eight prominent LETOR approaches along with popular classification and regression methods used for ranking applications (Keshvari et al., 2022). We also assign acronyms to each approach for notational convenience, which we will use throughout the rest of the paper.

6.3. Case study 1: nDCG with joint upper expected value normalization

In each case study section, we first describe how to compute a more realistic expected value for the corresponding metric, ($nDCG$ for the first case study) i.e., the expected $nDCG$ in case of random ranking. Although (Gienapp, Stein, et al., 2020) proposed to use the expectation to estimate this value, no derivation process was provided. Note that, $nDCG$ is already an upper-bound normalized version of DCG. Therefore, we start with the original metric $DCG@k$, where, $REB[DCG@k]$ is the expected $DCG@k$ computed based on a randomly ranked list. Thus, we use the terms $E[DCG@k]$ and $REB[DCG@k]$ interchangeably throughout the paper. Additionally, the expected value normalized *nDCG* and upper expected value normalized *DCG* also mean the same thing and we will use them interchangeably throughout the paper as well.

6.3.1. Expected DCG@k

Let R be a random variable denoting the relevance label of a query-document pair and R can assume values from a discrete finite set $\phi = \{0, 1, 2, 3..., r\}$. Also, let the current query be q and the total number of documents that need to be ranked for the current query q is n, let us denote this set by D_q. To derive the formula of $E[DCG@k]$, we start with the definition of expectation in probability theory.

$$E[\text{DCG@k}] = E\left[\sum_{i=1}^{k} \frac{2^{R_i} - 1}{\log_b(i+1)}\right] = \sum_{i=1}^{k} \frac{E\left[2^{R_i} - 1\right]}{\log_b(i+1)}$$

So, the computation of $E[DCG@k]$ is based on the computation of $E[2^{R_i} - 1]$, which is the expected relevance label of the retrieved document at position i. Below we show how to estimate $E[2^{R_i} - 1]$ and first begin with the definition of expectation.

$$E[2^{R_i} - 1] = \sum_{j=0}^{r} (2^j - 1) \cdot Pr(R_i = j)$$

Here, $Pr(R_i = j)$ is the probability that the retrieved document at position i in a randomized ranking would assume a relevance label of j with respect to the current query. Let us assume that n_j is the number of documents with relevance label j, where $j \in \phi$, with respect to the current query. Thus, the constraint $\sum_{j=1}^{r} n_j = n$ holds, where n is the total number of documents in D_q. Thus, $Pr(R_i = j)$ can essentially be computed by counting all the possible rankings which contain a document with relevance label j (with

Table 3
nDCG scores of different LETOR methods for variable k on MSLR-WEB30K data-set.

Method	nDCG@				
	5	10	15	20	30
ARank	0.321	0.349	0.370	0.389	0.423
LNet	0.153	0.182	0.206	0.228	0.268
RBoost	0.306	0.334	0.357	0.377	0.414
RF	0.383	0.411	0.432	0.449	0.479
RNet	0.154	0.183	0.207	0.229	0.269
CA	0.398	0.413	0.428	0.442	0.470
L2LR	0.197	0.237	0.269	0.297	0.344
LMART	0.436	0.454	0.470	0.485	0.513

Table 4
nDCG scores of different LETOR methods for variable k on MQ2007 data-set.

Method	nDCG@				
	5	10	15	20	30
ARank	0.388	0.415	0.448	0.479	0.537
LNet	0.376	0.403	0.438	0.468	0.528
RBoost	0.383	0.414	0.449	0.480	0.535
RF	0.403	0.428	0.460	0.491	0.547
RNet	0.380	0.413	0.445	0.476	0.536
CA	0.392	0.420	0.454	0.482	0.539
L2LR	0.387	0.415	0.447	0.477	0.538
LMART	0.393	0.420	0.453	0.485	0.544

Table 5
Upper & Expected Value Normalized DCG (V_1, V_2) scores of different LETOR methods for variable k: Each cell shows a particular DCG_V^{UE} score with a particular k on MSLR-WEB30K data-set.

Method	$DCG_{V_1}^{UE}$@					$DCG_{V_2}^{UE}$@				
	5	10	15	20	30	5	10	15	20	30
ARank	0.249	0.261	0.271	0.281	0.299	0.237	0.253	0.264	0.276	0.296
LNet	0.097	0.112	0.125	0.138	0.161	0.046	0.060	0.072	0.083	0.104
RBoost	0.232	0.247	0.260	0.270	0.290	0.221	0.237	0.250	0.262	0.285
RF	0.304	0.318	0.328	0.336	0.350	0.308	0.326	0.338	0.347	0.365
RNet	0.098	0.113	0.126	0.138	0.162	0.047	0.061	0.072	0.084	0.105
CA	0.318	0.320	0.325	0.330	0.342	0.325	0.328	0.334	0.340	0.354
L2LR	0.137	0.160	0.180	0.198	0.227	0.098	0.124	0.147	0.167	0.199
LMART	0.354	0.358	0.364	0.370	0.381	0.367	0.374	0.382	0.390	0.405

respect to the current query) at position i and dividing it by the total number of possible rankings up to position k. Below we show the exact formula which is based on the permutation theory.

$$\boldsymbol{E}[\mathbf{2}_i^{\boldsymbol{R}} - \mathbf{1}] = \sum_{j=0}^{r}(2^j - 1)\cdot\left[\frac{^{n_j}P_1 \cdot ^{n-1}P_{k-1}}{^{n}P_k}\right] = \sum_{j=0}^{r}(2^j - 1)\cdot\left[\frac{\frac{n_j!}{(n_j-1)!}\cdot\frac{(n-1)!}{(n-k)!}}{\frac{n!}{(n-k)!}}\right]$$

$$= \sum_{j=0}^{r}(2^j - 1)\cdot\left(\frac{n_j}{n}\right) = \sum_{j=0}^{r}(2^j - 1)\cdot Pr(R = j) = \boldsymbol{E}[\mathbf{2}^{\boldsymbol{R}} - \mathbf{1}]$$

Note that, $E[2^R - 1]$ is different from $E[2^{R_i} - 1]$ because the former is independent of the position of a document in the ranked list, while latter is dependent. However, the above derivation reveals that $E[2^{R_i} - 1]$ is indeed independent of the position i and equals to $E[2^R - 1]$ for any i. Thus, the final formula for computing $E[DCG@k]$ boils down to the following formula:

$$E[DCG@k] = E[2^R - 1]\cdot\sum_{i=1}^{k}\frac{1}{log_2(i+1)} \tag{7}$$

6.3.2. nDCG case-study observations

This section discusses some observed differences between the original *nDCG* and proposed DCG^{UE}. For deeper analysis, we also created two special sub-sets of queries, i.e., (1) *Uninformative* query-set and (2) *Ideal* query-set, based on how close their average (of eight LETOR methods and five cut-off k) expected *nDCG* is to the average real *nDCG*. To achieve this, we computed both average expected *nDCG* and average real *nDCG* for eight LETOR methods and five different cut-offs. Specifically, we followed the steps

Table 6
Upper & Expected Value Normalized DCG (V_1, V_2,) scores of different LETOR methods for variable k: Each cell shows a particular DCG_V^{UE} score with a particular k on MQ2007 data-set.

Method	$DCG_{V_1}^{UE}$@					$DCG_{V_2}^{UE}$@				
	5	10	15	20	30	5	10	15	20	30
ARank	0.288	0.299	0.315	0.331	0.355	0.134	0.209	0.258	0.299	0.363
LNet	0.277	0.288	0.306	0.321	0.348	0.114	0.187	0.248	0.284	0.345
RBoost	0.282	0.297	0.316	0.331	0.354	0.135	0.206	0.263	0.304	0.363
RF	0.299	0.309	0.326	0.340	0.364	0.168	0.235	0.285	0.322	0.386
RNet	0.279	0.295	0.313	0.327	0.353	0.117	0.204	0.255	0.290	0.357
CA	0.291	0.303	0.320	0.333	0.358	0.151	0.221	0.276	0.302	0.366
L2LR	0.285	0.299	0.315	0.329	0.356	0.133	0.209	0.259	0.300	0.368
LMART	0.290	0.301	0.319	0.335	0.361	0.163	0.231	0.280	0.318	0.380

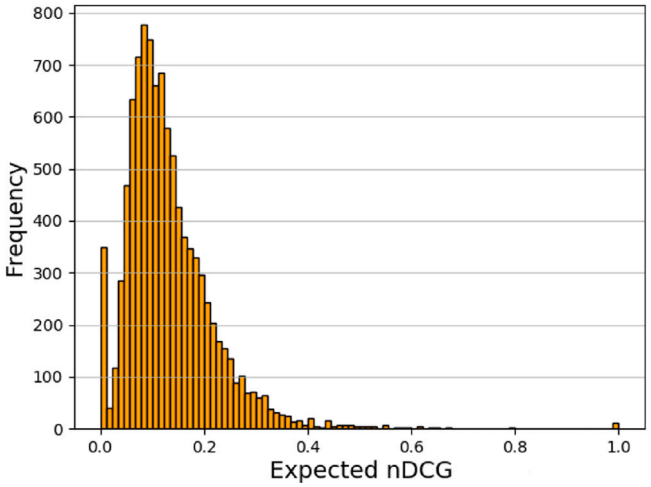

Fig. 2. Histogram of expected $nDCG$ scores of 10,000 queries from the "MSLR-WEB30K" data-set.

from Santu et al. (2020) to compute baseline *nDCG* scores. Tables 3 and 4 summarize the average (original) $nDCG$ scores of different LETOR methods for different values of k, i.e., $k = [5, 10, 15, 20, 30]$ for "MSLR-WEB30K" and "MQ2007" data-sets, respectively. One general observation from Tables 3 and 4 is that average $nDCG@k$ obtained by each method increases as we increase k and the extent of this change is indeed significant. For example, RankNet achieves $nDCG$ value of 0.154 and 0.269 for $k = 5$ and $k = 30$ respectively with an increase of 74.6% (Table 3, "MSLR-WEB30K" data-set).

Next, we computed the expected $nDCG$ score for each query according to Eq. (7). Fig. 2 shows the histogram of expected $nDCG$ scores of 10,000 queries from the "MSLR-WEB30K" data-set. It is interesting to note that, a large portion of "MSLR-WEB30K" queries indeed demonstrates a large variance with high values in the ranges [0.5–0.6]. This justifies our position that expected value for each query can be very different and therefore, expected value normalization should not be ignored while evaluating ranking performances. To further verify this, we calculated the average performances of different LETOR methods in terms of two variants of joint upper-bound and expected-value normalized DCG, i.e. (DCG_V^{UE}), for different values of k on two data-sets (See Tables 5 and 6). Indeed, in "MSLR-WEB10K" data-set (Table 5), both DCG_V^{UE} variants generate significantly smaller values compared with the corresponding original $nDCG$ values (in Table 3). Similarly, in our second data-set "MQ2007", two variants of DCG_V^{UE} achieve lower values (Table 6) compared with corresponding original $nDCG$ values (Table 4).

Subsequently, we created two special sub-sets of queries based on the difference between their expected *nDCG* and the average real *nDCG* obtained by eight LETOR methods, as defined below:

- **Uninformative Query-set:** These are the top 1000 queries among the 10,000 "MSLR-WEB30K" pool (500 in case of MQ-2007 data-set), where difference between the expected *nDCG* and the average real *nDCG* is *minimal*. In other words, these are the top 1000 (500) queries where the LETOR methods struggle to perform better than the random baseline.
- **Ideal Query-set:** These are the top 1000 queries among the 10,000 "MSLR-WEB30K" pool (500 in case of MQ-2007 data-set), where difference between the Expected *nDCG* and the average real *nDCG* is *maximal*. In other words, these are the top 1000 (500) queries where the LETOR methods outperforms the random baseline by the largest margin.

Expected value normalized *nDCG* yields different rankings compare to Original *nDCG* for *Uninformative* query-set: We first test whether our proposed metrics generate different ranking results compared with the original *nDCG* or not. Table 7 shows the Kendall's τ rank correlations between two rankings induced by $nDCG$ and DCG^{UE} scores in *All*, *Uninformative* or *Ideal* query

Table 7
Kendall's τ rank correlations between LETOR method ranks based on $nDCG$ and two DCG^{UE} on *All*, *uninformative* or *ideal* query sets from two data-sets.

Data-set	Kendall's τ			
	Version	All	Uninform.	Ideal
MSLR-WEB30K	nDCG vs V1	1	**0.928**	1
	nDCG vs V2	1	**0.850**	1
MQ2007	nDCG vs V1	1	1	1
	nDCG vs V2	**0.785**	**0.928**	1

Table 8
We used Student's t-test to verify whether statistically significant difference occurred between a pair of LETOR methods while using $nDCG$ and DCG^{UE} and counted the total number of disagreements on *All*, *uninformative* or *ideal* query sets from two data-sets.

Data-set	Conflict cases			
	Version	All	Uninform.	Ideal
MSLR-WEB30K	nDCG vs V1	0	**18**	0
	nDCG vs V2	0	**46**	0
MQ2007	nDCG vs V1	0	**20**	1
	nDCG vs V2	**6**	**24**	8

collections from the two data-sets. We can notice that for both data-sets, $DCG^{UE}_{V_2}$ and $nDCG$ generate different rankings for Uninformative queries resulting the Kendall's τ less than 1 (i.e. 0.85 and 0.928). While for $DCG^{UE}_{V_1}$, it generates different rankings for Uninformative queries in 'MSLR-WEB30K" but not in "MQ2007". Also, as expected in case of Ideal collections, there was no difference between $nDCG$ and DCG^{UE} in both data-sets (Kendall's τ is 1). Another interesting observation is while we use all query collections, only $DCG^{UE}_{V_2}$ generate different ranking results in case of "MQ2007".

Statistical Significance Test Yields Different Outcomes for Original *nDCG* Vs Expected value normalized *nDCG*: Next we conducted statistical significance tests for every pair of LETOR methods based on their original $nDCG$ and DCG^{UE} scores to see how many times the two metrics disagree on the relative performance between two competing LETOR methods. Specifically, we followed the *bootstrap* Studentised Test (student's t-test) from Sakai (2006) to verify whether the observed difference has occurred due to mere random fluctuations or not for each pair of LETOR methods. Using the most widely used confidence value of 0.05 as the threshold, a p-value larger than 0.05 means the two distributions are statistically same, otherwise the pair of distributions are statistically different. More specifically, we compared each pair of LETOR methods (${}^8C_2 = 28$ pairs in total) with respect to five cut-off k, i.e., $k = [5, 10, 15, 20, 30]$. Thus, the total number of comparisons is $28 \times 5 = 140$.

Table 8 summarizes the number of disagreements between $nDCG$ and DCG^{UE} in two data-sets. For instance, based on student's t-test, $DCG^{UE}_{V_2}$ disagreed with original $nDCG$ on 46 (32%) pairs of LETOR methods for *Uninformative* query set from "MSLR-WEB30K", while **zero** disagreements for *Ideal* query set. In "MQ2007", we can also observe 24(17%) pairs of disagreements for *Uninformative* query set as well as there are 8 pairs of conflicts in *Ideal* query set. In particular, we also see $DCG^{UE}_{V_2}$ disagreed with original $nDCG$ on 6 pairs for all query set from "MQ2007".

Given the difference in outcomes and disagreements between the original $nDCG$ metric and its expected value normalized version, a natural follow-up question now is: which metric is better? To answer this question, we compared the $nDCG$ and DCG^{UE} metrics in terms of their *Discriminative power* and *Consistency* (Sakai, 2006). These are two popular methods for comparing evaluation measures.

Distinguishability: We first focus on the implication of expected value normalization in terms of its capability to distinguish among multiple competing LETOR method pairs. To quantify distinguishability, we first utilize the *discriminative power*, which is a popular method for comparing evaluation metrics by performing a statistical significance test between each pair of LETOR methods and counting the number of times the test yields a significant difference (Chen et al., 2021; Sakai, 2006; Yu, Jatowt, Blanco, Joho, & Jose, 2017). Note that *discriminative power* is not about whether the metrics are right or wrong: it is about how often differences between methods can be detected with high confidence (Sakai et al., 2011). We again follow Sakai (2006) to use student's t-test to conduct this experiment and again use 0.05 as our threshold. Using the aforementioned *Uninformative* and *Ideal* query collections, Table 9 shows the total number of statistically significant differences that can be detected between pairs of LETOR methods in case of All queries, Uninformative queries and Ideal queries (from both data-sets), individually by the $nDCG$ and two DCG^{UE} metrics.

On *"MSLR-WEB30K" Uninformative* query set, $nDCG$ could detect only 33 (23%) significantly different pairs. In contrast, both two proposed $DCG^{UE}_{V_1}$ and $DCG^{UE}_{V_2}$ can detect more cases of significant differences. Additionally, $DCG^{UE}_{V_2}$ achieve the best performance which detected **78** (55%) significantly different pairs on the same set. On the other hand, on *"MSLR-WEB30K" Ideal* query-set, both $nDCG$ and two DCG^{UE} detected **130** significantly different pairs. It is evident that, both two DCG^{UE} can better distinguish between two LETOR methods than $nDCG$ on "MSLR-WEB30K" data-set, while not compromising distinguishability in case of *Ideal* queries, which is desired. We also observed similar improvements by DCG^{UE} in case of "MQ2007" data-set. More importantly, DCG^{UE} not only improves the distinguishability in case of *uninformative* query set, it can also detect more different cases while using *All* query set (for $DCG^{UE}_{V_2}$) and *Ideal* query set (for both DCG^{UE}), which is a bonus.

Table 9
Student T-test induced total number of statistically significant differences detected based on $nDCG$ and DCG^{UE} on *All*, *uninformative* or *ideal* query sets from two data-sets.

Data-set	Number of Stat-Sig difference			
	Version	All	Uniform.	Ideal
MSLR-WEB30K	nDCG	133	33	130
	V1	133	**51**	130
	V2	133	**78**	130
MQ2007	nDCG	0	9	7
	V1	0	**29**	**8**
	V2	**6**	**33**	**15**

Table 10
Percentage Absolute Difference between pairs of LETOR methods in terms of average $nDCG$ and DCG^{UE} scores on *All*, *uninformative* or *ideal* query sets from two data-sets.

Metrics	PAD score					
	All query		Uninformative		Ideal	
	MSLR	MQ2007	MSLR	MQ2007	MSLR	MQ2007
nDCG	31.000	1.740	7.390	5.850	35.740	1.610
$DCG^{UE}_{V_1}$	**35.700**	**3.600**	**9.980**	**7.825**	**40.210**	**1.980**
$DCG^{UE}_{V_2}$	**46.700**	**6.420**	**41.750**	**44.810**	**44.530**	**2.980**

We also computed another metric to quantify distinguishability: *Percentage Absolute Differences (PAD)*. More specifically, we computed the percentage absolute differences between pairs of LETOR methods in terms of their original *nDCG* and DCG^{UE} scores, separately. The intuition here is that metrics with higher distinguishability will result in higher percentage of absolute differences between pairs of LETOR methods. To elaborate, we first calculated the average value of both $nDCG$ and DCG^{UE} with varying k ($k = \{5, 10, 15, 20, 30\}$) for each LETOR method and then, computed the percentage absolute difference between each pair of LETOR methods in terms of those two metrics separately (one percentage for $nDCG$ and another for DCG^{UE}), then we calculated the average of those percentage absolute differences. This experiment was performed on both data-sets. Mathematically, we used the following formula for percentage absolute differences (PAD) in terms of original $nDCG$:

$$PAD(nDCG) = \frac{|nDCG^{avg}_{M_1} - nDCG^{avg}_{M_2}|}{\max\left(nDCG^{avg}_{M_1}, nDCG^{avg}_{M_2}\right)} \times 100\% \tag{8}$$

Here, M_1 and M_2 are two different LETOR methods and $nDCG^{avg}_{M_1}$ is the average $nDCG$ score obtained by method M_1 with respect to varying k. The equation for $PAD(DCG^{UE})$ is similar and thus omitted. Besides, we use this equation for the PAD calculation of our second case-study. Table 10 shows these average percentage absolute differences of all possible LETOR method pairs in terms of original $nDCG$ and DCG^{UE} scores on our two data-sets.

From this table, we can observe that while using DCG^{UE}, the PAD score of DCG^{UE} is higher than the same for original $nDCG$ for all types of query collections, i.e., using *All* queries, *Uninformative* and *Ideal* query sub-sets. For instance, the average PAD of $nDCG$ on "MQ2007" is **1.74**; while for $DCG^{UE}_{V_2}$, the score is **6.42** (using all query). Similarly, we discovered that for *Uninformative* query-set, DCG^{UE} achieves a significant boost compared to the same in *Ideal* query-set in both data-sets.

These results show that the proposed UE normalization enhances the distinguishability of the original nDCG metric and can differentiate between two competing LETOR methods with a larger margin, which is a nice property of UE normalization.

Consistency: This experiment focuses to compare the relative ranking of LETOR methods in terms of their $nDCG$ and DCG^{UE} scores, separately, *across* different data-sets ("MQ2007" Vs "MSLR-WEB30K") as well as *across Uninformative* and *Ideal* query collections within the same data-set. The goal here is to see which metric yields a more stable ranking of LETOR methods across various types of documents and queries as well as across diverse sets of data-sets. We computed *swap rate* (Sakai, 2006) to quantify the consistency of rankings induced by $nDCG$ and DCG^{UE} metrics across different data-sets. The essence of swap rate is to investigate the probability of the event that two experiments are contradictory given an overall performance difference.

Table 11 shows our swap rate results for $nDCG$ and DCG^{UE} across the two data-sets, "MSLR-WEB30K" and "MQ2007". Note that in our original setup, we selected Uninformative/ Ideal 1000 queries from "MSLR-WEB30K". To make our results comparable, in this experiment we select 500 Uninformative/Ideal queries from "MSLR-WEB30K" and compare the ranking result with the one from "MQ2007". It can be observed that both $nDCG$ and DCG^{UE} share an identical swap rate probability when we conduct the experiment on the All/Uninformative/Ideal query collection (swap rate across data-sets is 0.107, 0.42 and 0.35 for both metrics).

Table 12 also shows our swap rate results for $nDCG$ and DCG^{UE} across *Uninformative* Vs *Ideal* queries from the same data-set. We can still observe that both $nDCG$ and DCG^{UE} generate the identical swap rate probability when we compare the ranking results across *Uninformative* and *Ideal* sets, except for $DCG^{UE}_{V_1}$ (generate a higher swap rate in "MSLR-WEB30K").

Table 11
Swap rates between method ranks on *All/uniform/Ideal* queries across "MSLR-WEB30K" and "MQ2007" data-sets.

Metric	Swap rate		
	All	Uninform.	Ideal
nDCG	0.107	0.420	0.350
$DCG^{UE}_{V_1}$	0.107	0.420	0.350
$DCG^{UE}_{V_2}$	0.107	0.420	0.350

Table 12
Swap rates between method ranks on *MSLR-WEB30K/MQ2007* data-sets across "uninformative" and "Ideal" query collections.

Metric	Swap rate	
	MSLR-WEB30K	MQ2007
nDCG	0.210	0.500
$DCG^{UE}_{V_1}$	0.250	0.500
$DCG^{UE}_{V_2}$	0.210	0.500

Table 13
Swap rates between method ranks on *MSLR-WEB30K* data-sets across "broad" and "focused" query collections.

Metric	Swap rate
	MSLR-WEB30K
nDCG	0.250
$DCG^{UE}_{V_1}$	0.250
$DCG^{UE}_{V_2}$	0.250

Alternative Query and Document Partitioning: To further test the stability of the proposed UE normalization technique across different sets of queries and documents, we conducted two additional experiments. These experiments are inspired by previous works that have studied robust evaluation of IR systems by randomly partitioning queries and documents (see, e.g., Faggioli, Ferro, & Fuhr, 2022; Moffat, Scholer, & Thomas, 2012; Voorhees, Samarov, & Soboroff, 2017); we present the corresponding experiment details and results below.

In the first experiment, we investigated whether the proposed UE normalization is can be effective for other criteria of defining the "difficulty" of queries (besides our previously defined "Uninformative" and "Ideal" query sets). To achieve this, we borrowed the *threshold-based* strategy proposed by Mothe, Laporte, and Chifu (2019) to define the difficulty of a query. To be more specific, we used the proportion of highly relevant documents (in the evaluation set) as the threshold to partition the original "MSLR-WEB30K" data set into "Broad" and "Focused" query sets. Formally, a query is labeled as "broad" if at least 50% of its associated documents have a relevance label greater or equal to 2 in the testing set. Otherwise, the query is labeled as "focused" because of the few number of relevant documents associated with it. Intuitively, a "broad" query is much easier to rank due to its high proportion of high-relevant documents, whereas, for the exact opposite reason, it is more challenging to rank documents for a "focused" query. We also keep the number of "broad" and "focused" queries balanced in our testing data-set to ensure fairness.

Next, we conducted the same "Consistency" experiments for the nDCG metric. Table 13 concludes the swap rate (consistency) results between method ranks across "broad" and "focused" query sets while using $nDCG$ and DCG^{UE} for the "MSLR-WEB30K" data-set. Interestingly, we still observe that both $nDCG$ and DCG^{UE} generate the identical swap rate probability when we compare the ranking results across *Broad* and *Focused* sets, indicating that our proposed metric does not sacrifice consistency while comparing across different query partitions, where the partitions were created based on query difficulty.

Our second experiment takes a closer look at the consistency property of the UE normalization technique while using replicates, i.e., different document partitions. We followed Voorhees et al. (2017), who proposed an approach to obtain the required replicate measurements by randomly splitting the documents into n partitions and evaluating each of the document set partitions. Due to the relatively low average number of documents (119.06) associated with each query in "MSLR-WEB30K" data set, we divided the documents into just two parts, referred to as the "left" and "right" document sets, using a random split.

Table 14 shows the swap rate (consistency) results between method ranks across "left" and "right" document sets while using $nDCG$ and DCG^{UE} for the "MSLR-WEB30K" data-set. We can observe that both $nDCG$ and DCG^{UE} hold the same ranking while evaluating methods on the "left" and "right" partitions of documents, resulting in the swap rate as 0 for both metrics. This again shows that the proposed UE normalization technique does not reduce the consistency of the original nDCG metric.

Table 14
Swap rates between method ranks on *MSLR-WEB30K* data-sets across "left" and "right" document collections.

Metric	Swap rate
	MSLR-WEB30K
nDCG	0
$DCG^{UE}_{V_1}$	0
$DCG^{UE}_{V_2}$	0

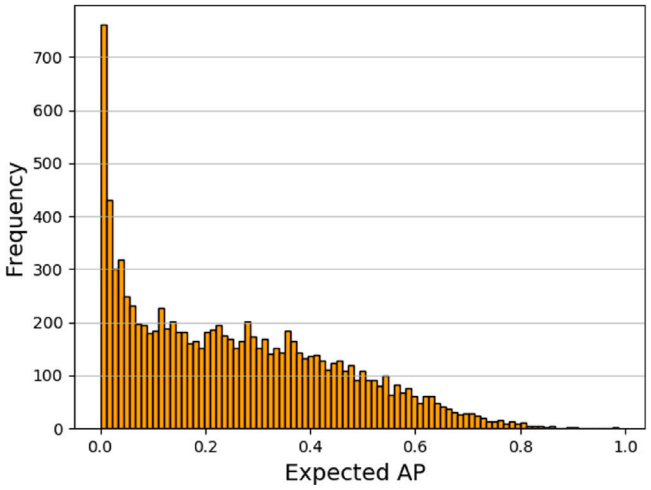

Fig. 3. Histogram of expected AP scores of 10,000 queries from the "MSLR-WEB30K" data-set.

6.4. Case study 2: MAP with joint upper & expected value normalization

For our second case study, we selected another popular evaluation metric called Mean Average Precision (MAP). However, original MAP computation needs binary label while our two data-sets are multi-relevance label. For consistency, in this paper, we only consider 0 relevance score as negative and others are positive for both two data-sets. Tables 15 and 16 show the original MAP scores from two data-sets. Below, we will first present how we can compute a realistic expected value for *Sum Precision* (SP) by computing its expected value in case of a randomly ranked list of documents. Then, demonstrate our findings of expected value normalized MAP. Again, expected value normalized MAP essentially means upper expected value normalized *MSP*.

First, we also show the histogram of expected AP score for 10,000 queries from "MSLR-WEB30K" data-sets. Fig. 3 shows the histogram of expected AP scores of 10,000 queries from the "MSLR-WEB30K" data-set. We can still observe that a large variance of high expected AP appeared in this data-set, indicating that cannot be ignored (also verified by the results in Tables 17 and 18). Noted that we again created two special sub-sets of queries based on the difference between their Expected AP and average real AP obtained by eight LETOR methods to define **Uninformative query-set** and **Ideal query-set** (Details in 6.3.2).

6.4.1. Expected value of SP (SP for random ranking)

Given a query q, assume that N_p is the total number of relevant documents, N_n is the number of non-relevant document for query q. Also, assume $N_p > k$ and $N_n > k$, k is the cutoff variable. $Prec(i)$ is the precision at position i and R_i is the relevance at position i. Then, expectation of $SP@k$ in case of random ranking is the following:

$$E[SP@k] = \sum_{i=1}^{k} E[Prec(i) \cdot R_i]$$

Now assuming $Prec(i)$ and R_i are independent, we have

$$E[SP@k] = \sum_{i=1}^{k} E[Prec(i)] \cdot E[R_i], \text{ where,}$$

$$E[R_i] = P[R_i = 1] \cdot 1 + P[R_i = 0] \cdot 0 = P[R_r = 1] = \frac{N_p}{N_p + N_n}$$

$$E[Prec@i] = \frac{1}{i}\left[P\left(Prec@i=\frac{1}{i}\right)\right] + \frac{2}{i}\left[P\left(Prec@i=\frac{2}{i}\right)\right] + \cdots + \frac{i}{i}\left[P\left(Prec@i=\frac{i}{i}\right)\right]$$
$$= \left(\frac{1}{i}\right)\left[\frac{\binom{N_p}{1}\binom{N_n}{i-1}}{\binom{N_p+N_n}{i}}\right] + \left(\frac{2}{i}\right)\left[\frac{\binom{N_p}{2}\binom{N_n}{i-2}}{\binom{N_p+N_n}{i}}\right] + \cdots + \left(\frac{i}{i}\right)\left[\frac{\binom{N_p}{i}\binom{N_n}{i-i}}{\binom{N_p+N_n}{i}}\right]$$
$$= \left(\frac{1}{i}\right)\frac{1}{\binom{N_p+N_n}{i}}\sum_{j=1}^{i} j\binom{N_p}{j}\binom{N_n}{i-j}$$

We will later prove that, $\sum_{j=1}^{i} j\binom{N_p}{j}\binom{N_n}{i-j} = \frac{N_p}{N_p+N_n} i\binom{N_p+N_n}{i}$

Thus, $E[Prec@i] = \frac{N_p}{N_p+N_n}$, Hence:

$$E[SP@k] = \sum_{i=1}^{k} E[Prec(i)] \cdot E[R_i] = \sum_{i=1}^{k}\left(\frac{N_p}{N_p+N_n}\right)^2 = k\left(\frac{N_p}{N_P+N_n}\right)^2$$

Now, we will use induction to prove the following:

$$\sum_{j=1}^{i} j\binom{N_p}{j}\binom{N_n}{i-j} = \left(\frac{N_p}{N_p+N_n}\right) i\binom{N_p+N_n}{i} \tag{9}$$

Base case: For i = 1, L.H.S $= 1\binom{N_p}{1}\binom{N_n}{1-1} = N_p$

$$R.H.S = \left(\frac{N_p}{N_p+N_n}\right) 1\binom{N_p+N_n}{1} = \frac{N_p}{N_p+N_n}\left(N_p+N_n\right) = N_p$$

So, Eq. (9) is true for $i = 1$

Induction step: Now, Let us assume Eq. (9) is true for $i = i$-1, then we get the following:

$$\sum_{j=1}^{i-1} j\binom{N_p}{j}\binom{N_n}{i-1-j} = \frac{N_p}{N_p+N_n}(i-1)\binom{N_p+N_n}{i-1} \tag{10}$$

Now, $\sum_{j=1}^{i} \boldsymbol{j\binom{N_p}{j}\binom{N_n}{i-j}} = \sum_{j=1}^{i-1} j\binom{N_p}{j}\binom{N_n}{i-j} + i\binom{N_p}{i} = \sum_{j=1}^{i-1} j\binom{N_p}{j}[\binom{N_n+1}{i-j} - \binom{N_n}{i-j-1}] + i\binom{N_p}{i}$

$$= \left[\sum_{j=1}^{i-1} j\binom{N_p}{j}\binom{N_n+1}{i-j}\right] + i\binom{N_p}{i} - \left[\sum_{j=1}^{i-1} j\binom{N_p}{j}\binom{N_n}{i-j-1}\right]$$
$$= \sum_{j=1}^{i} j\binom{N_p}{j}\binom{N_n+1}{i-j} - \left(\frac{N_p}{N_p+N_n}\right)(i-1)\binom{N_p+N_n}{i-1} \qquad \text{[From (10)]}$$
$$= \sum_{j=1}^{i} N_p\binom{N_p-1}{j-1}\binom{N_n+1}{i-j} - \left(\frac{N_p}{N_p+N_n}\right)(i-1)\binom{N_p+N_n}{i-1} \qquad As, \left[j\binom{N_p}{i} = N_p\binom{N_p-1}{j-1}\right]$$
$$= N_p\sum_{j=1}^{i}\binom{N_p-1}{j-1}\binom{N_n+1}{i-j} - \left(\frac{N_p}{N_p+N_n}\right)(i-1)\binom{N_p+N_n}{i-1}$$
$$= N_p\binom{N_p+N_n}{i-1} - \left(\frac{N_p}{N_p+N_n}\right)(i-1)\binom{N_p+N_n}{i-1}$$
$$= \binom{N_p+N_n}{i-1}\left(\frac{N_p}{N_P+N_n}\right)[N_p+N_n-i+1] = \left[(N_p+N_n-i+1)\binom{N_p+N_n}{i-1}\right]\left(\frac{N_p}{N_p+N_n}\right)$$
$$= \boldsymbol{i\binom{N_p+N_n}{i}\left(\frac{N_p}{N_p+N_n}\right)}$$

Proof completed because, $(n-r+1)\binom{n}{r-1} = r\binom{n}{r}$

Expected value normalized *MAP* yields different rankings compare to Original MAP for *Uninformative* query-set: Table 19 shows the Kendall's τ rank correlations between two rankings induced by MAP and MSP^{UE} scores in *All*, *Uninformative* or *Ideal* query collections for the two data-sets. Firstly, we can notice that for both data-sets, $MSP^{UE}_{V_1}$ and MAP generate identical rankings for different query set which indicate that there is no difference between MAP with $MSP^{UE}_{V_1}$ in terms of Kendall's τ rank test. While for $MSP^{UE}_{V_2}$, it generates different rankings for all kinds of query collections in both two data-sets. For instance, in "MQ2007", Kendall's τ correlation between MAP and $MSP^{UE}_{V_2}$ are **0.785**, **0.624** and **1** for *all*, *uninformative* and *ideal* query set, suggesting that $MSP^{UE}_{V_2}$ achieves different outcomes. In addition, the impact is more prominent in case of *uninformative* compared with *ideal*.

Table 15
MAP scores of different LETOR methods for variable k on 'MSLR-WEB30K' dataset.

Method	MAP@				
	5	10	15	20	30
ARank	0.541	0.494	0.472	0.459	0.449
LNet	0.320	0.299	0.293	0.291	0.294
RBoost	0.544	0.496	0.475	0.461	0.452
RF	0.621	0.571	0.543	0.524	0.505
RNet	0.321	0.300	0.293	0.291	0.295
CA	0.623	0.563	0.530	0.510	0.490
L2LR	0.356	0.335	0.333	0.335	0.345
LMART	0.648	0.592	0.561	0.541	0.519

Table 16
MAP scores of different LETOR methods for variable k on 'MQ2007' dataset.

Method	MAP@				
	5	10	15	20	30
ARank	0.306	0.292	0.302	0.317	0.362
LNet	0.337	0.323	0.332	0.346	0.390
RBoost	0.346	0.336	0.347	0.363	0.403
RF	0.367	0.352	0.358	0.373	0.414
RNet	0.328	0.317	0.327	0.344	0.387
CA	0.359	0.345	0.356	0.371	0.412
L2LR	0.354	0.338	0.345	0.360	0.404
LMART	0.358	0.345	0.353	0.369	0.410

Table 17
Upper & Expected Value Normalized MSP (V_1, V_2) scores of different LETOR methods for variable k: Each cell shows a particular MSP_V^{UE} score with a particular k. MSLR-WEB30K dataset.

Method	$MSP_{V_1}^{UE}$@					$MSP_{V_2}^{UE}$@				
	5	10	15	20	30	5	10	15	20	30
ARank	0.385	0.338	0.315	0.301	0.286	0.347	0.305	0.279	0.261	0.237
LNet	0.197	0.173	0.164	0.159	0.157	−0.072	−0.057	−0.050	−0.045	−0.038
RBoost	0.390	0.342	0.319	0.305	0.290	0.350	0.301	0.275	0.256	0.233
RF	0.457	0.407	0.379	0.359	0.336	0.478	0.427	0.390	0.360	0.322
RNet	0.198	0.174	0.165	0.160	0.158	−0.071	−0.055	−0.049	−0.045	−0.038
CA	0.459	0.400	0.367	0.347	0.323	0.483	0.412	0.367	0.338	0.297
L2LR	0.226	0.201	0.196	0.195	0.198	−0.022	−0.004	0.014	0.031	0.055
LMART	0.482	0.426	0.394	0.374	0.348	0.525	0.463	0.421	0.389	0.346

Table 18
Upper & Expected Value Normalized MSP (V_1, V_2) scores of different LETOR methods for variable k: Each cell shows a particular MSP_V^{UE} score with a particular k. MQ2007 dataset.

Method	$MSP_{V_1}^{UE}$@					$MSP_{V_2}^{UE}$@				
	5	10	15	20	30	5	10	15	20	30
ARank	0.236	0.219	0.222	0.228	0.247	0.039	0.077	0.011	0.140	0.190
LNet	0.267	0.249	0.251	0.257	0.274	0.090	0.131	0.163	0.184	0.225
RBoost	0.273	0.260	0.264	0.271	0.285	0.123	0.154	0.188	0.213	0.251
RF	0.291	0.273	0.273	0.280	0.294	0.158	0.190	0.206	0.226	0.272
RNet	0.259	0.244	0.247	0.255	0.272	0.085	0.130	0.156	0.182	0.222
CA	0. 286	0.268	0.272	0.279	0.294	0.142	0.174	0.198	0.220	0.258
L2LR	0.280	0.262	0.263	0.269	0.286	0.123	0.154	0.184	0.209	0.254
LMART	0.282	0.267	0.269	0.275	0.290	0.154	0.194	0.213	0.236	0.272

Statistical Significance Test Yields Different Outcomes for Original *MAP* Vs expected value normalized *MAP*: We again conducted statistical significance tests for every pair of LETOR methods based on their original MAP and MSP^{UE} scores to see how many times the two metrics disagree on the relative performance between two competing LETOR methods. Table 20 summarizes the number of disagreements between MAP and MSP^{UE} in two data-sets. For instance, based on student's t-test, $MSP_{V_2}^{UE}$ disagreed with original MAP on 36 (26%) pairs of LETOR methods for *Uninformative* query set from "MSLR-WEB30K", while **4** disagreements for *Ideal* query set. Although none of MSP^{UE} disagree with original MAP while using *All* query set from "MSLR-WEB30K", there are still 1 and 8 conflicts appeared in "MQ2007" for two UE normalized version respectively.

Table 19
Kendall's τ rank correlations between LETOR method ranks based on MAP and two MSP^{UE} on *All*, *uninformative* or *ideal* query sets from two data-sets.

Data-set	Kendall's τ			
	Version	All	Uninform.	Ideal
MSLR-WEB30K	MAP vs V1	1.000	1.000	1.000
	MAP vs V2	**0.928**	**0.857**	**0.928**
MQ2007	MAP vs V1	1.000	1.000	1.000
	MAP vs V2	**0.785**	**0.624**	1.000

Table 20
We used Student's t-test to verify whether a statistically significant difference occurred between a pair of LETOR methods while using MAP and MSP^{UE} and counted the total number of disagreements on *All*, *uninformative* or *ideal* query sets from two data-sets.

Data-set	Conflict cases			
	Version	All	Uninform.	Ideal
MSLR-WEB30K	MAP vs V1	0	**15**	2
	MAP vs V2	0	**36**	4
MQ2007	MAP vs V1	1	**2**	3
	MAP vs V2	**8**	**21**	17

Table 21
Student T-test induced total number of statistically significant differences detected based on MAP and MSP^{UE} on *All*, *uninformative* or *ideal* query sets from two data-sets.

Data-set	Number of Stat-Sig difference			
	Version	All	Uniform.	Ideal
MSLR-WEB30K	MAP	129	61	122
	V1	129	**76**	124
	V2	129	**81**	122
MQ2007	MAP	45	0	71
	V1	**50**	**2**	**74**
	V2	**59**	**21**	**88**

Given the difference in outcomes and disagreements between the original MAP metric and its expected value normalized version, we still trying to compare these two metrics in terms of their *Discriminative power* and *Consistency* just like what we did in $nDCG$.

6.4.2. Distinguishability

We again follow Sakai (2006) to use student's t-test to conduct this experiment and use 0.05 as our threshold. Using the aforementioned *Uninformative* and *Ideal* query collections, Table 21 shows some interesting results of these statistical tests for different query sets in 'MSLR-WEB10K' and'"MQ2007" data-sets.

On *"MSLR-WEB30K" Uninformative* query set, although MAP detect 61 (43%) significantly different pairs, both two proposed $MSP^{UE}_{V_1}$ and $DCG^{UE}_{V_2}$ can detect more cases of significant differences. What can be clearly seen is $MSP^{UE}_{V_2}$ still achieve the best performance which detected **81** (57%) significantly different pairs on the same set. On the other hand, on *"MSLR-WEB30K" Ideal* query set, both MAP and two MSP^{UE} detected around **122** significantly different pairs. More interestingly, in "MQ2007", while original MAP detect 45 cases of different pairs using all query set, MSP^{UE} indeed improve this performance (for $MSP^{UE}_{V_1}$ is 50 and $MSP^{UE}_{V_2}$ is 59). Specifically in uninformative query set, MAP cannot detect any significantly different pairs. However, $MSP^{UE}_{V_2}$ can detect 21 pairs of difference, which is very important. On the other hand, $MSP^{UE}_{V_2}$ can even detect more cases in the $ideal$ query set. It is evident that both two MSP^{UE} can better distinguish between two LETOR methods than MAP on two data-sets, while not compromising distinguishability in case of *Ideal* queries (even improve the distinguishability in "MQ2007").

Again, we use the formula (8) to compute the percentage of absolute differences between pairs of LETOR methods in terms of their original MAP and MSP^{UE}, separately. Here, **X** represents MAP and $MSP^{UE}_{V_{1,2}}$. (Details of PAD can be found in 6.3.2).

Table 22 illustrates the PAD score in case of MAP and proposed two MSP^{UE} from two data-sets for different query collections. From this table, we can still observe that while using MSP^{UE} can achieve higher PAD score than the same for original MAP for all types of query collections, i.e., using *All* queries, *Uninformative* and *Ideal* query sub-sets. For instance, the average PAD of MAP on "MSLR-WEB30K" is **25.57**; while for $MSP^{UE}_{V_2}$, the score is **97.63** (using all query). Similarly, we can still discovered that for *Uninformative* query-set, both MSP^{UE} versions achieve a significant boost compared to the same in *Ideal* query set in both data-sets.

These results show that the proposed UE normalization again improve the distinguishability of original MAP and can better differentiate between the quality of two LETOR methods with a larger margin.

Table 22
Percentage Absolute Difference between pairs of LETOR methods in terms of average MAP and MSP^{UE} scores on *All*, *uninformative* or *ideal* query sets from two data-sets..

Metrics	PAD score					
	All Query		Uninform		Ideal	
	MSLR	MQ2007	MSLR	MQ2007	MSLR	MQ2007
MAP	25.570	5.910	12.280	5.890	30.180	6.770
$MSP^{UE}_{V_1}$	**31.840**	**6.860**	**16**	**7.190**	**35.530**	**8.040**
$MSP^{UE}_{V_2}$	**97.630**	**20.010**	**25.650**	**28.270**	**48.290**	**13.490**

Table 23
Swap rates between method ranks on *All/uniform/Ideal* queries across "MSLR-WEB30K" and "MQ2007" data-sets.

Metric	Swap rate		
	All	Uninform.	Ideal
MAP	0.250	0.357	0.285
$MSP^{UE}_{V_1}$	0.250	0.321	0.250
$MSP^{UE}_{V_2}$	**0.178**	**0.250**	0.321

Table 24
Swap rates between method ranks on *MSLR-WEB30K/MQ2007* data-sets across "uninformative" and "Ideal" query collections.

Metric	Swap rate	
	MSLR-WEB30K	MQ2007
MAP	0.142	0.392
$MSP^{UE}_{V_1}$	0.142	0.392
$MSP^{UE}_{V_2}$	**0.107**	**0.285**

Table 25
Swap rates between method ranks on *MSLR-WEB30K* data-sets across "broad" and "focused" query collections.

Metric	Swap rate
	MSLR-WEB30K
MAP	0.03
$MSP^{UE}_{V_1}$	0.03
$MSP^{UE}_{V_2}$	**0.00**

6.4.3. Consistency

This experiment again focuses to compare the relative ranking of LETOR methods in terms of their MAP and MSP^{UE} scores, separately, *across* different data-sets ("MQ2007" Vs "MSLR-WEB30K") as well as *across Uninformative* and *Ideal* query collections within the same data-set. We computed *swap rate* to quantify the consistency of rankings induced by MAP and MSP^{UE} metrics across different data-sets. Table 23 shows our swap rate results for MAP and MSP^{UE} across the two data-sets, "MSLR-WEB30K" and "MQ2007". In contrast to *identical* swap rate scores in $nDCG$ and DCG^{UE}, $MSP^{UE}_{V_2}$ can achieve a overall **lower** swap rate(swap rate of MAP is 0.25 while 0.178 for $MSP^{UE}_{V_2}$) across a data-sets comparison while considering all query set.

Table 24 also shows our swap rate results for MAP and MSP^{UE} across *Uninformative* Vs *Ideal* queries from the same data-set. Similarly, we can still observe that $MSP^{UE}_{V_2}$ can obtain a more consistent ranking results across different query collection, which is very useful for an evaluation metric.

Alternative Query and Document Partitioning: We also conducted two additional experiments to measure the stability/consistency of expected value normalization on MAP. Using the aforementioned "broad" and "focused" query partitions, we conducted the same consistency experiment as in Section 6.3.2. Table 25 shows the swap rate numbers for MAP and MSP^{UE} between method ranks for "MSLR-WEB30K" data set between "broad" and "focused" query partitions. Interestingly, we can notice that $MSP^{UE}_{V_2}$ even shows better consistency (swap rate is 0) compared to the original MAP(swap rate is 0.03). Similarly, in Table 26, we can see that both MAP and MSP^{UE} maintain the same rank when evaluating methods on the "left" and "right" document partitions (see Section 6.3.2 for definitions of "left" and "right" partitions).

7. Discussions and conclusion

In this paper, we presented a novel perspective towards evaluation of Information Retrieval (IR) systems. Specifically, we performed two case-study on *nDCG* and *MAP*, both are widely popular metrics for IR evaluation, and started with the observation

Table 26
Swap rates between method ranks on *MSLR-WEB30K* data-sets across "left" and "right" document collections.

Metric	Swap rate
	MSLR-WEB30K
MAP	0
$MSP^{UE}_{V_1}$	0
$MSP^{UE}_{V_2}$	0

that, traditional *nDCG* and *MAP* computation does not include a query-specific expected value normalization although they include a query-specific upper-bound normalization. In other words, the current practice is to assume a uniform expected value (zero) across all queries while computing *nDCG* and *MAP*, an assumption that is incorrect. This limitation raises a question mark on the previous comparative studies involving multiple ranking methods where an average evaluation metric score is reported, because *Uninformative* vs. *Informative* vs. *Ideal* queries are rewarded equally in traditional IR evaluation metric computation and the expected value of the evaluation metric is ignored. *How can we incorporate query-specific expected value normalization into IR evaluation metrics and how will it impact IR evaluation in general?* This is the central issue we investigated in this paper.

Conceptual Leap: To address the aforementioned issue, we proposed to penalize the traditional IR evaluation metric score of each query with an expected value normalization term specific to that query. To achieve this, we introduced a joint upper and expected value normalization (UE-normalization) framework and instantiated two versions of the UE-normalization, V_1 V_2, for two popular IR evaluation metric $nDCG$ and MAP, essentially creating four new evaluation metrics.

The next challenge in our work was to estimate a more realistic query-specific expected value for above two metrics. For this estimation, we argued that a reasonable ranking method should be at least as good as a random ranking method, so a more realistic expected value should be the score expected by a mere random ranking of the document collection rather than the current practice of assuming zero as an expected value across all queries. Using probability and permutation theory, we derived a closed-form formula to compute the expected DCG in case of random ranking. The proof was completed by showing that the expected relevance label of a document at position i is actually independent of the position and can be replaced by the expected relevance label of the document collection associated with the particular query in the validation data-set. For expected SP, we also use probability and induction to prove the correctness of our assumption. The derivation details can be found in each case study section.

Depth of Impact: Using two publicly available web search and learning-to-rank data-sets, we conducted extensive experiments with eight popular LETOR methods to understand the implications DCG^{UE} and MSP^{UE}. The implications are briefly summarized as follows:

1. Kendall's τ rank correlation coefficient test on two different rankings of multiple LETOR methods, where the ranks are induced by both traditional metric (i.e. $nDCG$ and MAP) vs UE-normalized metrics(i.e. DCG^{UE} and MSP^{UE}) yields **different conclusions** regarding the relative ranking of multiple LETOR methods.
2. Statistical Significance tests can lead to **conflicting conclusions** regarding the relative performance between a pair of LETOR methods, when comparing them in terms of traditional metrics vs UE-normalized metrics scores.
3. The above two observations are more prominent in case of *Uninformative* query collection.

Next, we systematically compared the traditional evaluation metric and UE-normalized metrics from two important perspectives: *distinguishability* and *consistency*. The findings are briefly summarized below.

1. Discriminative power analysis and PAD scores suggest that our metric can better **distinguish** between two closely performing LETOR methods. These results were confirmed through Student's t-test and PAD score analysis.
2. For *consistency*, $MSP^{UE}_{V_2}$ achieves the **lowest** swap rate across a data-sets comparison as well as the **lowest** swap rate while we compare the ranking results from *uninformative* vs. *ideal* query sets. On the other hand, the proposed DCG^{UE} metric is identical to the original $nDCG$ metric in terms of **consistency** across different data-sets as well as across *Uninformative/Ideal* query sets within the same data-set.
3. All above experiments reveal that the impact of expected value normalization is **more substantial** in case of "Uninformative" queries in comparison to "Ideal" queries, suggesting, expected value normalization is crucial when the validation set contains a large number of *Uninformative* queries (i.e., the ranking methods fail to perform significantly better than the randomly ranked output).

Breadth of Impact: The proposed expected value normalization technique is very general and can be potentially extended to other IR evaluation metrics like ERR, which is an exciting future direction. Another direction can be to investigate such expected value normalization for evaluation in domains other than IR, for example, ROUGE metric from the text summarization and NLP literature.

Final Words: The key take-away message from this paper is the following: *The IR community should consider expected value normalization seriously while evaluating any IR system.* Our work takes a first step towards this important direction and can serve as a pilot study to demonstrate the importance and implications of expected value normalization.

CRediT authorship contribution statement

Dongji Feng: Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Shubhra Kanti Karmaker:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing.

Data availability

Data will be made available on request.

Acknowledgment

This research has been funded by the College of Engineering at Auburn University. We also thank the Department of Computer Science and Software Engineering at Auburn University for their continuous support. All authors approved the version of the manuscript to be published.

References

Amigó, E., Mizzaro, S., & Spina, D. (2022). *Ranking interruptus: When truncated rankings are better and how to measure that* (pp. 588–598). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3477495.3532051.

Asadi, N., & Lin, J. (2013). Effectiveness/efficiency tradeoffs for candidate generation in multi-stage retrieval architectures. In *Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval* (pp. 997–1000).

Aslam, J. A., Yilmaz, E., & Pavlu, V. (2005). The maximum entropy method for analyzing retrieval measures. In *Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 27–34).

Bailey, P., Moffat, A., Scholer, F., & Thomas, P. (2015). User variability and IR system evaluation. In R. Baeza-Yates, M. Lalmas, A. Moffat, & B. A. Ribeiro-Neto (Eds.), *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval* (pp. 625–634). Santiago, Chile: ACM.

Billerbeck, B., & Zobel, J. (2004). Questioning query expansion: An examination of behaviour and parameters. In K. Schewe, & H. E. Williams (Eds.), *CRPIT*: *vol. 27*, *Database technologies 2004, proceedings of the fifteenth Australasian database conference* (pp. 69–76). Dunedin, New Zealand: Australian Computer Society.

Buckley, C., & Voorhees, E. M. (2017). Evaluating evaluation measure stability. In *ACM SIGIR forum, vol. 51, no. 2* (pp. 235–242). NY, USA: ACM New York.

Burges, C. J. (2010). From ranknet to lambdarank to lambdamart: An overview. *Learning*, *11*, 23–581.

Burges, C. J. C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., et al. (2005). Learning to rank using gradient descent. In L. D. Raedt, S. Wrobel (Eds.), *ACM international conference proceeding series*: *vol. 119*, *Machine learning, proceedings of the twenty-second international conference* (pp. 89–96). Bonn, Germany: ACM.

Cao, Z., Qin, T., Liu, T., Tsai, M., & Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In Z. Ghahramani (Ed.), *ACM international conference proceeding series*: *vol. 227*, *Machine learning, proceedings of the twenty-fourth international conference* (pp. 129–136). Corvallis, Oregon, USA: ACM.

Caragea, C., Honavar, V., Boncz, P., Larson, P., Dietrich, S., Navarro, G., et al. (2009). Mean average precision. *Encyclopedia of Database Systems*, 1703.

Chen, J., Liu, Y., Mao, J., Zhang, F., Sakai, T., Ma, W., et al. (2021). Incorporating query reformulating behavior into web search evaluation. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 171–180). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3459637.3482438.

Chen, N., Zhang, F., & Sakai, T. (2022). Constructing better evaluation metrics by incorporating the anchoring effect into the user model.

Clarke, C. L., Culpepper, J. S., & Moffat, A. (2016). Assessing efficiency–effectiveness tradeoffs in multi-stage retrieval systems without using relevance judgments. *Information Retrieval Journal*, *19*(4), 351–377.

Egghe, L. (2008). The measures precision, recall, fallout and miss as a function of the number of retrieved documents and their mutual interrelations. *Information Processing & Management*, *44*(2), 856–876.

Faggioli, G., Ferro, N., & Fuhr, N. (2022). Detecting significant differences between information retrieval systems via generalized linear models. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 446–456).

Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., & Lin, C. J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, *9*, 1871–1874.

Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, *4*, 933–969.

Ganjisaffar, Y., Caruana, R., & Lopes, C. V. (2011). Bagging gradient-boosted trees for high precision, low variance ranking models. In W. Ma, J. Nie, R. Baeza-Yates, T. Chua, & W. B. Croft (Eds.), *Proceeding of the 34th international ACM SIGIR conference on research and development in information retrieval* (pp. 85–94). Beijing, China: ACM.

Gienapp, L., Fröbe, M., Hagen, M., & Potthast, M. (2020). The impact of negative relevance judgments on NDCG. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 2037–2040).

Gienapp, L., Stein, B., Hagen, M., & Potthast, M. (2020). Estimating topic difficulty using normalized discounted cumulated gain. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 2033–2036).

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, *20*(4), 422–446.

Jia, Y., Wang, H., Guo, S., & Wang, H. (2021). Pairrank: Online pairwise learning to rank by divide-and-conquer. In *Proceedings of the web conference 2021* (pp. 146–157).

Jiang, J., & Allan, J. (2016). Adaptive effort for search evaluation metrics. In N. Ferro, F. Crestani, M. Moens, J. Mothe, F. Silvestri, G. M. D. Nunzio, & et al. (Eds.), *Lecture notes in computer science*: *vol. 9626*, *Advances in information retrieval - 38th European conference on IR research, ECIR 2016, March 20-23, 2016. proceedings* (pp. 187–199). Padua, Italy: Springer.

Kanoulas, E., & Aslam, J. A. (2009). Empirical justification of the gain and discount function for nDCG. In D. W. Cheung, I. Song, W. W. Chu, X. Hu, & J. J. Lin (Eds.), *Proceedings of the 18th ACM conference on information and knowledge management* (pp. 611–620). Hong Kong, China: ACM.

Karmaker Santu, S. K., Sondhi, P., & Zhai, C. (2016). Generative feature language models for mining implicit features from customer reviews. In *Proceedings of the 25th ACM international on conference on information and knowledge management* (pp. 929–938).

Karmaker Santu, S. K., Sondhi, P., & Zhai, C. (2017). On application of learning to rank for e-commerce search. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (pp. 475–484).

Keshvari, S., Ensan, F., & Yazdi, H. S. (2022). ListMAP: Listwise learning to rank as maximum a posteriori estimation. *Information Processing & Management*, *59*(4), Article 102962.

Kuzi, S., Labhishetty, S., Karmaker Santu, S. K., Joshi, P. P., & Zhai, C. (2019). Analysis of adaptive training for learning to rank in information retrieval. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 2325–2328).

Leo, B. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Lin, C. J., Weng, R. C., & Keerthi, S. S. (2008). Trust region newton method for logistic regression. *Journal of Machine Learning Research*, *9*, 627–650.

Metzler, D., & Croft, W. B. (2007). Linear feature-based models for information retrieval. *Information Retrieval*, *10*(3), 257–274.

Moffat, A., Scholer, F., & Thomas, P. (2012). Models and metrics: IR evaluation as a user process. In *Proceedings of the seventeenth Australasian document computing symposium* (pp. 47–54).

Moffat, A., Thomas, P., & Scholer, F. (2013). Users versus models: what observation tells us about effectiveness metrics. In Q. He, A. Iyengar, W. Nejdl, J. Pei, & R. Rastogi (Eds.), *22nd ACM international conference on information and knowledge management* (pp. 659–668). San Francisco, CA, USA: ACM.

Moffat, A., & Zobel, J. (2008). Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems*, *27*(1), http://dx.doi.org/10.1145/1416950.1416952.

Mothe, J., Laporte, L., & Chifu, A. G. (2019). Predicting query difficulty in IR: impact of difficulty definition. In *2019 11th international conference on knowledge and systems engineering* (pp. 1–6). IEEE.

Qin, T., & Liu, T. (2013). Introducing LETOR 4.0 datasets, CoRR abs/1306.2597. URL: http://arxiv.org/abs/1306.2597.

Qin, T., Liu, T. Y., Ding, W., Xu, J., & Li, H. (2010). Microsoft learning to rank datasets. Retrieved September 7, 2015.

Qin, T., Liu, T. Y., Xu, J., & Li, H. (2010). LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, *13*(4), 346–374.

Ravikumar, P., Tewari, A., & Yang, E. (2011). On NDCG consistency of listwise ranking methods. In G. J. Gordon, D. B. Dunson, & M. Dudík (Eds.), *JMLR proceedings*: *vol. 15*, *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 618–626). Fort Lauderdale, USA: JMLR.org.

Robertson, S. E., Kanoulas, E., & Yilmaz, E. (2010). Extending average precision to graded relevance judgments. In *Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval* (pp. 603–610).

Roitero, K., Maddalena, E., Mizzaro, S., & Scholer, F. (2021). On the effect of relevance scales in crowdsourcing relevance assessments for information retrieval evaluation. *Information Processing & Management*, *58*(6), Article 102688. http://dx.doi.org/10.1016/j.ipm.2021.102688, URL: https://www.sciencedirect.com/science/article/pii/S0306457321001734.

Sakai, T. (2006). Evaluating evaluation metrics based on the bootstrap. In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 525–532).

Sakai, T. (2007). On the reliability of information retrieval metrics based on graded relevance. *Information Processing and Management*, *43*(2), 531–548.

Sakai, T. (2016). A simple and effective approach to score standardisation. In *Proceedings of the 2016 ACM international conference on the theory of information retrieval* (pp. 95–104).

Sakai, T., Ishikawa, D., Kando, N., Seki, Y., Kuriyama, K., & Lin, C.-Y. (2011). Using graded-relevance metrics for evaluating community QA answer selection. In *Proceedings of the fourth ACM international conference on web search and data mining* (pp. 187–196).

Sakai, T., & Robertson, S. (2008). Modelling a user population for designing information retrieval metrics. In T. Sakai, M. Sanderson, & N. Kando (Eds.), *Proceedings of the 2nd international workshop on evaluating information access*. National Institute of Informatics (NII), URL: http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings7/pdf/EVIA2008/07-EVIA2008-SakaiT.pdf.

Santu, S. K. K., Sondhi, P., & Zhai, C. (2020). Empirical analysis of impact of query-specific customization of nDCG: A case-study with learning-to-rank methods. In M. d'Aquin, S. Dietze, C. Hauff, E. Curry, & P. Cudré-Mauroux (Eds.), *CIKM '20: The 29th ACM international conference on information and knowledge management* (pp. 3281–3284). Ireland: ACM.

Sarkar, S., & Karmaker Santu, S. K. (2022). Concept annotation from users perspective: A new challenge. In *Companion proceedings of the web conference 2022* (pp. 1180–1188).

Shukla, S., Lease, M., & Tewari, A. (2012). Parallelizing ListNet training using spark. In W. R. Hersh, J. Callan, Y. Maarek, & M. Sanderson (Eds.), *The 35th international ACM SIGIR conference on research and development in information retrieval* (pp. 1127–1128). Portland, OR, USA: ACM.

Tonellotto, N., Macdonald, C., & Ounis, I. (2013). Efficient and effective retrieval using selective pruning. In *Proceedings of the sixth ACM international conference on web search and data mining* (pp. 63–72).

Voorhees, E. M. (2001). Evaluation by highly relevant documents. In W. B. Croft, D. J. Harper, D. H. Kraft, & J. Zobel (Eds.), *SIGIR 2001: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 74–82). New Orleans, Louisiana, USA: ACM.

Voorhees, E. M., Samarov, D., & Soboroff, I. (2017). Using replicates in information retrieval evaluation. *ACM Transactions on Information Systems (TOIS)*, *36*(2), 1–21.

Wang, Y., Wang, L., Li, Y., He, D., & Liu, T. (2013). A theoretical analysis of NDCG type ranking measures. In S. Shalev-Shwartz, & I. Steinwart (Eds.), *JMLR workshop and conference proceedings*: *vol. 30*, *COLT 2013 - the 26th annual conference on learning theory* (pp. 25–54). Princeton University, NJ, USA: JMLR.org.

Webber, W., Moffat, A., & Zobel, J. (2010). The effect of pooling and evaluation depth on metric stability. In T. Sakai, M. Sanderson, & W. Webber (Eds.), *Proceedings of the 3rd international workshop on evaluating information access* (pp. 7–15). Tokyo, Japan: National Institute of Informatics (NII).

Xu, J., & Li, H. (2007). AdaRank: a boosting algorithm for information retrieval. In W. Kraaij, A. P. de Vries, C. L. A. Clarke, N. Fuhr, & N. Kando (Eds.), *SIGIR 2007: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 391–398). Amsterdam, The Netherlands: ACM.

Yilmaz, E., & Aslam, J. A. (2006). Estimating average precision with incomplete and imperfect judgments. In *Proceedings of the 15th ACM international conference on information and knowledge management* (pp. 102–111).

Yilmaz, E., Kanoulas, E., & Aslam, J. A. (2008). A simple and efficient sampling method for estimating AP and NDCG. In S. Myaeng, D. W. Oard, F. Sebastiani, T. Chua, & M. Leong (Eds.), *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 603–610). Singapore: ACM.

Yilmaz, E., Shokouhi, M., Craswell, N., & Robertson, S. (2010). Expected browsing utility for web search evaluation. In J. Huang, N. Koudas, G. J. F. Jones, X. Wu, K. Collins-Thompson, & A. An (Eds.), *Proceedings of the 19th ACM conference on information and knowledge management* (pp. 1561–1564). Toronto, Ontario, Canada: ACM.

Yilmaz, E., Verma, M., Craswell, N., Radlinski, F., & Bailey, P. (2014). Relevance and effort: An analysis of document utility. In J. Li, X. S. Wang, M. N. Garofalakis, I. Soboroff, T. Suel, & M. Wang (Eds.), *Proceedings of the 23rd ACM international conference on conference on information and knowledge management* (pp. 91–100). Shanghai, China: ACM.

Yu, H. T., Jatowt, A., Blanco, R., Joho, H., & Jose, J. M. (2017). An in-depth study on diversity evaluation: The importance of intrinsic diversity. *Information Processing & Management*, *53*(4), 799–813. http://dx.doi.org/10.1016/j.ipm.2017.03.001, URL: https://www.sciencedirect.com/science/article/pii/S030645731630591X.