Upper and Expected Value Normalization for Evaluating Information Retrieval and Text Generation Systems

by

Dongji Feng

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
Aug 5, 2023

Keywords: Information Retrieval, Natural Language Processing, Evaluation, Expected Value Normalization

Approved by

Shubhra Kanti Karmaker ("Santu"), Chair, Assistant Professor, Department of CSSE
Cheryl Seals, Charles W. Barkley Professor, Department of CSSE
Yang Zhou, Assistant Professor, Department of CSSE
Ashish Gupta, Professor, Department of Systems and Technology
Mehdi Sadi (University Reader), Assistant Professor Department of ECE

Abstract

Evaluation metric is a crucial part to improve the performance of any system. An accurate evaluation metric is capable to detect and compare multiple different models and thus serving the user's need in a specific domain. Ranking in Information Retrieval (IR) and text summarization in Natural Language Processing (NLP) are two important tasks and often served as a key component within an intelligent system. For instance, search engine uses a ranking algorithm to determine the order in which the search results are displayed. The ranking algorithm analyzes various factors to evaluate the relevance and quality of web pages and then assigns them a ranking based on their perceived value to the user. Although many different evaluation metrics had been proposed for a better understanding of the ranking/summarization models and to improve an intelligent system, empirical evaluation is still a challenge.

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. For the NLP domain, we specifically consider ROUGE and BERTScore in the text summarization evaluation and conducted the two case studies by instantiating the new framework for ROUGE/BERTScore to observe the implications, where the expected ROUGE/BERTScore is calculated by an expected summary given a source document, resulting in an instance-level penalty for each source document.

For the ranking task, 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.

For the text summarization task, we also conducted the expected value normalization on two widely used metrics, ROUGE and BERTScore. Experiments on CNN/Daily Mail datasets with 12 different abstractive summarization models also demonstrate the following properties of the new expected value normalized metric: 1) When compared against the original metric, our proposed UE normalized BERTScore demonstrate higher human correlation w.r.t. four important perspectives (Consistency, Coherence, Relevance, Fluency) across 12 abstractive summarization methods, especially in *Heterogeneous documents*, 2) Human judgment favors Upper expected value normalized BERTScore against original version across comparison of 6 extractive summarization methods. On the other hand, for ROUGE score, UE normalization does not help much in terms of human correlation with abstraction summarization methods, though it improves the human correlation with extractive summarization methods. These findings suggest that the IR and NLP community should consider UE normalization seriously when computing *nDCG*, *MAP*, *ROUGE* and *BERTScore*, more in-depth study of UE normalization for general IR and NLP evaluation is warranted.

料峭春风吹酒醒，微冷，山头斜照却相迎。
回首向来萧瑟处，归去，也无风雨也无晴。

-苏轼《定风波》

-Su Shi, a Chinese Poet during Song Dynasty

Acknowledgments

My Ph.D. journey has been incredibly fortunate because of all the wonderful people I have met who generously extended their help along the way. First and foremost, I want to thank my amazing advisor Dr.Shubhra Kanti Karmaker Santu, for his trust, mentorship, understanding, and unwavering support. I will always admire his rigorous attitude toward scientific research and his passion for teaching.

I would like to thank my committee, Dr.Cheryl Seals, Dr.Yang Zhou, and Dr.Ashish Gupta for your feedback and advice. I also want to thank my university reader, Dr.Mehdi Sadi for providing insightful perspectives.

I want to thank all the people I met from Auburn University: Auburn University CSSE faculty members, especially Dr.Jakita O. Thomas, for giving me the chance to enhance my research perspectives; Dr.Xiao Qin for always helping and advising me, and other Auburn University friends, Dr.Yuanzhi Yao for always encouraging me. Shangjun Wang, Jindong Wei, Zheng Zhang, Mostafa Rahgouy, Xiaoniu Du, Shuxin Liu, Nan Yang, Dr.Wenying Li, Dr.Yin Sun, Dr.Linqiang Ge, Dr.Junyao Yang, and Dr.JingJing Li for our cooking/basketball/soccer/research time

I want to thank my excellent BDI lab members, especially Souvika and Mousumi for your help during our team works. Then I want to thank all my other collaborators who have given me valuable advice.

I want to thank my old friends, Kailei Ning, Mochun Li, Jing Bai, Zhicheng Zhou, and Zifan Zhao, without whom and all the memories they bought me I could not overcome the most difficult time.

Finally, I want to sincerely thank all my family. Especially my parents, my wife, and my son. All beyond research.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the current landscape of AI advancements, choosing one model over another is as important as designing a new one, even more difficult. The key point lies in the evaluation metric, a pivotal measure of a model's performance. Nevertheless, the task of designing an appropriate evaluation metric becomes intricate due to varying standards for model quality, contingent on individual goals and objectives. This diversity of perspectives presents a challenge in creating a well-crafted evaluation metric. This thesis centers on the absence of a prior evaluation metric design which we call ”expected value normalization.” We extensively conduct experiments using four distinct metrics within the domains of Information Retrieval (IR) and Natural Language Processing (NLP) to form the basis of this research. The ultimate goal is to advocate for the adoption of expected value normalization to enhance model selection and utilization for practitioners and researchers alike.

1.1 Evaluation in IR

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 $nDCG$ [37], MAP [14] 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.

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 relevant 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 thesis, 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, and 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 well 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 [12], RankBoost [27], AdaRank [91], Random Forest [8], LambdaMART [11], CoordinateAscent [53], ListNet [13] and L2 regularized Logistic Regression [26, 49]. 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 [71, 69]) 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 3.5 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 the case of discriminative power, we followed [69, 72] to use the 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 IR evaluation and consider expected value normalization seriously. In summary, we make the following contributions to the thesis in the IR domain:

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.

1.2 Evaluation in NLP

Automatic evaluation of natural language generation is an important component to improve the natural language understanding system such as text summarization, machine translation, and caption generation [100]. Text summarization, for instance, can be considered as a text-to-text task where the input is a long document and the output is the corresponding summary which is shorter, human-readable, and only generated from the source document. Previous researchers had proposed many different text summarization evaluation metrics such as ROUGE [50] which is based on n-gram overlapping and more advanced metrics such as BERTScore and BARTScore [98] which utilized the large language model and transformers to improve the understanding of the semantic meaning of source document that results in the higher human correlation.

With a similar question, as we had from IR evaluation, these two (ROUGE and BERTScore) important and widely used evaluation metrics had neither expected nor upper-value normalization. Using our proposed two frameworks, we conducted the upper and expected value normalization toward two metrics by proposing different expected scores. For ROUGE, which measures the number of overlapping textual tokens, we leverage a unigram language model to generate the expected token based on the distribution of words in the source document. For BERTScore, we use a transformer encoder to tokenize the source document and get the similarity/contribution of each token w.r.t. the entire document, then generate the expected summary based on the similarity.

Different from the IR evaluation metric, the key point of understanding the performance in NLP is the human correlation [24, 1, 98]. In this thesis, we use the [24] dataset and calculate the human correlation of our proposed metrics from 4 perspectives w.r.t. 12 abstractive text summarization methods. We also conducted the human correlation w.r.t. 6 extractive text summarization which was annotated by three NLP Ph.D. experts. The empirical results indicate that our UE-BERTScore achieves a higher correlation on 12 abstractive summarization methods as well as 6 extractive summarization methods. In particular, we found UE normalization involves the maximum improvement for documents with heterogeneous contextualization (HeteDoc), in which the contextualized word embedding are different from contextualized document vector.

These findings suggest that the community should also consider expected value normalization in the NLP evaluation task. In summary, we make the following contributions to the thesis in the NLP domain:

1. We propose two different variants of the proposed UE normalized version for two popular NLP evaluation metrics, BERTScore and ROUGE.
2. We show how we can compute a more realistic instance-specific expected value for two NLP evaluation metrics separately: For ROUGE, we use an instance-specific unigram language model to sample the expected summary while for BERTScore, we greedily select the word from original source document based on contextualized word embedding to generate the expected summary. For both metrics, we use their expected summary to calculate the expected ROUGE/BERTScore individually.
3. We conducted extensive experiments to understand the implications of the expected value normalized metric and compared our proposed metric against the original metric from the **Human Correlation** perspective.
4. Our proposed framework is very general and can be easily extended to other NLP evaluation metrics.

1.3 Thesis Outline

The rest of the thesis is organized as follows: Chapter 2 reviews related works from the past literature. Chapter 3 provides essential background about our four experimental metric computations and motivation for expected value normalization. Chapter 4 provides the details about data-sets and LETOR/Summarization methods that have been conducted in our experiment. Chapter 5 presents our proposed framework with instance-specific upper and expected value normalization. In Chapter 6, we first present how to compute a reasonable expected *DCG* by computing the corresponding score of a randomly ranked document collection in the case of each individual query. Then show the implications of our two proposed upper and expected value normalized *DCG*, i.e., $DCG^{UL}_{V_{1,2}}$. Chapter 7 follows the same format as chapter 6 to demonstrate our experimental results concerning another popular IR evaluation metrics: *MAP*. In Chapter 8, we first explain how we understand the expected score should be calculated in the case of BERTScore and demonstrate the UE normalization impact from the perspective of human correlation. Chapter 9 shows the results of UE normalization in ROUGE, which is our last case study in this thesis.

Finally, Chapter 10 discusses our current implementations on the traditional Information Retrieval domain and NLP domain. Nevertheless, this is not a destination but another beginning of a wonderful journal. I will also discuss and set my future plan in Chapter 11 from two perspectives to continue this work.

Chapter 2

Related Work

In this Chapter, we will go over previous literature related to the customization of traditional evaluation metrics, more specifically, about *nDCG*, *MAP*, *ROUGE* and *BERTScore* in Information Retrieval and NLP. We also discuss the limitations of the above four metrics and how previous researchers tried to solve them. Then we explain how our work is distinct from prior studies.

2.1 Evaluation in IR

Traditional IR evaluation metric: Many metrics have been introduced for the IR system evaluation [52] in recent years. The two most frequent and basic metrics for the performance evaluation of the IR system are *precision* and *recall*. Empirical studies of retrieval performance have shown a tendency for *precision* to decline as *recall* increases [9]. Due to the trade-off between the two basic calculations, researchers also use other complex single metrics such as *F-measure* which can evenly weight the *precision* and *recall*. Other popular metrics such as MAP (Mean Average Precision), Normalized Discounted Cumulative Gain ($nDCG$), and Expected Reciprocal Rank (ERR) 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 [37]. The benefit of *nDCG* is it can be applied to multi-level relevance judgments and is also sensitive to small changes in a ranked list. Many researchers have investigated its properties (see, e.g., [94, 65, 86]). The fact that the general concept of $nDCG$ can be implemented in a variety of ways was recognized in the previous work [41], 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 $nDCG$ [43]. 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 [84]. Regarding $nDCG$ cutoff-depths, Sakai and others [70] 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 [42], 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, [31] 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 [10] and highly informative measure [3]. Whereas Mean Average Precision (*MAP*) [14] is the average AP of each class which can reflect the overall performance among multiple topics. However, the assumption behind *MAP* is that retrieved documents can be considered as either relevant or non-relevant to the user's information need, which is not accurate. Previous researchers have studied the properties of *MAP* in terms of different relevance judgments. [93], 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, [66] 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, [17] proposed a framework for query-level evaluation metrics by incorporating the anchoring effect into the user model and achieving a better correlation with user satisfaction. [15] proposed query reformulation aware metric as query reformulating behaviors may reflect user's search intents. [47] 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. [55] followed by [5] 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 [7]. Such work has shown the need to employ a ranking function for each individual query. [23] 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. [67] studied the effect of the choice of relevance scales on the evaluation of IR system. [87] 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 [39] showed that the adaptive effort metrics can better indicate user's search experience compared with conventional metrics. [95] 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 [96]. [73] modeled a user population to assess the appropriateness of different evaluation metrics.

2.2 Evaluation in NLP

Traditional evaluation in NLP: Natural Language text generation task such as text summarization is commonly evaluated using annotated references. Given a golden reference and model generation, a better evaluation metric can achieve a higher human correlation. In the early stage, the metric for text summarization is to count the n-gram that occurs in the reference and model generation, which is a simple overlapping matching process. Due to the simplicity, the most commonly used metric is ROUGE [50] (Recall-Oriented Understudy for Gisting Evaluating) which measures the number of overlapping units (n-grams) between golden reference and model generation even though many limitations have been discovered [1, 32, 21]. Other variants of ROUGE such as ROUGE-WE [57], ROUGE-L had been proposed to capture longer overlapping. However, those N-gram-based metrics ignored contextualized information from source and reference documents resulting in a lower human correlation.

BERTScore: Existing Pre-trained model-based metrics can be categorized into three paradigms: *matching*, *regression*, and *generation* [78]. As a matching-based evaluation metric, BERTScore utilized the BERT or other pre-trained model (such as RoBERTa) to capture the contextual information from reference and model generation at a token level and greedily maximize the cosine similarity between contextualized token embeddings from BERT. Although this metric can achieve higher human correlation, previous researchers had investigated the property and the implementation on different domains [90, 68]. For instance, [78] demonstrated the popular pre-trained language model-based metrics exhibit significantly higher social bias than traditional metrics on six sensitive attributes. Other concerns regarding the sensitivity of BERTScore on translation tasks where the incorrect penalization was included when lexical similarity exists between the translations and references [34].

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 instance-level (query-specific) expected value normalization for IR and NLP evaluation metrics. Furthermore, our work is groundbreaking in that it proposes a generic upper expected value normalization framework and effectively applies it to four prominent evaluation metrics, crossing two important domains. We additionally compute the expectations over a randomized ranked list to estimate a more realistic expected value and also give the derivation for IR; while also proposing metric-specific expected scores in two text summarization evaluation tasks. Our research clearly articulates the effects of such expected value normalization on four popular evaluation metrics and lays the foundation for future research in this direction.

2.3 Chapter Summary

This chapter briefly reviews the customization for IR and NLP evaluation metrics from past literature and also explains how our proposal is different from previous studies. The next chapter will talk about the background information of IR and NLP evaluation metrics are conducted in this thesis.

Chapter 3

Background of Original IR and NLP Evaluation Metrics

In this Chapter, we start by providing some essential background about *nDCG*, *MAP*, *ROUGE* and *BERTScore* computation and then introduce our motivation of expected value normalization for the four metrics based on our observation that none of above metrics involve an instance-level expected value bound normalization.

3.1 Computation of the 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 [37]. This is accomplished by introducing a penalty term that penalizes the gain value logarithmically proportional to the position of the result [86]. Mathematically:

$$DCG@k = \sum_{i=1}^{k} \frac{2^{R_i} - 1}{\log_b(i+1)} \tag{3.1}$$

Here, i denotes the position of a document in the search ranked list and R_i is the relevance label of the $i-th$ 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.

$$nDCG@k \;=\; \frac{DCG@k}{IDCG@k} \tag{3.2}$$

3.2 Computation of the Standard MAP

For our third 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.

$$AP@k = \frac{1}{k}\sum_{i=1}^{k} Prec(i) \cdot R_i \tag{3.3}$$

Here, R_i is an indicator variable that says whether i^{th} item is relevant ($R_i = 1$) or non-relevant ($R_i = 0$). From Formula 3.3, 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 an upper-bound normalized version of $SP@k$, like $nDCG@k$ is for $DCG@k$. Finally, the 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), which is defined as follows:

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 lower bound normalization.*

$$SP@k = \sum_{i=1}^{k} Prec(i) \cdot R_i \tag{3.4}$$

3.3 Computation of the standard ROUGE

Our first NLP evaluation metric is the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scoring algorithm which calculates the similarity between a candidate document and a collection of reference documents. [50] introduced a ROUGE package with four different ROUGE measures: ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. We specifically introduce ROUGE-N, which is the N-gram Co-Occurrence statistics. For more details, refer to [50]. Formally, ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries. Where n stands for the length of the n-gram, $gram_n$, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries. ROUGE-N is computed as follows:

$$= \frac{\sum_{S \in References} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in References} \sum_{gram_n \in S} Count(gram_n)} \tag{3.5}$$

3.4 Computation of the standard BERTScore

For the second NLP evaluation metric, we utilize the recently introduced model-based metric, BERTScore. Given a reference sentence $X = < x_1, ...x_k >$ and a candidate sentence $\hat{X} = < \hat{x}_1, ...\hat{x}_k >$, BERTScore uses contextual embeddings to represent the tokens, and compute matching using cosine similarity [100]. The BERTScore Recall, Precision, and F1 scores are below:

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i^{\top} \hat{x}_j \tag{3.6}$$

$$R_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^\top \hat{x}_j \tag{3.7}$$

$$F_{BERT} = 2\frac{P_{BERT}.R_{BERT}}{P_{BERT} + R_{BERT}} \tag{3.8}$$

The complete score matches each token in X to a token in $\hat{X}$ to compute recall and each token in $\hat{X}$ to a token in X to compute precision. They use greedy matching to maximize the matching similarity score, where each token is matched to the most similar token in the other sentence. Then combine the precision and recall to compute the F1 score. In this thesis, all BERTScore results are using the BERTScore-F1 score.

3.5 Motivation for Expected Value Normalization in IR

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 [30] 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 thesis:

1. **Informative Queries:** These are queries where a *reasonable* ranking method performs significantly better than a pure random ranking system. Essentially, these are queries that contain the "right" keywords to find out the most relevant documents according to the user's information needs. Therefore, the actual evaluation metric scores are much higher than the expected value (the lower triangle region of the plot 8.2).

   Ideal Queries: These are special cases of *Informative* queries where the difference between the 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 (the 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 [2, 19, 80]) and there is little room for improving beyond random ranking.

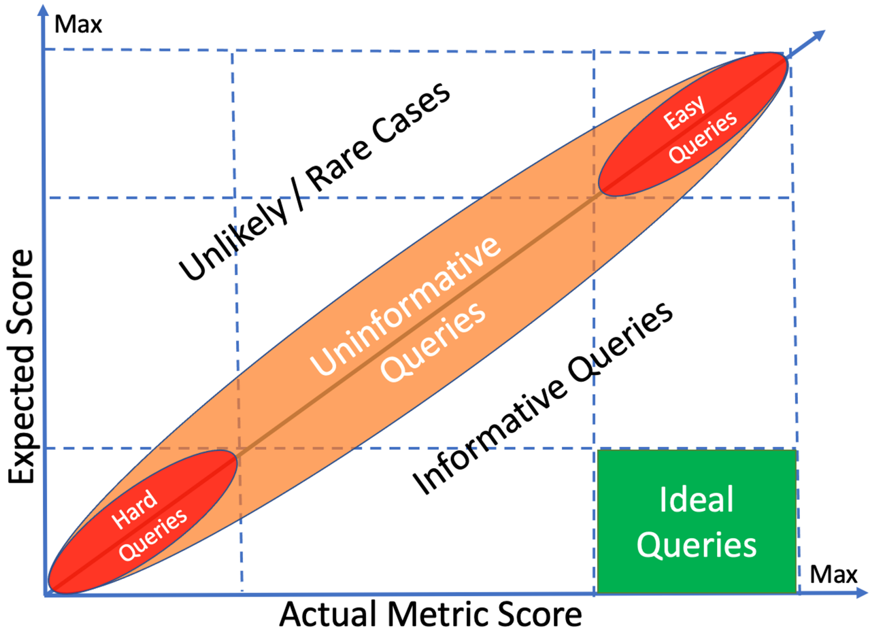


Figure 3.1: Query types with different expected values of evaluation metric.

Figure 8.2 shows an illustration of different types of queries with different combinations of evaluation metric expected value and actual metric score. As apparent from Figure 8.2, 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 the 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.

3.6 Motivation for Expected Value Normalization in NLP

As we can see from the formula of ROUGE, the original ROUGE score has neither upper nor expected value bound which assumes the expected ROUGE score should be 0, an inaccurate assumption that we want to argue with. Think about the process of extracting the word from the source document to generate the summary, because ROUGE omits the order sensitivity, words that occur more often are more likely to be picked up during a random selection and the sequence of words does not affect the ROUGE value. Thus, an expected value with a random extraction should be high if those words occur equally high in both original source document and reference because a random extraction can still achieve a reasonable ROUGE score, a customized penalty should be involved for different cases.

Although BERTScore has been designed to be bounded between 0 to 1, which means there is an upper bound, there is no expected value bound normalization. However, different source document has a different distribution of dominant words, which can be considered as keywords that would mostly impact the entire document, making the difficult for contextual understanding differently and further, generating the expected summarization.

Due to the existence of different types of documents, the expected value normalization is necessary because $1)$ it can better capture the order sensitivity as well as variations across different source documents. $2)$ it makes fairer comparisons and averaging across different documents.

3.7 Chapter Summary

In this Chapter, we first provide some basic information about the two IR evaluation metrics, *nDCG*, *MAP*, and two text summarization evaluation metrics, *ROUGE* and *BERTScore*. Then, we explain our motivation for this expected value normalization and also provide the justification. The next chapter will discuss the details of our experimental design such as data-sets and ranking/summarization methods.

Chapter 4

Experimental Design

This chapter provides some background information on the two data-sets and eight LETOR methods used in our IR experiments as well as CNN/DailyMail data set and extractive/abstractive summarization methods in text summarization experiments. We also give the criteria of human correlation from four perspectives and how we collect the human annotation from 5 pairs of comparison.

4.1 Data Set

4.1.1 Data Set in IR

We used two LETOR benchmark data-sets, i.e., “MSLR-WEB30K” [62] and ”MQ2007” [61] for our experiments. The first and second data-set includes 30,000 and 1,700 queries respectively and have widely been used as benchmarks for LETOR tasks [28, 77, 38, 44].

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 the 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 the relevance label, the more relevant the query-document pair is. Features related to each query-document pair are represented by a 136-dimensional feature vector for “MSLR-WEB30K” and a 46-dimensional feature vector for “MQ2007” data-set [42]. For more details on how the features were constructed, see [61] and [62]. Table 4.1 shows the number of queries, documents, and features for each data-set

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

Table 4.1: IR data set statistics

Algorithm	Short form	Algorithm	Short form
RankNet [12]	RNet	LambdaMART [11]	LMART
RankBoost [27]	RBoost	CoordinateAscent [53]	CA
AdaRank [91]	ARank	ListNet [13]	LNet
Random Forest [8]	RF	Logistic Regression [26]	L2LR

Table 4.2: Popular learning to rank algorithms

[44]. The documents in MQ2007 are retrieved from 25 million pages in the Gov2 web page collection [63] 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 reporting 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.

4.1.2 Data Set in NLP

We used CNN/Daily Mail [35] data-set and the human correlation from [24] to conduct our text summarization experiment. In this benchmark paper, Fabbri assembled and re-evaluated 14 evaluation metrics in a comprehensive and consistent fashion and using expert and crowd-sourced human annotations as a golden reference, addressed the existing shortcoming of summarization evaluation methods. The human annotations were evaluated along the following four dimensions, as in [45]:

	CNN			Daily Mail		
	train	valid	test	train	valid	test
# months	95	1	1	56	1	1
# docs	90,166	1,220	1,093	196,961	12,148	10,397
# queries	380,298	3,924	3,198	879,450	64,835	53,182
Max # entities	527	187	396	371	232	245
Avg # entities	26.4	26.5	24.5	26.5	25.5	26.0
Avg # tokens	762	763	716	813	774	780
Vocab size	118,497			208,045		

Table 4.3: Corpus statistics. Articles were collected starting in April 2007 for CNN and June 2010 for the Daily Mail, both until the end of April 2015. Validation data is from March, and test data from April 2015. Articles of over 2000 tokens and queries whose answer entity that did not appear in the context were filtered out.

1. **Coherence**: We follow the dimension with DUC quality question [20] of structure and coherence whereby "the summary should be well-structured and well-organized".
2. **Consistency**: the factual alignment between the summary and the summarized source. A factually consistent summary contains only statements that are entailed by the source document. Annotators were also asked to penalize summaries that contained hallucinated facts
3. **Fluency**: the quality of individual sentences. Drawing again from the DUC quality guidelines, sentences in the summary "should have no formatting problems, capitalization errors, or obviously ungrammatical sentences (e.g., fragments, missing components) that make the text difficult to read."
4. **Relevance**: selection of important content from the source. The summary should include only important information from the source document.

Human Annotation

We use the human annotations collection from [24] which contains summary evaluations of 16 recent neural summarization models solicited from crowd-sourced and expert judges. Annotations were collected for 100 articles randomly picked from CNN/Daily Mail test set. Statistical information of CNN/Daily Mail can be found in table 4.3. We also have extra human annotation w.r.t 5 pair of extractive summarization methods comparison which are annotated by three NLP Ph.D. experts for our experiments.

4.2 Methods

4.2.1 Learning to Rank (LETOR) Methods

Learning to rank is the application of machine learning algorithms in the construction of ranking tasks for information retrieval systems. In general, the three major approaches to learning to rank tasks are known as pointwise, pairwise, and listwise. In this proposal, we use the benchmark collection for research on Learning to Rank(LETOR) for Information Retrieval [63] and select eight popular LETOR methods as our evaluation target. Table 4.2 contains our selected eight prominent LETOR approaches along with popular classification and regression methods used for ranking applications. We also assign acronyms to each approach for notational convenience, which we will use throughout the rest of the proposal. Basic information related to each LETOR method is listed below:

RankNet: A pairwise approach introduced by Burges et al. where [12] proposed a probabilistic cost for training systems to learn ranking functions using pairs of training examples and explored the implementation using a neural network formulation.

LambdaMART: A pairwise approach [89] that is a combination of LambdaRank and MART [11]. While MART leveraged a gradient-boosted decision tree to tackle the prediction task. LambdaMART improved this technique by introducing a cost function derived from LambdaRank on the gradient-boosted decision tree to order any ranking task.

RankBoost: A pairwise approach proposed by Freund et al. [27] that can iteratively create and aggregates a collection of “weak rankers” to build an effective ranking procedure.

Coordinate Ascent: A listwise approach described as an optimization method in the paper [53]. This method optimizes through minimization of measure-specific loss, more specifically, the mean average precision (*MAP*).

AdaRank: Xu et al. [91] proposed this listwise approach within the framework of boosting, which can minimize a loss function directly defined on the performance measures. AdaRank repeatedly constructs ’weak rankers’ on the basis of reweighted training data and finally linearly combines the weak rankers for making ranking predictions.

ListNet: A listwise approach proposed by Cao et al.from [13] where they employed a new learning method for optimizing the listwise loss function based on top one probability, with the neural network as model and gradient descent as an optimization algorithm.

Random Forest: A pointwise approach proposed by Breiman [8]. Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest

Logistic Regression: In this proposal we also provide ranking results of l2-regularized logistic regression which is a simple but popular classifier that has been used in ranking tasks [26].

4.2.2 Abstractive Text Summarization Methods

Pointer Generator: [76] propose a variation of encoder-decoder models, the Pointer Generator Network, where the decoder can choose to generate a word from the vocabulary or copy a word from the input. A coverage mechanism is also proposed to prevent repeatedly attending to the same part of the source document.

Fast-abs-rl: [18] propose a model which first extracts salient sentences with a Pointer Network and rewrites these sentences with a Pointer Generator Network.

Bottom-Up: [29] introduce a bottom-up approach whereby a content selection model restricts the copy attention distribution of a pre-trained Pointer Generator Network during inference.

Improve-abs: [46] extend the model of [59] by augmenting the decoder with an external LSTM language model and add a novelty RL-based objective during training.

Unified-ext-abs: [36] propose to use the probability output of an extractive model as sentence-level attention to modifying word-level attention scores of an abstractive model, introducing an inconsistency loss to encourage consistency between these two levels of attention.

ROUGESal: [58] propose a keyphrase-based salience reward as well as an entailment-based reward in addition to using a ROUGE-based reward in a REINFORCE setting, optimizing rewards simultaneously in alternate mini-batches.

Multi-task (Ent + QG): [33] propose question generation and entailment generation as auxiliary tasks in a multi-task framework along with a corresponding multi-task architecture.

Closed book decoder: [40] build upon a Pointer Generator Network by adding copy-less and attention-less decoders during training time to force the encoder to be more selective in encoding salient content

T5: [64]perform a systematic study of transfer learning techniques and apply their insights to a set of tasks all framed as text-input to text-output generation tasks, including summarization.

GPT-2: [101] build off of GPT-2 and fine-tune the model by using human labels of which of four sampled summaries are the best to direct fine-tuning in a reinforcement learning framework.

BART: [48]introduce a denoising autoencoder for the pretraining sequence to sequence tasks which are applicable to both natural language understanding and generation tasks.

PEGASUS: [99] introduce a model pre-trained with a novel objective function designed for summarization by which important sentences are removed from an input document and then generated from the remaining sentences.

4.2.3 Extractive Text Summarization Methods

BERT: [22] designed pre-train deep bidirectional representations from the unlabeled text by joint conditioning on both left and right context in all layers.

MobileBERT: [79] is a thin version of BERT-LARGE, while equipped with bottleneck structures and a carefully designed balance between self-attentions and feed-forward networks.

DistilBERT: [74] is a small, fast, cheap, and light Transformer model based on the BERT architecture. Knowledge distillation is performed during the pre-training phase to reduce the size of a BERT model by 40%.

RoBERTa: [51] builds on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates.

XLNet: [92] is an extension of the Transformer-XL model pre-trained using an autoregressive method to learn bidirectional contexts by maximizing the expected likelihood over all permutations of the input sequence factorization order

4.3 Chapter Summary

This chapter introduces the basic information about the three data-sets used in our experiment (two in the IR task and one in the text summarization task). We also provide some background of each LETOR method, 12 abstractive text summarization methods, and 6 extractive text summarization methods conducted in our implication experiment.

Chapter 5

Evaluation with Joint Upper & Expected value Normalization

This chapter defines the general framework with both upper and expected value normalization for our evaluation metrics in IR and NLP system and briefly explain the intuition behind this consideration.

5.1 General Framework for IR:

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.

- $\mathbf{IUB}[\mathbf{A@k}]$: 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.
- $\mathbf{REB}[\mathbf{A@k}]$**:** 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 $[A@k]^U = \frac{A@k}{IUB[A@k]}$.

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.

$$[A@k]^{UE}_{V_1} = \left(\frac{A@k}{IUB[A@k]}\right)\left(\frac{A@k}{(A@k + REB[A@k])}\right) \tag{5.1}$$

$$[A@k]^{UE}_{V_2} = \begin{cases} \frac{A@k - REB[A@k]}{IUB[A@k] - REB[A@k]}, & \text{if } A \geq REB \\ \\ \frac{A@k - REB[A@k]}{REB[A@k]}, & \text{otherwise} \end{cases} \tag{5.2}$$

In the first Equation **5.1**, we introduce a linear penalty term for Upper Expected Value Normalization while in the second Equation **5.2** 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.

5.2 General Framework for NLP:

Now, we introduce two different variations of Joint Upper & Expected Value Normalization for NLP, which is denoted by $[A]^{UE}$. We still call the two versions as V_1, V_2. The difference between IR and NLP evaluation is we do not have a cutting position *K*, instead, we could calculate the expected value for each instance. Thus, the only difference is omitting *K* in our V1 and V2 from IR domain, the following equations can show the framework for NLP UE normalization:

$$[A]^{UE}_{V_1} = \left(\frac{A}{IUB[A]}\right)\left(\frac{A}{(A + REB[A])}\right) \tag{5.3}$$

$$[A]^{UE}_{V_2} = \begin{cases} \frac{A-REB[A]}{IUB[A]-REB[A]}, & \text{if } A \geq REB \\ \\ \frac{A-REB[A]}{REB[A]}, & \text{otherwise} \end{cases} \tag{5.4}$$

5.3 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 Equation **5.1**, 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 (Equation **5.2**) 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 Equation **5.1**, the range is still $[0, 1]$; while for Equation **5.2**, 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. The range of UE in IR is the same as the range of UE in NLP.

5.4 Chapter Summary

This chapter provides detailed explanations of our general framework which involve both upper and expected value normalization. We also discuss the intuition and range of this framework for both IR and NLP domains.

Chapter 6

nDCG with Joint Upper & Expected Value Normalization

In this chapter, we will introduce our first case study that we implement our general framework to *nDCG*, a widely used evaluation metric for IR systems. We first describe how to compute a more realistic Expected value for *DCG*, i.e., the expected DCG in case of a randomly ranked document for a particular query. Then discuss the implications of the new proposed upper and expected value normalized DCG.

6.1 Expected DCG@k:

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 proposal.

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 ϕ = {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.

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

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 relevant label j (with 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.

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

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

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 the 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:

$$E[DCG@k] = E[2^R-1]\cdot\sum_{i=1}^{k}\frac{1}{log_2(i+1)} \tag{6.1}$$

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 6.1: $nDCG$ scores of different LETOR methods for variable k on MSLR-WEB30K data-set.

Method	nDCG@				
	5	**10**	**15**	**20**	**30**
ARank	0.3881	0.4156	0.448	0.4797	0.5372
LNet	0.3767	0.4035	0.4384	0.4687	0.5282
RBoost	0.3834	0.414	0.449	0.4807	0.5355
RF	0.4035	0.4286	0.4609	0.4914	0.5476
RNet	0.3809	0.4131	0.4451	0.4764	0.536
CA	0.3928	0.4207	0.4544	0.4824	0.5399
L2LR	0.3873	0.4159	0.4474	0.4779	0.538
LMART	0.3931	0.4206	0.4535	0.4857	0.5441

Table 6.2: $nDCG$ scores of different LETOR methods for variable k on MQ2007 data-set.

6.2 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 from [42] to compute baseline *nDCG* scores. Table 6.1 and 6.2 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 Table 6.1 and 6.2 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

	$DCG^{UE}_{V_1}$@					$DCG^{UE}_{V_2}$@				
Method	**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

Table 6.3: Upper & Expected Value Bound Normalized DCG (V_1,V_2) scores of different LETOR methods for variable k: Each cell shows a particular DCG^{UE}_{V} score with a particular k on MSLR-WEB30K data-set

	$DCG^{UE}_{V_1}$@					$DCG^{UE}_{V_2}$@				
Method	**5**	**10**	**15**	**20**	**30**	**5**	**10**	**15**	**20**	**30**
ARank	0.2882	0.2991	0.3157	0.3314	0.3558	0.1348	0.2092	0.2587	0.2995	0.3638
LNet	0.2777	0.2886	0.3068	0.3213	0.3481	0.1141	0.1872	0.2485	0.284	0.3453
RBoost	0.2822	0.2975	0.3161	0.3317	0.3542	0.1359	0.2061	0.2633	0.3042	0.3635
RF	0.2992	0.3095	0.3262	0.3409	0.3642	0.1681	0.2356	0.2859	0.3223	0.3866
RNet	0.2791	0.2957	0.3133	0.3271	0.3536	0.1177	0.2044	0.2554	0.2908	0.3573
CA	0.2911	0.3031	0.32	0.3335	0.3582	0.1512	0.2214	0.2762	0.3029	0.3666
L2LR	0.2858	0.2991	0.315	0.3295	0.3562	0.1331	0.2097	0.2599	0.3001	0.368
LMART	0.2901	0.3019	0.3192	0.3355	0.361	0.1636	0.2311	0.2806	0.3183	0.3801

Table 6.4: Upper & Expected Value Bound Normalized DCG (V_1,V_2,) scores of different LETOR methods for variable k: Each cell shows a particular DCG^{UE}_{V} score with a particular k on MQ2007 data-set

for $k = 5$ and $k = 30$ respectively with an increase of 74.6% (Table 6.1, “MSLR-WEB30K” data-set).

Next, we computed the expected $nDCG$ score for each query according to equation 6.1. Figure 6.1 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.

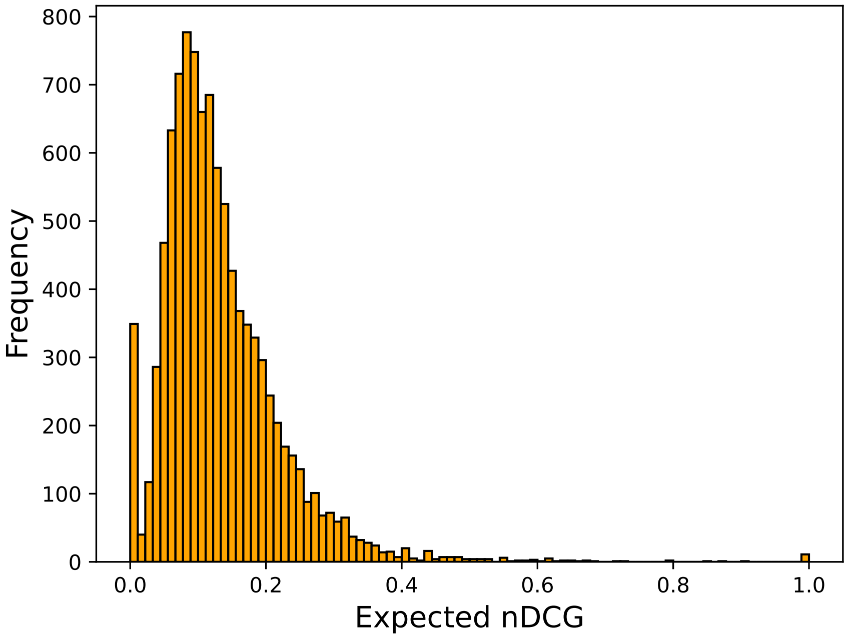


Figure 6.1: Histogram of expected $nDCG$ scores of $10,000$ queries from the “MSLR-WEB30K” data-set

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 $1,000$ 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 $1,000$ (500) queries where the LETOR methods struggle to perform better than the random baseline.

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 6.5: 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.

- **Ideal Query-set:** These are the top $1,000$ 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 $1,000$ (500) queries where the LETOR methods outperforms the random baseline by the largest margin.

6.3 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 6.5 shows the Kendall's τ rank correlations between two rankings induced by $nDCG$ and DCG^{UE} scores in *All*, *Uninformative* or *Ideal* query 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".

6.4 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

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

Table 6.6: 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.

relative performance between two competing LETOR methods. Specifically, we followed the *bootstrap* Studentised Test (student's t-test) from [69] 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 (${}^{8}C_{2} = 28$ 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 6.6 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* [69]. These are two popular methods for comparing evaluation measures.

6.5 Distinguishability of UE in nDCG:

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 [16, 97, 69]. 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 [72]. We again follow [69] 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 6.7 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.

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 6.7: 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.

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.

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 $nDCG$:

$$PAD(nDCG) = \frac{|nDCG^{avg}_{M_1} - nDCG^{avg}_{M_2}|}{\max\left(nDCG^{avg}_{M_1}, nDCG^{avg}_{M_2}\right)} \times 100\% \tag{6.2}$$

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 6.8 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.

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

Table 6.8: 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.

6.6 Consistency of UE in nDCG:

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* [69] 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 6.9 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).

Metric	Swap Rate		
	All	**Uninform.**	**Ideal**
nDCG	0.107	0.420	0.350
$\mathbf{DCG^{UE}_{V_1}}$	0.107	0.420	0.350
$\mathbf{DCG^{UE}_{V_2}}$	0.107	0.420	0.350

Table 6.9: Swap rates between method ranks on *All/ uninform/Ideal* queries across "MSLR-WEB30K" and "MQ2007" data-sets.

Metric	Swap Rate	
	MSLR-WEB30K	**MQ2007**
nDCG	0.210	0.500
$\mathbf{DCG^{UE}_{V_1}}$	0.250	0.500
$\mathbf{DCG^{UE}_{V_2}}$	0.210	0.500

Table 6.10: Swap rates between method ranks on *MSLR-WEB30K/MQ2007* data-sets across "uninformative" and "Ideal" query collections.

Table 6.10 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").

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., [85, 25, 54]); 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 [56] 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 6.11 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 [85], 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 6.12 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.

6.7 Chapter Summary

This chapter demonstrates the application of using our proposed framework on *nDCG*. We first provide how to compute a reasonable Expected value bound of *DCG*, then theoretically

	Swap Rate
Metric	**MSLR-WEB30K**
nDCG	0.25
$\mathbf{DCG^{UE}_{V_1}}$	0.25
$\mathbf{DCG^{UE}_{V_2}}$	**0.25**

Table 6.11: Swap rates between method ranks on MSLR-WEB30K data-sets across “broad” and “focused” query collections.

	Swap Rate
Metric	**MSLR-WEB30K**
nDCG	0.25
$\mathbf{DCG^{UE}_{V_1}}$	0.25
$\mathbf{DCG^{UE}_{V_2}}$	**0.25**

Table 6.12: Swap rates between method ranks on *MSLR-WEB30K* data-sets across “left” and “right” document collections.

prove its correctness. Then we demonstrated the usefulness of expected value normalization in terms of two important perspectives: consistency and discriminative power.

Chapter 7

MAP with Joint Upper & Expected Value Normalization

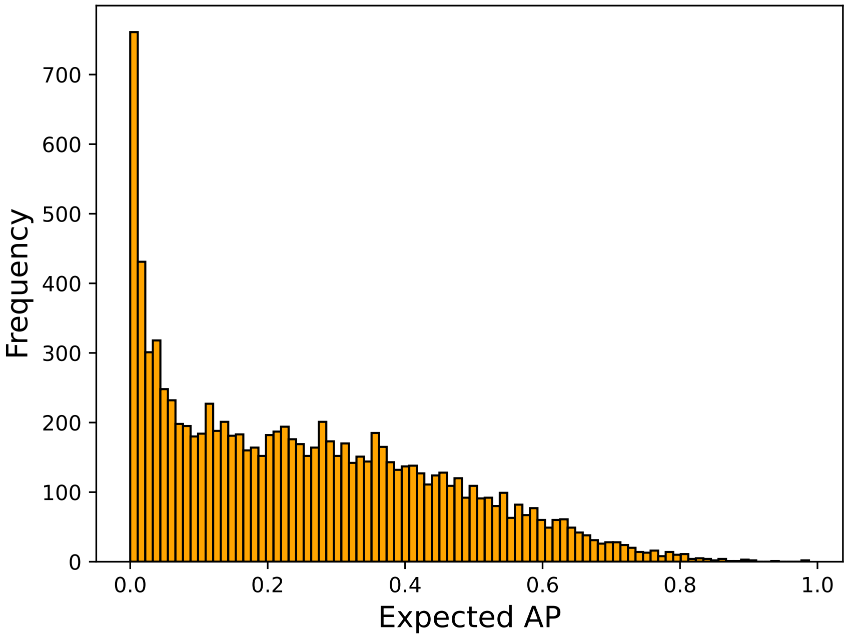


Figure 7.1: Histogram of expected AP scores of $10,000$ queries from the "MSLR-WEB30K" data-set

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. Table 7.1 and 7.2 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. Figure 7.1 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 can not be ignored. Noted that we again created two special sub-sets of queries based on the difference between their Expected AP and and average real AP obtained by eight LETOR methods to define **Uninformative query-set** and **Ideal query-set**(Details in 6.2).

7.1 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

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

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

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

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:

$$\sum_{j=1}^{i} j\binom{N_p}{j}\binom{N_n}{i-j} = \left(\frac{N_p}{N_p+N_n}\right) i\binom{N_p+N_n}{i} \tag{7.1}$$

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, equation 7.1 is true for i = 1

Induction step: Now, Let's assume equation 7.1 is true for i = i-1, then we get the following:

$$\sum_{j=1}^{i-1} j\binom{N_p}{j}\binom{N_n}{i-1-j} = \frac{N_p}{N_p+N_n}(i-1)\binom{N_p+N_n}{i-1} \tag{7.2}$$

Now, $\sum\limits_{\boldsymbol{j=1}}^{\boldsymbol{i}} \boldsymbol{j}\binom{\boldsymbol{N_p}}{\boldsymbol{j}}\binom{\boldsymbol{N_n}}{\boldsymbol{i-j}} = \sum\limits_{j=1}^{i-1} j\binom{N_p}{j}\binom{N_n}{i-j} + i\binom{N_p}{i}$

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

$$= \left[\sum_{j=1}^{i-1} j\binom{N_p}{j}\binom{N_n+1}{i-j}\right] + i\binom{N_p}{i} - \left[\sum_{j=1}^{i-1} j\binom{N_p}{j}\binom{N_n}{i-j-1}\right]$$

$$= \sum_{j=1}^{i} j\binom{N_p}{j}\binom{N_n+1}{i-j} - \left(\frac{N_p}{N_p+N_n}\right)(i-1)\binom{N_p+N_n}{i-1} \qquad \text{[From (7.2)]}$$

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

$As,\left[j\binom{N_p}{i} = N_p\binom{N_p-1}{j-1}\right]$

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

$$= N_p\binom{N_p+N_n}{i-1} - \left(\frac{N_p}{N_p+N_n}\right)(i-1)\binom{N_p+N_n}{i-1}$$

$$= \binom{N_p+N_n}{i-1}\left(\frac{N_p}{N_P+N_n}\right)[N_p+N_n-i+1] = \left[(N_p+N_n-i+1)\binom{N_p+N_n}{i-1}\right]\left(\frac{N_p}{N_p+N_n}\right)$$

$$= \boldsymbol{i}\binom{\boldsymbol{N_p}+\boldsymbol{N_n}}{\boldsymbol{i}}\left(\frac{\boldsymbol{N_p}}{\boldsymbol{N_p}+\boldsymbol{N_n}}\right)$$

Proof completed because

$$(n-r+1)\binom{n}{r-1} = r\binom{n}{r}$$

7.2 Expected value normalized *MAP* yields different rankings compare to Original MAP for *Uninformative* query-set:

Table 7.5 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 $MSP^{UE}_{V_1}$

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 7.1: MAP scores of different LETOR methods for variable k on 'MSLR-WEB30K' data-set.

Method	MAP@				
	5	**10**	**15**	**20**	**30**
ARank	0.3066	0.2923	0.302	0.3173	0.3624
LNet	0.3379	0.3233	0.3328	0.3468	0.3905
RBoost	0.3467	0.3366	0.3477	0.3636	0.4035
RF	0.3674	0.352	0.3585	0.3736	0.414
RNet	0.3281	0.3175	0.3275	0.3443	0.3878
CA	0.3597	0.3457	0.356	0.3716	0.4127
L2LR	0.3543	0.3386	0.3458	0.3607	0.404
LMART	0.3582	0.3459	0.3539	0.3692	0.4101

Table 7.2: MAP scores of different LETOR methods for variable k on 'MQ2007' data-set.

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*.

7.3 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 7.6 summarizes the number of disagreements between MAP and MSP^{UE} in two data-sets. For instance, based on student's t-test, $MSP^{UE}_{V_2}$ disagreed with original MAP on 36

	$MSP^{UE}_{V_1}@$					$MSP^{UE}_{V_2}@$				
Method	**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 7.3: Upper & Expected Value Bound Normalized MSP (V_1,V_2) scores of different LETOR methods for variable k: Each cell shows a particular MSP^{UE}_{V} score with a particular k on MSLR-WEB30K data-set

	$MSP^{UE}_{V_1}@$					$MSP^{UE}_{V_2}@$				
Method	**5**	**10**	**15**	**20**	**30**	**5**	**10**	**15**	**20**	**30**
ARank	0.2366	0.219	0.2222	0.2287	0.2478	0.0392	0.0778	0.0116	0.14	0.1905
LNet	0.2676	0.2492	0.2519	0.257	0.2744	0.0909	0.1315	0.1638	0.1846	0.2257
RBoost	0.2738	0.2603	0.2647	0.2714	0.2853	0.123	0.154	0.188	0.213	0.2513
RF	0.2914	0.2732	0.2739	0.28	0.2941	0.1586	0.1904	0.206	0.226	0.2729
RNet	0.259	0.2443	0.2476	0.2552	0.2724	0.085	0.1308	0.1567	0.1825	0.222
CA	0. 2863	0.2689	0.2728	0.2794	0.2941	0.1422	0.1741	0.198	0.2204	0.2584
L2LR	0.2806	0.2622	0.2633	0.2693	0.2861	0.1232	0.1548	0.1846	0.2093	0.2543
LMART	0.2829	0.2673	0.2691	0.2755	0.2905	0.1541	0.1949	0.2138	0.2369	0.2725

Table 7.4: Upper & Expected Value Bound Normalized MSP (V_1,V_2) scores of different LETOR methods for variable k: Each cell shows a particular MSP^{UE}_{V} score with a particular k on MQ2007 data-set

	Kendall's τ			
Data-set	**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 7.5: 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	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 7.6: 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	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**

Table 7.7: 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.

(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.

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 $nDCG$.

7.4 Distinguishability of UE in MAP:

We again follow [69] 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 7.7 shows some interesting results of these statistical tests for different query sets in 'MSLR-WEB10K' and""MQ2007" data-sets.

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

Table 7.8: 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..

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 can not 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 6.2 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 6.5).

Table 7.8 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

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

Table 7.9: Swap rates between method ranks on *All/ uniform/Ideal* queries across "MSLR-WEB30K" and "MQ2007" data-sets.

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.

7.5 Consistency of UE in MAP:

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 7.9 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 7.10 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.

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

Table 7.10: Swap rates between method ranks on *MSLR-WEB30K*/*MQ2007* data-sets across "uninformative" and "Ideal" query collections.

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

Table 7.11: Swap rates between method ranks on *MSLR-WEB30K* data-sets across "broad" and "focused" query collections.

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 6.2. Table 7.11 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 7.12, 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 6.2 for definitions of "left" and "right" partitions).

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

Table 7.12: Swap rates between method ranks on *MSLR-WEB30K* data-sets across "left" and "right" document collections.

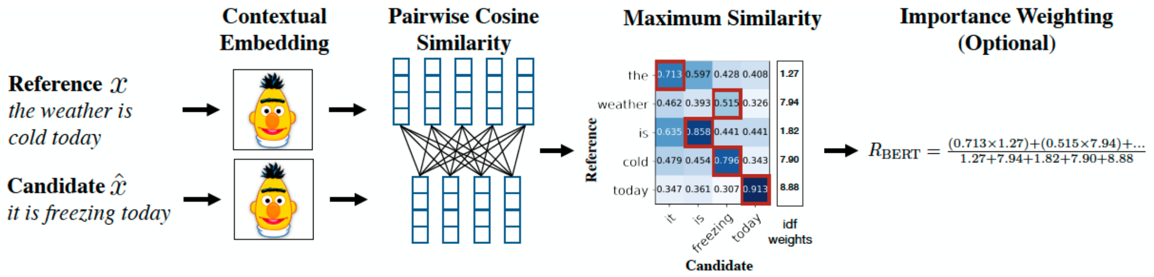
7.6 Chapter Summary

This chapter demonstrates the application of using our proposed framework on our second case study target: *MAP*. We first provide how to compute a reasonable Expected Value of *SP* and theoretically prove its correctness by using induction. Finally, we analyze the implications of these new metrics by comparing them with the original MAP in terms of two important perspectives: consistency and discriminative power.

Chapter 8

BERTScore with Joint Upper & Expected Value Normalization

For our first case study in NLP, we selected one popular evaluation metric for text summarization, **BERTScore**. As we have discussed in the previous chapter, BERTScore utilized contextual embeddings, such as BERT [22] and ELMo [75]. Figure 8.1 illustrates the computation.


Contextual Embedding
Pairwise Cosine Similarity
Maximum Similarity
Importance Weighting (Optional)
Reference x
the weather is cold today
Candidate $\hat{x}$
it is freezing today
Reference
Candidate
idf weights
$R_{\text{BERT}} = \frac{(0.713\times1.27)+(0.515\times7.94)+...}{1.27+7.94+1.82+7.90+8.88}$

	it	is	freezing	today	idf weights
the	0.713	0.597	0.428	0.408	1.27
weather	0.462	0.393	0.515	0.326	7.94
is	0.635	0.858	0.441	0.441	1.82
cold	0.479	0.454	0.796	0.343	7.90
today	0.347	0.361	0.307	0.913	8.88

Figure 8.1: Illustration of the computation of the recall metric R_{BERT}. Given the reference x and candidate $\hat{x}$, BERTScore leverages BERT embeddings and pairwise cosine similarity. [100]

8.1 Heterogeneous Vs Homogeneous:

To better explain our proposed expected value normalization in BERTScore, we also define the following two types of document which we will use throughout the thesis:

1. **Heterogeneous document:** These are documents with heterogeneous contextualization, where the contextualized word embeddings are different from the document vector in a maximum margin.
2. **Homogeneous document:** These are documents with homogeneous contextualization, where contextualized word embeddings are aligned with the document vector in a maximum margin.

Figure 8.2 shows an illustration of the sensitivity of our defined two types of documents: heterogeneous document (HeteDoc) and homogeneous document(HomoDoc). For heterogeneous documents, the difference between two expected BERTScores which are generated by sorting words (from original source document) from Best (most similar) to Worst (least similar) (or vice versa) is minimum. Which we call the difference between B2W (Best 2 worst) and W2B(Worst 2 Best). While the homogeneous documents, the difference is maximum. As we can see, the heterogeneous documents essentially are order-insensitive documents because changing the order of words can not change the expected BERTScore while the homogeneous documents are order-sensitive documents in which reordering words would generate very different BERTScore.

After the sorting, we then calculated the similarity between contextualized word embedding and document embedding and calculated the standard deviation along the most heterogeneous documents and most homogeneous documents, we found that for heterogeneous documents, there is a higher variance in the similarity between contextualized word embedding and document embedding compared with homogeneous documents.

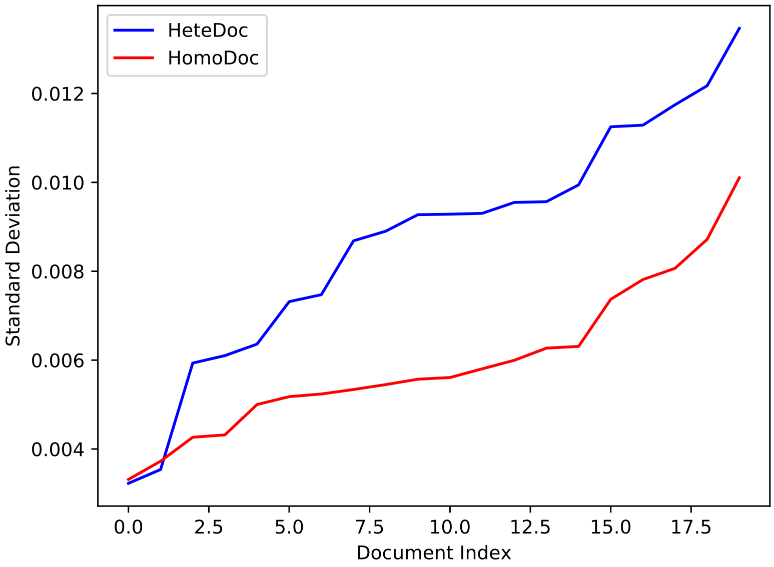


Figure 8.2: Document types with different order sensitivity.

8.2 Expected BERTScore:

As we can see from the figure 8.1, BERTScore essentially calculated the contextual information at a token level and tried to compare the similarity of the semantic knowledge between reference and candidate. An initiative idea to generate an expected summary of original source document is to select these contextualized keywords that contain the maximum information (most similar to the contextualized document information). For instance, given a sentence: ”It is freezing today”, the word ”freezing” should be more important than ”it” in terms of the information provided. Thus, we propose our greedy algorithm 1 to generate the expected summary from a source document. Note that this greedy algorithm just generates the text summary based on the idea of BERTScore, then we can directly use this expected summary to calculate the expected BERTScore:

Algorithm 1 Greedy algorithm to generate the Expected Summary

```
Require: : Hyperparameter: length of generated summary $k$
  $D \leftarrow dictionary$
  $S \leftarrow EmptyString$
  Calculate the Sentence Embedding of Source Document
  for <each token in source document> do
    <Calculate Cosine Similarity between token embedding and
  Sentence Embedding>
    <D[token] = similarity>
  end for
  Sort the Dictionary by value in an increasing order
  for <selection first k token in Dictionary > do
    <append token to S>
  end for
  Output is the expected summary $S$
```

Algorithm 1 explained how to generate the expected summary based on the source document. First, we use pre-trained models such as BERT or RoBERTa to get the sentence embedding of the source document. Then we calculate the similarity between source document embedding and each token embedding, greedily using the first *K* similar (where *K* is a hyperparameter that determines how long summary we want to generate) tokens to generate the expected summary. Although our expected summary may not be meaningful from a human perspective (no sequence), since BERTScore calculates the contextual information [100], our generated summary can be considered an expected version from an embedding level. Then we use the expected summary and the given golden reference to calculate the expected BERTScore. Using our proposed V1 and V2 framework, we can get the Upper expected value normalized BERTScore. Note that the range of V1 is between 0 to 1 and the range of V2 is between -1 to 1.

8.3 UE normalized BERTScore increase the Human correlation

This section discusses some observed differences between the original $BERTScore$ and proposed $BERTScore^{UE}$. Since NLP and IR are two different domains and the method to demonstrate the performance of evaluation metrics are totally different. We utilize the most widely used perspective, that is human correlation, to show the performance of UE normalization in BERTScore. For deeper analysis, we also created two special sub-sets of source documents: i.e., 1) *diverse* document-set and 2) *uniform* document-set, based on how many special words (keyword) exist in the source document. The special words are the most informative words that can determine the contextual information of entire sentence embedding. We follow [24] to use the 100 annotated summarization data-set that were randomly picked from CNN/DailyMail test set and evaluate our proposed metrics as well as original BERTScore metrics from four dimensions. Specifically, we use hashcode = “roberta-large-L17-no-idf-version=0.3.12(hug-trans=4.254.0)” in the computation of BERTScore.

Table 8.1 demonstrated the human correlation of original BERTScore and our proposed Upper expected value normalized BERTScore from 4 perspectives, averaged 100 source documents, and 12 abstractive summarization models. Clearly, we can see our metrics achieve higher human correlation in general. For instance, for our BERTScore V1, human correlation w.r.t. coherence improved from 0 to 0.03 and the same for Fluency (improved from 0.167 to 0.25) and Relevance (-0.03 to 0.06). Although, V1 did not achieve higher human correlation in Consistency (dropped from $-0,106$ to $-0,198$). Our V2 essentially competed against the original BERTScore for all four perspectives in a maximum range, especially for consistency and relevance, original BERTScore achieved a negative correlation with human annotation while UE BERTScore V2 achieves a much higher correlation (0.045 and 0.18 respectively).

Human Correlation For 100 documents				
	Coherence	Consistency	Fluency	Relevance
BERTScore	0	-0.106	0.167	-0.03
BERTScore V1	0.03	-0.198	0.25	0.06
BERTScore V2	0.15	0.045	0.44	0.18
PAI V1	3	-86.79	49.7	300
PAI V2	15	142	163	700
Average of V1 and V2	160			

Table 8.1: Human Correlation of summaries along four evaluation dimensions averaged 100 documents and 12 abstractive summarization models. The difference between V1/V2 and BERTScore is the PAI and we calculate the average PAI of the two versions

We also calculated the percentage absolute increase (PAI) of our proposed version compared with the original BERTSocre. Mathematically, we used the following formula 8.1 for percentage absolute increasing (PAI) in terms of original $BERTScore$, the formula to calculate the PAI for V2 is the same and omits in this case:

$$PAI(V1) = \frac{BERTScoreV1 - BERTScore}{|BERTScore|} \times 100\% \tag{8.1}$$

8.4 UE normalized BERTScore has maximum impact in Heterogeneous document than Homogeneous document

To better understand the implication of the Upper expected value normalization impact on BERTScore, we want to know for which kind of document, our UE normalization can involve the maximum impact. More specifically, according to the definition of section 3.6, for the 100 documents, we sort those documents based on how the contextualized word embeddings are aligned with the document embedding which we define the **Heterogeneous documents** are documents with heterogeneous contextualization, where the contextualized word embeddings are different from the document vector in a maximum margin. On the other hand, **Homogeneous document** set includes documents with homogeneous contextualization, where contextualized word embeddings are aligned with the document vector in a maximum margin. Below is the algorithm 2 to show the process to select Heterogeneous documents Vs Homogeneous documents:

Algorithm 2 Algorithm to sort the document based on the number of keywords that exist

```
Require: : Hyperparameter: length of generated summary $k$
  $D \leftarrow dictionary$
  for <each document in source documents> do
     <Get the most k similar words>
     <Get the least k similar words>
     <use Algorithm  1 to get the expected BERTScores based on
  most k and last k similar words>
     <get the absolute difference between two Expected
  BERTScores, that's the difference>
     D[document] = difference
  end for
  Sort the Dictionary by value in an increasing order
  Output is the dictionary of documents with an order from most "Heterogeneous document"
  to "Homogeneous document"
```

From algorithm 2, we first select the most k and last k words from original documents that are most/least relevant to the source document based on how similar the contextualized word embedding to the document embedding is to generate the expected document. If a document is a heterogeneous contextualization, which means the words in the document are heterogeneously contextualized with the entire document, the difference between B2W and W2B(see the definition in section 3.6) is minimum, indicating these documents are order-insensitive because shuffling the order of words from these document does not change the expected BERTScore much, in which our expected value normalization should involve the maximum impact because a random selection can amplify the order sensitivity of Heterogeneous documents which help the human correlation with human judgment. On the other hand, the UE normalization should have minimum impact on the Homogeneous documents or even negative performance because those documents are homogeneous contextualized which are already order sensitive.

After the sorting, we also call "the most Heterogeneous documents" as "First 20 documents " and "the most Homogeneous documents" as "Last 20 documents". Table 8.2 and 8.3 show the Upper expected value normalization impact on the first 20 documents and last 20 documents if we use the algorithm 2 to sort the documents. We also use the average of PAI to indicate the impact of upper expected value normalization. First, we compare these two tables: PAI in V1 of the first 20 documents achieves the maximum scores. For example, the UE BERTScore V2 gets the relevance human correlation as 0.303, resulting 605 PAI score. The

Human Correlation For most 20 HeteDoc Document				
	Coherence	Consistency	Fluency	Relevance
BERTScore	-0.198	0.03	0.29	-0.06
BERTScore V1	-0.076	0.09	0.351	0.06
BERTScore V2	0.106	0.151	0.412	0.303
PAI V1	61	200	21	200
PAI V2	153	403	42	605
Average of V1 and V2	210			

Table 8.2: Human Correlation of summaries along four evaluation dimensions averaged the most 20 HeteDoc documents and 12 abstractive summarization models. The difference between V1/V2 and BERTScore is the PAI and we calculate the average PAI of the two versions

Human Correlation For most 20 HomoDoc Documents				
	Coherence	Consistency	Fluency	Relevance
BERTScore	-0.18	-0.3	-0.2	-0.24
BERTScore V1	-0.18	-0.24	-0.11	-0.3
BERTScore V2	-0.21	-0.39	-0.078	-0.39
PAI V1	0	20	45	-25
PAI V2	-16	-30	61	-62.5
Average of V1 and V2	-1.02			

Table 8.3: Human Correlation of summaries along four evaluation dimensions averaged the most 20 HomoDoc documents and 12 abstractive summarization models. The difference between V1/V2 and BERTScore is the PAI and we calculate the average PAI of the two versions

average PAI of two versions is 210 for the first 20 documents, which proves our hypothesis that our UE normalization has maximum impact for Heterogeneous documents. On the other hand, for the last 20 documents (Homogeneous documents), the average PAI is -1.02 and essentially lowers the original BERTScore performance in human correlation from multiple perspectives, such as relevance and consistency, showing that for the document with homogeneous contextualization, our UE normalization would provide opposite impact.

Second, we compare Table 8.1 and Table 8.2. In the overall document set, our UE normalization has a positive impact (PAI is 160). For the first 20 cases, our UE normalization achieves even higher PAI which is because the first 20 are the most diverse documents. Although the UE normalization has a negative impact on the last 20 documents, essentially it will increase the human correlation for the entire document set.

	Win	Lost	Tie
V1 Vs BERTScore	24%	10%	66%
V2 Vs BERTScore	24%	10%	66%

Table 8.4: Statistics for UE normalized BERTScore wins, losses, and ties against BERTScore. Results report the average of 5 pairs (BERTbase vs. MobileBERT, MobileBERT vs. DistilBERT, DistilBERT vs. RoBERTa, RoBERTa vs. XLNet and XLNet vs. GPT-2) evaluated by humans.

8.5 Human judgment favors UE normalized BERTScore

We next took a deeper look into the cases where UE normalized BERTScore disagreed with the original BERTScore while comparing two extractive summarization models. We asked humans to blindly evaluate the quality of the summaries generated by two models and make a judgment on which summary was better as suggested by [60, 1]. Specifically, we considered 5 pairs of models (BERTbase vs. MobileBERT, MobileBERT vs. DistilBERT, DistilBERT vs. RoBERTa, RoBERTa vs. XLNet, and XLNet vs. GPT-2) and provided humans with outputs for each pair of models, hiding the model's name. We asked the annotators to say which extractive summary is better and matched their decision against both BERTScore and two UE normalized BERTScore's conclusions. Our annotators were three doctoral students all working in NLP. We took the majority voting judgment from annotators and the results are reported in Table 8.4. As summarized in Table 8.4, blind evaluation by humans indicated UE normalized BERTScore was more accurate than original BERTScore in the case of disagreements between the two, thus confirming that UE normalized BERTScore captures semantics better than BERTScore.

8.6 Explanation of UE normalization in BERRScore from Attention perspective

Attention mechanisms [4] are the fundamental component in NLP tasks such as text generation. The purpose of attention is to allow a model to focus on specific parts of the input when processing or generating the output. [82] shows a simple network architecture based solely on attention mechanism and achieves huge improvement on two machine translation tasks. While people debate the relationship between utilizing attention and its impact on performance [88, 6, 81], people still use attention to understand the philosophy of inner part of deep neural network [83]. We specifically discovered the last layer and average of 16 heads attention of our encoder (RoBERTa-large) and calculated the similarity between each contextualized token attention with the document vector, then we calculated their standard deviation/ mean score, for Heterogeneous document and Homogeneous document respectively.

Figure 8.3 illustrates the information of attention inside the encoder. However, we can not claim that there is a huge difference between the distribution of attention within two opposite document types from this figure. For instance, according to the standard deviation, both two document types show an increasing trend to an identical degree.

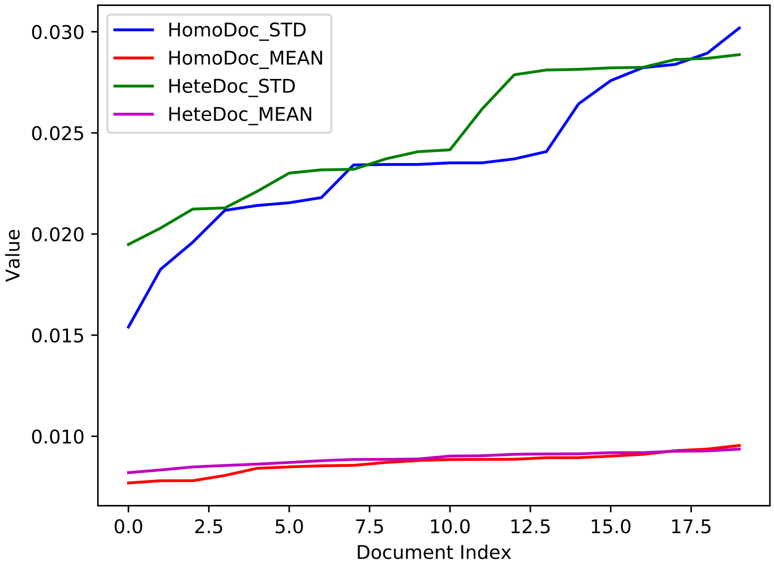


Figure 8.3: Heterogeneous/Homogeneous documents Standard Deviation and the mean score of attention distribution in the source document. Heterogeneous document Standard Deviation and mean score are slightly higher than the same of Homogeneous document for most documents

8.7 Hypothesis for the improvement of UE normalization in BERTScore.

As we can see the empirical experimental results from 8.2 and 8.3. Our proposed UE normalization can involve boosting improvement in terms of documents with heterogeneous contextualization (given the PAI score of 210) while slightly negative performance in terms of documents with homogeneous contextualization (given the PAI score of -1.02). Due to the order insensitivity of the Heterogeneous document, our randomized expected normalization can amplify the order sensitivity of the Heterogeneous document which helps the human correlation with human judgment. On the other hand, the Homogeneous document itself exhibits sensitivity to order, making further normalization (UE normalization) resulting in a slightly negative performance. Noted that this is a reasonable hypothesis but still needs to be proved in our further experiments.

8.8 Chapter Summary

In this chapter, we implemented our general Upper expected value normalization framework to a widely used text summarization metric, BERTScore. We found that UE normalization is able to greatly increase the human correlation of BERTScore while comparing the abstractive summarization methods. Meanwhile, human judgment favors UE normalization while comparing extractive summarize. However, the interpretability of the boosting improvement is still unclear based on the current implementation. Although people use attention to show the correlation between the inner part and downstream performance, in our case, we can not use attention to explain. Further experiments are needed to better understand the secret behind the empirical results.

Chapter 9

ROUGE with Joint Upper & Expected Value Normalization

The previous chapter has demonstrated that our expected value normalization involves impact if a metric is order sensitive. To better prove our hypotheses, in this chapter, we focused on ROUGE, a widely used simple metric for text generation tasks without considering the order of generation. We first introduce how to calculate the expected ROUGE score, then show the experimental results.

9.1 Expected ROUGE:

As we have discussed in the previous chapter, we specifically use a unigram language model that has been trained on source documents to generate the expected summary, then we use the expected summary to calculate a ROUGE score, which is our expected ROUGE. To make the generation process more general, we do not use the history information while directly utilizing the occurrence of each word. The unigram model is defined as follows:

$$p(w_i|w_1...w_{i-1}) \approx p(w_i) = \frac{c(w_i)}{\sum_{\tilde{w}} c(\tilde{w})} \tag{9.1}$$

9.2 UE normalized ROUGE not help in Human correlation

First, we focus on the human correlation of abstractive summarization. Table 9.1 demonstrated the human correlation of original ROUGE and our proposed Upper expected value normalized ROUGE from 4 perspectives, averaged 100 source documents, and 12 abstractive summarization models. Compared with UE normalization in BERTScore, we can see our proposed metric

Human Correlation For 100 documents				
	Coherence	Consistency	Fluency	Relevance
ROUGE	0.08	0.09	0.04	0.13
ROUGE V1	0.08	0.1	0.04	0.12
ROUGE V2	0.08	0.1	0.4	0.12
PAI V1	-1.9	1.12	8.34	-5.2
PAI V2	-0.84	1.5	6.2	-3
Average of V1 and V2	0.77			

Table 9.1: Human Correlation of summaries along four evaluation dimensions averaged 100 documents and 12 abstractive summarization models. The difference between V1/V2 and ROUGE is the PAI and we calculate the average PAI of the two versions

essentially generate identical results against the original ROUGE score. Specifically, except for consistency, UE normalized ROUGE v1 and v2 are equal to the human correlation for 100 documents for the other three perspectives, resulting in the final PAI (for PAI the definition, see section 8.1) score of 0.77.

Noted that we did not specifically select the Heterogeneous document and Homogeneous document while conducting the ROUGE score because the calculation of ROUGE does not involve the contextualized word/document embedding, thus we did not implement similar ablation experiments here.

9.3 Human Judgement favors UE normalized ROUGE

Although we did not observe the improvement of UE normalization in ROUGE in terms of abstraction summarization method human correlation, we next took a deeper look into the cases where UE normalized ROUGE disagreed with the original ROUGE while comparing two extractive summarization models. We again utilize our collected human annotations for the 5 pairs of models (BERTbase vs. MobileBERT, MobileBERT vs. DistilBERT, DistilBERT vs. RoBERTa, RoBERTa vs. XLNet, and XLNet vs. GPT-2) and match the human decision against both ROUGE and two UE normalized ROUGE's conclusions. We also took the majority voting judgment from annotators and the results are reported in Table 9.2. As summarized in Table 9.2, blind evaluation by humans indicated UE normalized ROUGE was more accurate than the original ROUGE in the case of disagreements between the two, thus confirming that UE

	Win	Lost	Tie
V1 Vs ROUGE	32%	12%	56%
V2 Vs ROUGE	32%	14%	54%

Table 9.2: Statistics for UE normalized ROUGE wins, losses, and ties against ROUGE. Results report the average of 5 pairs (BERTbase vs. MobileBERT, MobileBERT vs. DistilBERT, DistilBERT vs. RoBERTa, RoBERTa vs. XLNet and XLNet vs. GPT-2) evaluated by humans.

normalized ROUGE captures semantics better than ROUGE. Essentially, we can observe that UE normalized ROUGE wins original ROUGE more than UE normalized BERTScore wins original BERTScore in both our two versions (32% Vs 24%).

9.4 Chapter Summary

This chapter summarized the experimental results of the implementations of upper expected value normalization on another popular NLP domain metric, ROUGE. An interesting observation from ROUGE is although we did not see the improvement from human correlation according to the four perspectives, UE normalized ROUGE essentially beats original ROUGE from our human judgment where the human judgment reflects a more general understanding from a human perspective. A further interesting study could focus on finding which type of documents that UE normalized ROUGE would involve the maximum positive impact.

Chapter 10

Discussions

In this thesis, we presented a novel perspective on the evaluation of Information Retrieval (IR) and Natural Language Processing systems(NLP). Specifically, we performed two case study on *nDCG* and *MAP* (IR), both are widely popular metrics for IR evaluation, and two case studies on *ROUGE* and *BERTScore* for text summarization task. We started with the observation that traditional *nDCG* and *MAP* computation does not include a query-specific expected value normalization although they include a query-specific upper-bound normalization. For *ROUGE* and *BERTScore*, there is neither upper nor expected value normalization. In other words, the current practice is to assume a uniform expected value (zero) across all queries while computing *nDCG* and *MAP*, and a uniform expected value across all documents while computing *ROUGE* and *BERTScore*, an assumption that is incorrect.

This limitation raises a question mark on the previous comparative studies involving multiple ranking/summarization 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. In the NLP domain, documents with heterogeneous contextualization and documents with homogeneous contextualization are treated equally as well, even though their order sensitivity is different. *How can we incorporate query-specific (instance level) expected value normalization into IR/NLP evaluation metrics and how will it impact IR/NLP evaluation in general?* This is the central issue we investigated in this thesis.

Conceptual Leap: To address the aforementioned issue, we proposed to penalize the traditional evaluation metric score of each query with an expected value normalization term specific to that instance. 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, and two summarization metric $BERTScore$ essentially creating eight new evaluation metrics.

The next challenge in the IR domain was to estimate a more realistic query-specific expected value for the 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.

For the challenge in the NLP domain, we also have to consider the expected value accordingly. In BERTScore, which utilizes contextualized word embedding similarity, intuitively, to generate a summarization, the most important words should be selected. Based on this simple hypothesis, we greedily select the most important/dominant words from the source document and use them to generate the expected summary, then calculate the expected BERTScore based on the expected summary. Due to the simple overlapping consideration in ROUGE calculation, we also use the unigram language model which is trained from each instance to generate a expected summary that can be directly used to calculate the expected ROUGE score.

Depth of Impact: For IR, we use two publicly available web search and learning-to-rank datasets to conduct extensive experiments with eight popular LETOR methods to understand the implications DCG^{UE} and MSP^{UE}. For NLP, we use the most recent human-annotated dataset, 12 abstractive summarization, and 6 extractive summarization methods to test the human correlation from 4 perspectives: Coherence, Consistency, Fluency, and Relevance. We also have 3 NLP Ph.D. experts to provide the extractive human judgment annotation.

The implications of IR 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 the 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 the 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).

In *ROUGE* and *BERTScore*, the implications are briefly summarized as follows:

1. Upper expected value normalized BERTScore increases the **human correlation** from 4 important perspectives.
2. Based on empirical experimental results, upper expected value normalized BERTScore has maximum impact in the document with heterogeneous contextualization than a document with homogeneous contextualization. Although we can not properly explain this performance from a deeper perspective (such as attention mechanisms).
3. **Human judgment** favors Upper expected value normalized BERTScore and ROUGE score across 5 pair-wise extractive summarizer comparisons.

Breadth of Impact: The proposed expected value normalization technique is very general and can be potentially extended to other evaluation metrics like ERR (in IR) and BARTScore (in NLP), which is an exciting future direction.

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

Chapter 11

Future Plan

As we can see from this thesis that our proposed upper and expected value normalization framework is quite general and easy to be implemented in many different domains. Thus, an intriguing future direction will be to investigate UE normalization for additional evaluation metrics such as ERR in IR and BARTScore in NLP. This particular direction consists of the following sub-tasks: 1) propose a reasonable expected value normalization term of a particular metric, 2) introduce both upper and expected value normalization terms into the original metric, 3) conduct the proposed metric to evaluate different text summarization results of several models. 4) systematically compare the original metric with the new metric from different perspectives and conclude the implications.

One limitation of current experiments in BERTScore is the difficulty of explanation. Although we have achieved a huge improvement in terms of human correlation and also discovered the particular documents that our UE normalization would involve maximum impact, we did not explain it from a deeper level, that is, which is one important direction we want to do in the future.

Another limitation of expected value normalization is the expected value should be derived separately for different evaluation metrics, which requires domain knowledge of different metrics. Thus, one interesting follow-up work is to analyze a general expected value in a particular task. That is, task-specific instead of metric-specific.

Meanwhile, since we have been doing the evaluation metric and expected value normalization during my Ph.D. program for a long time, we really want to do a survey about the normalization technique in different domains.

References

[1] M. Akter, N. Bansal, and S. K. Karmaker. Revisiting automatic evaluation of extractive summarization task: Can we do better than rouge? In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1547–1560, 2022.

[2] N. Asadi and J. Lin. 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*, pages 997–1000, 2013.

[3] J. A. Aslam, E. Yilmaz, and V. Pavlu. 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*, pages 27–34, 2005.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[5] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. User variability and IR system evaluation. In R. Baeza-Yates, M. Lalmas, A. Moffat, and B. A. Ribeiro-Neto, editors, *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, August 9-13, 2015*, pages 625–634, Santiago, Chile, 2015. ACM.

[6] J. Bastings and K. Filippova. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? *arXiv preprint arXiv:2010.05607*, 2020.

[7] B. Billerbeck and J. Zobel. Questioning query expansion: An examination of behaviour and parameters. In K. Schewe and H. E. Williams, editors, *Database Technologies 2004, Proceedings of the Fifteenth Australasian Database Conference, ADC 2004, 18-22 January 2004*, volume 27 of *CRPIT*, pages 69–76, Dunedin, New Zealand, 2004. Australian Computer Society.

[8] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[9] M. Buckland and F. Gey. The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12–19, 1994.

[10] C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *ACM SIGIR Forum*, volume 51, pages 235–242. ACM New York, NY, USA, 2017.

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

[12] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hullender. Learning to rank using gradient descent. In L. D. Raedt and S. Wrobel, editors, *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pages 89–96, Bonn, Germany, 2005. ACM.

[13] Z. Cao, T. Qin, T. Liu, M. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In Z. Ghahramani, editor, *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 129–136, Corvallis, Oregon, USA,, 2007. ACM.

[14] C. Caragea, V. Honavar, P. Boncz, P. Larson, S. Dietrich, G. Navarro, B. Thuraisingham, Y. Luo, O. Wolfson, S. Beitzel, et al. Mean average precision. *Encyclopedia of Database Systems*, page 1703, 2009.

[15] J. Chen, Y. Liu, J. Mao, F. Zhang, T. Sakai, W. Ma, M. Zhang, and S. Ma. Incorporating query reformulating behavior into web search evaluation. In *Proceedings of the 30th ACM International Conference on Information &amp; Knowledge Management*, CIKM '21, page 171–180, New York, NY, USA, 2021. Association for Computing Machinery.

[16] J. Chen, Y. Liu, J. Mao, F. Zhang, T. Sakai, W. Ma, M. Zhang, and S. Ma. Incorporating query reformulating behavior into web search evaluation. In *Proceedings of the 30th ACM International Conference on Information &amp; Knowledge Management*, CIKM '21, page 171–180, New York, NY, USA, 2021. Association for Computing Machinery.

[17] N. Chen, F. Zhang, and T. Sakai. Constructing better evaluation metrics by incorporating the anchoring effect into the user model. 2022.

[18] Y.-C. Chen and M. Bansal. Fast abstractive summarization with reinforce-selected sentence rewriting. *arXiv preprint arXiv:1805.11080*, 2018.

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

[20] F. Dernoncourt, M. Ghassemi, and W. Chang. A repository of corpora for summarization. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[21] D. Deutsch and D. Roth. Understanding the extent to which content quality metrics measure the information quality of summaries. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 300–309, 2021.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[23] L. Egghe. 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, 2008.

[24] A. R. Fabbri, W. Kryściński, B. McCann, C. Xiong, R. Socher, and D. Radev. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409, 2021.

[25] G. Faggioli, N. Ferro, and N. Fuhr. Detecting significant differences between information retrieval systems via generalized linear models. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 446–456, 2022.

[26] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[27] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *The Journal of machine learning research*, 4:933–969, 2003.

[28] Y. Ganjisaffar, R. Caruana, and C. V. Lopes. Bagging gradient-boosted trees for high precision, low variance ranking models. In W. Ma, J. Nie, R. Baeza-Yates, T. Chua, and W. B. Croft, editors, *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, July 25-29, 2011*, pages 85–94, Beijing, China, 2011. ACM.

[29] S. Gehrmann, Y. Deng, and A. M. Rush. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*, 2018.

[30] L. Gienapp, M. Fröbe, M. Hagen, and M. Potthast. The impact of negative relevance judgments on ndcg. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2037–2040, 2020.

[31] L. Gienapp, B. Stein, M. Hagen, and M. Potthast. Estimating topic difficulty using normalized discounted cumulated gain. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2033–2036, 2020.

[32] T. Goyal, J. J. Li, and G. Durrett. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*, 2022.

[33] H. Guo, R. Pasunuru, and M. Bansal. Soft layer-specific multi-task summarization with entailment and question generation. *arXiv preprint arXiv:1805.11004*, 2018.

[34] M. Hanna and O. Bojar. A fine-grained analysis of bertscore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517, 2021.

[35] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.

[36] W.-T. Hsu, C.-K. Lin, M.-Y. Lee, K. Min, J. Tang, and M. Sun. A unified model for extractive and abstractive summarization using inconsistency loss. *arXiv preprint arXiv:1805.06266*, 2018.

[37] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[38] Y. Jia, H. Wang, S. Guo, and H. Wang. Pairrank: Online pairwise learning to rank by divide-and-conquer. In *Proceedings of the Web Conference 2021*, pages 146–157, 2021.

[39] J. Jiang and J. Allan. Adaptive effort for search evaluation metrics. In N. Ferro, F. Crestani, M. Moens, J. Mothe, F. Silvestri, G. M. D. Nunzio, C. Hauff, and G. Silvello, editors, *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, March 20-23, 2016. Proceedings*, volume 9626 of *Lecture Notes in Computer Science*, pages 187–199, Padua, Italy, 2016. Springer.

[40] Y. Jiang and M. Bansal. Closed-book training to improve summarization encoder memory. *arXiv preprint arXiv:1809.04585*, 2018.

[41] E. Kanoulas and J. A. Aslam. Empirical justification of the gain and discount function for ndcg. In D. W. Cheung, I. Song, W. W. Chu, X. Hu, and J. J. Lin, editors, *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, November 2-6, 2009*, pages 611–620, Hong Kong, China, 2009. ACM.

[42] S. S. K. Karmaker, P. Sondhi, and C. Zhai. 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, and P. Cudré-Mauroux, editors, *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, October 19-23, 2020*, pages 3281–3284, Ireland, 2020. ACM.

[43] S. K. Karmaker Santu, P. Sondhi, and C. Zhai. 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*, pages 475–484, 2017.

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

[45] W. Kryściński, N. S. Keskar, B. McCann, C. Xiong, and R. Socher. Neural text summarization: A critical evaluation. *arXiv preprint arXiv:1908.08960*, 2019.

[46] W. Kryściński, R. Paulus, C. Xiong, and R. Socher. Improving abstraction in text summarization. *arXiv preprint arXiv:1808.07913*, 2018.

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

[48] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[49] C.-J. Lin, R. C. Weng, and S. S. Keerthi. Trust region newton method for logistic regression. *The Journal of Machine Learning Research*, 9:627–650, 2008.

[50] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[51] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[52] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

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

[54] A. Moffat, F. Scholer, and P. Thomas. Models and metrics: Ir evaluation as a user process. In *proceedings of the seventeenth Australasian document computing symposium*, pages 47–54, 2012.

[55] A. Moffat, P. Thomas, and F. Scholer. Users versus models: what observation tells us about effectiveness metrics. In Q. He, A. Iyengar, W. Nejdl, J. Pei, and R. Rastogi, editors, *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, October 27 - November 1, 2013*, pages 659–668, San Francisco, CA, USA, 2013. ACM.

[56] J. Mothe, L. Laporte, and A.-G. Chifu. Predicting query difficulty in ir: impact of difficulty definition. In *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, pages 1–6. IEEE, 2019.

[57] J.-P. Ng and V. Abrecht. Better summarization evaluation with word embeddings for rouge. *arXiv preprint arXiv:1508.06034*, 2015.

[58] R. Pasunuru and M. Bansal. Multi-reward reinforced summarization with saliency and entailment. *arXiv preprint arXiv:1804.06451*, 2018.

[59] R. Paulus, C. Xiong, and R. Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

[60] M. Peyrard. Studying summarization evaluation metrics in the appropriate scoring range. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5093–5100, 2019.

[61] T. Qin and T. Liu. Introducing LETOR 4.0 datasets. *CoRR*, abs/1306.2597, 2013.

[62] T. Qin, T.-Y. Liu, W. Ding, J. Xu, and H. Li. Microsoft learning to rank datasets. *Retrieved September*, 7:2015, 2010.

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

[64] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[65] P. Ravikumar, A. Tewari, and E. Yang. On NDCG consistency of listwise ranking methods. In G. J. Gordon, D. B. Dunson, and M. Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pages 618–626, Fort Lauderdale, USA, 2011. JMLR.org.

[66] S. E. Robertson, E. Kanoulas, and E. Yilmaz. Extending average precision to graded relevance judgments. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 603–610, 2010.

[67] K. Roitero, E. Maddalena, S. Mizzaro, and F. Scholer. On the effect of relevance scales in crowdsourcing relevance assessments for information retrieval evaluation. *Information Processing & Management*, 58(6):102688, 2021.

[68] H. Saadany and C. Orasan. Bleu, meteor, bertscore: evaluation of metrics performance in assessing critical translation errors in sentiment-oriented text. *arXiv preprint arXiv:2109.14250*, 2021.

[69] T. Sakai. Evaluating evaluation metrics based on the bootstrap. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 525–532, 2006.

[70] T. Sakai. On the reliability of information retrieval metrics based on graded relevance. *Inf. Process. Manag.*, 43(2):531–548, 2007.

[71] T. Sakai. A simple and effective approach to score standardisation. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, pages 95–104, 2016.

[72] T. Sakai, D. Ishikawa, N. Kando, Y. Seki, K. Kuriyama, and C.-Y. Lin. Using graded-relevance metrics for evaluating community qa answer selection. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 187–196, 2011.

[73] T. Sakai and S. Robertson. Modelling A user population for designing information retrieval metrics. In T. Sakai, M. Sanderson, and N. Kando, editors, *Proceedings of the 2nd International Workshop on Evaluating Information Access, EVIA 2008, National Center of Sciences, December 16, 2008*, Tokyo, Japan, 2008. National Institute of Informatics (NII).

[74] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[75] J. Sarzynska-Wawer, A. Wawer, A. Pawlak, J. Szymanowska, I. Stefaniak, M. Jarkiewicz, and L. Okruszek. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304:114135, 2021.

[76] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.

[77] S. Shukla, M. Lease, and A. Tewari. Parallelizing listnet training using spark. In W. R. Hersh, J. Callan, Y. Maarek, and M. Sanderson, editors, *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, August 12-16, 2012*, pages 1127–1128, Portland, OR, USA, 2012. ACM.

[78] T. Sun, J. He, X. Qiu, and X. Huang. Bertscore is unfair: On social bias in language model-based metrics for text generation. *arXiv preprint arXiv:2210.07626*, 2022.

[79] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*, 2020.

[80] N. Tonellotto, C. Macdonald, and I. Ounis. Efficient and effective retrieval using selective pruning. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 63–72, 2013.

[81] M. Tutek and J. Šnajder. Staying true to your word:(how) can attention become explanation? *arXiv preprint arXiv:2005.09379*, 2020.

[82] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[83] J. Vig. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*, 2019.

[84] E. M. Voorhees. Evaluation by highly relevant documents. In W. B. Croft, D. J. Harper, D. H. Kraft, and J. Zobel, editors, *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001*, pages 74–82, New Orleans, Louisiana, USA, 2001. ACM.

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

[86] Y. Wang, L. Wang, Y. Li, D. He, and T. Liu. A theoretical analysis of NDCG type ranking measures. In S. Shalev-Shwartz and I. Steinwart, editors, *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013*, volume 30 of *JMLR Workshop and Conference Proceedings*, pages 25–54, Princeton University, NJ, USA, 2013. JMLR.org.

[87] W. Webber, A. Moffat, and J. Zobel. The effect of pooling and evaluation depth on metric stability. In T. Sakai, M. Sanderson, and W. Webber, editors, *Proceedings of the 3rd International Workshop on Evaluating Information Access, EVIA 2010, National Center of Sciences, June 15, 2010*, pages 7–15, Tokyo, Japan, 2010. National Institute of Informatics (NII).

[88] S. Wiegreffe and Y. Pinter. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*, 2019.

[89] Q. Wu, C. J. Burges, K. M. Svore, and J. Gao. Ranking, boosting, and model adaptation. Technical report, Technical report, Microsoft Research, 2008.

[90] Z. Wu, M. Zhang, M. Zhu, Y. Li, T. Zhu, H. Yang, S. Peng, and Y. Qin. Kg-bertscore: Incorporating knowledge graph into bertscore for reference-free machine translation evaluation. In *Proceedings of the 11th International Joint Conference on Knowledge Graphs*, pages 121–125, 2022.

[91] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In W. Kraaij, A. P. de Vries, C. L. A. Clarke, N. Fuhr, and N. Kando, editors, *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 23-27, 2007*, pages 391–398, Amsterdam, The Netherlands, 2007. ACM.

[92] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[93] E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 102–111, 2006.

[94] E. Yilmaz, E. Kanoulas, and J. A. Aslam. A simple and efficient sampling method for estimating AP and NDCG. In S. Myaeng, D. W. Oard, F. Sebastiani, T. Chua, and M. Leong, editors, *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, July 20-24, 2008*, pages 603–610, Singapore, 2008. ACM.

[95] E. Yilmaz, M. Shokouhi, N. Craswell, and S. Robertson. Expected browsing utility for web search evaluation. In J. Huang, N. Koudas, G. J. F. Jones, X. Wu, K. Collins-Thompson, and A. An, editors, *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, October 26-30, 2010*, pages 1561–1564, Toronto, Ontario, Canada, 2010. ACM.

[96] E. Yilmaz, M. Verma, N. Craswell, F. Radlinski, and P. Bailey. Relevance and effort: An analysis of document utility. In J. Li, X. S. Wang, M. N. Garofalakis, I. Soboroff, T. Suel, and M. Wang, editors, *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, November 3-7, 2014*, pages 91–100, Shanghai, China, 2014. ACM.

[97] H.-T. Yu, A. Jatowt, R. Blanco, H. Joho, and J. M. Jose. An in-depth study on diversity evaluation: The importance of intrinsic diversity. *Information Processing & Management*, 53(4):799–813, 2017.

[98] W. Yuan, G. Neubig, and P. Liu. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277, 2021.

[99] J. Zhang, Y. Zhao, M. Saleh, and P. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.

[100] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[101] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.