Evaluating Relative Retrieval Effectiveness with Normalized Residual Gain

Anonymous Author(s)

ABSTRACT

Traditional evaluation metrics, such as NDCG, have long been the conventional approach for evaluating the effectiveness of information retrieval methods. However, such metrics are focused on absolute measures of effectiveness. While they allow us to compare the absolute performance of one retrieval method to another, we do not know if systems with similar absolute performance achieve this performance by finding the same items, or by finding different items with similar relevance grades. To address this problem, several recent proposals have measured the relative performance of a retrieval method in the context of the results from one or more other methods. In this paper, we address theoretical limitations of these proposals and introduce a new metric called Normalized Residual Gain (NRG) that can be seen as an extension of the underlying absolute metric, rather than as an entirely new metric. Operating in the context of the results retrieved by one or more other methods, NRG adjusts gain values according to the browsing model of the absolute metric. Through testing over the MS MARCO dev small and TREC DL 2019 datasets, we find that higher absolute effectiveness does not necessarily correlate with a higher NRG score, which will vary depending on context. In particular, in the context of modern neural models, NRG suggests that a traditional BM25 ranker continues to find relevant items missed by even the best neural models.

ACM Reference Format:
Anonymous Author(s). 2024. Evaluating Relative Retrieval Effectiveness with Normalized Residual Gain. In *Proceedings of the 2nd International ACM SIGIR Conference on Information Retrieval in the Asia Pacific.* ACM, New York, NY, USA, 8 pages. https://doi.org/XX.XXXX/XXXXXXX.XXXXXXX

1 INTRODUCTION

Information Retrieval (IR) systems are traditionally assessed using offline effectiveness metrics like MRR and NDCG [18]. These well-established effectiveness metrics rely on a fixed set of queries and corresponding relevance judgments over some collection of items. However, it's important to note that these metrics provide an absolute measure of retrieval effectiveness for a single ranking; they do not directly compare the output of one ranker to that of another. When comparing the performance of competing rankers over a query set, this comparison is traditionally based on comparing the absolute value of the metric on each query, or the average over the query set, which may not fully capture the nuances of their performance differences. One crucial aspect that these metrics fail to encompass is the ability to discern whether the relative performance of a particular ranker, when compared to others, derives from a better ordering of the same items, or from the discovery of different, more-relevant items.

In this paper, we develop a metric called *Normalized Residual Gain* (NRG) that allows us to measure the additional value provided by a new ranker, as measured against a set of existing rankers, particularly in terms of its ability to find unique relevant items. A ranker that is known to return distinct relevant items from a collection can provide value, even if its absolute performance is inferior to that of another ranker, for example, by providing insights into paths for further improvements. In addition, if we are building a reusable test collection, a retrieval method that offers unique relevant contributions to the pool of judged items provides value by improving re-usability [31].

Recently, researchers have proposed metrics intended to address this problem [1, 30]. Türkmen et al. [30] introduce a novel metric, *rareness-based precision-at-k*, which incorporates the rareness of items within a group of competing rankings when determining the precision for a new ranking. Arabzadeh et al. [1] introduce a metric known as TaSC. The TaSC metric focuses on quantifying the subspace coverage within a new ranking, when compared against a set of existing rankings. TaSC is designed to assess the extent to which the new ranking represents similar or distinct query subspaces.

While both papers experimentally demonstrate their value, both of these metrics combine scoring functions in *ad hoc* ways, and do not generalize to a full range of metrics. For example, "rareness-based precision-at-k" is an extension of precision@K only. It focuses solely on whether or not other rankers retrieve an item, without taking into account the positions where the rankers rank those items. The TaSC metric also does not consider the position at which items appear in the existing rankings. It works directly with aggregated effectiveness scores, and it does not prioritize items that are relevant but may not have been retrieved by other rankers. Nonetheless, inspired by the ideas in these papers, we take a step back and consider how to approach the problem in a manner consistent with the theoretical searcher browsing models underlying offline effectiveness metrics, proposing NRG as the solution.

In the next section, we derive NRG in terms of a theoretical framework grounded in the principles established by approaches such as C/W/L [3, 23–25][1] and other searcher browsing models [2, 6, 9, 10, 35, 36]. NRG can be viewed as an extension of an existing underlying absolute metric, such as MRR or NDCG. We then experimentally validate NRG with two different test collection, the MS MARCO development set and the TREC Deep Learning 2019 collection, which use MRR and NDCG as their primary metrics, respectively. Our experiments reveal an interesting insight into the differences between traditional ranking methods and modern neural ranking methods. While modern neural ranking methods exhibit superior

Submitted to SIGIR-AP 2024, December 9-12, 2024, Tokyo, Japan

ACM ISBN XXX-X-XXXX-XXXX-X/XX/XX. . . $XX.XX
https://doi.org/XX.XXXX/XXXXXXX.XXXXXXX

[1] Appendix A analyses theoretical connections between NRG and C/W/L.

performance in absolute terms, traditional methods can outperform neural methods in relative terms, finding relevant material not found by neural methods. One practical conclusions is that, when building future reusable test collections, traditional methods should continue to form the baselines, since they may find unique relevant items not found by modern neural methods[2].

2 NORMALIZED RESIDUAL GAIN

We can express many of the most widely used offline evaluation metrics as a user browsing model with the form [7, 8, 21, 25]:

$$S(R) \ = \ \frac{1}{\mathcal{N}} \sum_{i=1}^{K} G(d_i) \cdot \text{seen}(i), \tag{1}$$

where $R =< d_1, d_2, ... >$ is a ranked list with depth $\geq K$ produced by a ranker for a query q; $G(d)$ is the gain associated with the searcher observing item d; $\text{seen}(i)$ is a survival probability that the searcher observes the item at rank i; and $\mathcal{N}$ is a normalization factor, typically intended to map the metric into a value in the range $[0, 1]$. We assume $\text{seen}(i)$ is monotonically decreasing with increasing i, and for later convenience we define $\text{seen}(\infty) = 0$. To evaluate a ranker, $S(R)$ is usually averaged over a large set of queries $q \in Q$.

NDCG@K, RBP, precision@K and MRR all fit this model, each with different instantiations for $\mathcal{N}$, $G(d)$, and $\text{seen}(i)$ [6, 26]. For example, for NDCG@K gain $G(i)$ is typically based on human relevance labels and $\text{seen}(i) \ = \ 1/(\log(i) + 1)$. The normalization factor is based on the maximum possible value for the raw, unnormalized score $S_{\text{raw}}(R) \ = \ \sum_{i=1}^{K} G(d_i) \cdot \text{seen}(i)$. Given a collection of items with known gain values, we sort them by decreasing gain, breaking ties arbitrarily, to create an *ideal ranking* R_{ideal} of depth K. The normalization factor is then $\mathcal{N} \ = \ S_{\text{raw}}(R_{\text{ideal}})$, so that:

$$S(R) \ = \ \frac{S_{\text{raw}}(R)}{S_{\text{raw}}(R_{\text{ideal}})}. \tag{2}$$

Now, suppose the searcher has already observed a set of rankings for q, each produced by a different ranker, $\mathcal{R} = \{R_1, ..., R_M\}$, which we call the *prior rankings* or *prior set*. While an item can appear in multiple prior rankings, or even all of them, we assume that no item appears multiple times in a given ranking R_i. Then, for an item d appearing in the top K of ranking R, we can define $\text{pos}(d, R)$ as the position where it appears. If the item does not appear in the top K, we define $\text{pos}(d, R) = \infty$.

We assume that once a searcher has observed an item, they will received no further gain from observing it again, so that we can compute a *residual gain* for an item in terms of the survival probability of the item in each prior ranking:

$$G(d|\mathcal{R}) \ = \ G(d) \cdot \prod_{j=1}^{M} \left(1 - \text{seen}(\text{pos}(d, R_j)\right). \tag{3}$$

To define *normalized residual gain* (NRG) of a ranker R with regard to a set of prior rankings $\mathcal{R}$ we substitute Equation 3 for the gain value in Equation 1, giving:

$$S(R|\mathcal{R}) \ = \ \frac{1}{\mathcal{N}} \sum_{i=1}^{K} G(d_i|\mathcal{R}) \cdot \text{seen}(i), \tag{4}$$

If an ideal ranking is required for normalization, this ideal ranking should be derived by sorting items according to residual gain.

If $M = 0$, we have $G(d|\mathcal{R}) \ = \ G(d)$, and so Equation 4 reduces to Equation 1. Equation 1 is known to provide a reasonable — if overly simplistic — model of searcher behavior. However, for values of M greater than 1 or 2, Equation 4 probably does not. If $M = 1$ we might view Equation 4 as measuring the expected benefit of, for example, switching to a different search engine or clicking on a reformulation. If $M \gg 2$, as it does in later experiments, we not claim that Equation 4 provides a meaningful model of searcher behavior. No searcher could be expected to mechanically scan dozens of rankings, looking for unobserved items. We claim only that Equation 4 is derived from Equation 1 in a theoretically justifiable way in terms of survival probabilities. As we see in the next two sections, Equation 4 can also provide novel insights into experimental comparisons[3].

3 IMPROVEMENTS OVER TIME

In this section we apply NRG to measure performance difference between retrieval methods released over a period of a time, between June 2020 and September 2021. At the time of release, each represented a performance improvement over the prior methods, as measured by MRR@10 on the 8.8 million passages of the MS MARCO test collection using its "small dev" query set [4]. This query set comprises 6,980 queries with associated binary relevance judgments over the passages. When compared to most other test collections, this test collection has a larger number of queries, but fewer judgments per query. Most queries (94%) have only a single judged relevant passage; no query has more than 4 judged relevant passages. Given the sparsity of judgments and the use of binary relevance, MRR@10 is the standard effectiveness measure for this collection.

This period of time represents a major transition in retrieval methods, away from methods based on term and document statistics and towards neural methods. Many different neural methods were proposed and developed over this period, including sparse neural methods, dense methods and various hybrid methods. For our experiments, we employ a diverse range of retrieval models, categorized into four distinct groups: 1) BM25 [28], an unsupervised sparse retrieval model based on term and document statistics; 2) Four dense retrieval models, namely RepBERT [38], ColBERT [19], ANCE [32], and S-BERT [27]; 3) A hybrid model, ColBERT-H [19]; and 4) two learned sparse retrieval models, UniCOIL [20] and SPLADE [17].

Figure 1(a) tracks changes in the neural methods over that period. Methods are in chronological by release date, with release dates shown. For each method, the prior set consists of those methods released before it. For example, for ANCE, both BM25 and RepBERT serve as the prior set. For RepBERT, BM25 alone serves as the prior set. In the case of the latest retrieval method, SPLADE, all other ranking methods are included in the prior set. With each new release, MRR increases while NRG drops, flattening over time. The initially high NRG value associated with RepBERT reflects its ability to return relevant passages that are missed by BM25.

Figure 1(b) provides an NRG-based comparison between the same methods where the prior set does not increase over time. Instead, the

[2] Appendix B provides an analysis of the role of NRG in test collection construction.

[3] The implementation of the NRG metric used for these experiments is available at https://anonymous.4open.science/r/Normalized-Residual-Gain-0B4A/

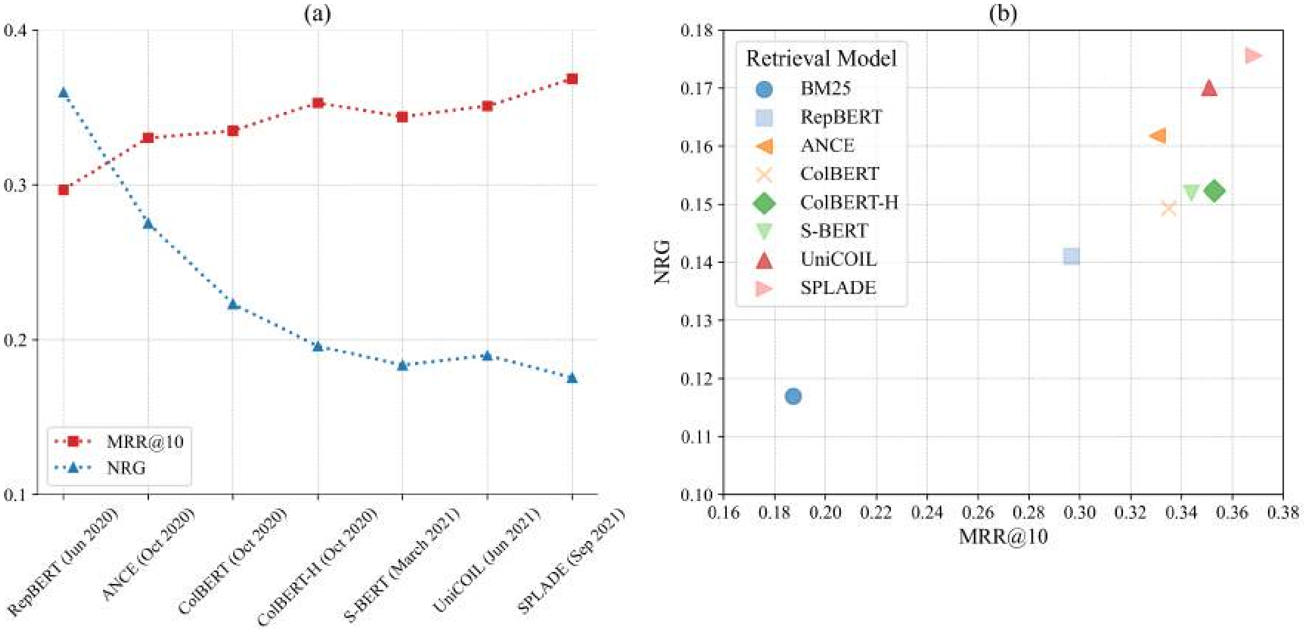

Figure 1: NRG vs. MRR@10 comparison of retrieval methods over the MS MARCO small dev test collection. (a) Chronological comparison of retrieval methods demonstrating that while newly released methods enhance retrieval effectiveness, NRG decreases over time. (b) Results of an inter-method comparisons between all retrieval methods, where the prior set for computing NRG consists of all other methods.

prior set for each method consists of all other methods. The MRR@10 value for on this plot is the same as in Figure 1(a). The other points correspond to the NRG value obtained if the method had been released last. While NRG generally correlates with MRR@10, we see some interesting difference that might warrant further investigation. For example, if we are creating a hybrid ranker that combines several methods, we might consider including ANCE over ColBERT, since ANCE provides a higher NRG.

4 COMPARISONS BETWEEN SYSTEMS

In this section we apply NRG to compare systems submitted to the passage-retrieval task of the TREC 2019 Deep Learning track [14]. The track employed the same MS MARCO passage corpus with 43 new queries. While the number of queries is much smaller than the "small dev" set, these queries are judged on a 4-point graded relevance scale: 0 ("Irrelevant"), 1 ("Related"), 2 ("Highly relevant"), and 3 ("Perfectly relevant"). As is usual for TREC experiments, experimental runs submitted by track participants were pooled and judged to a fixed depth, creating 9,260 judgments in total, with between 132 and 582 judgments per query.

Figure 2 plots NRG vs. NDCG@10 for all runs submitted to the passage-retrieval task of the TREC 2019 Deep Learning Track. For each run, the prior set use for computing its NRG consists of the best run by NDCG@10 submitted from each other group. We include only the single best run from each group in the prior set since some groups submitted more runs than others, and we want each group to be represented equally. We also exclude from each run's prior set other runs from the same group since they may be only trivially different.

With the exception of the "BASELINE" runs, the runs are labeled by the participating group that submitted it. While we treat the BASELINE as a single group for the purpose of computing NRG, these runs were solicited from various groups who were encouraged "to run strong traditional IR baselines, and submit them as additional runs," creating a valuable opportunity to directly compare neural and non-neural methods. These runs represent serious "best effort" attempts at the retrieval task without neural methods. The best of these BASELINE runs, as measured with NDCG@10 (`bm25tuned_prf_p`) incorporates grid search for parameters tuning and query expansion using pseudo-relevance feedback [37].

Most of the non-BASELINE runs, including all of the top 20 runs under NDCG@10, employ neural methods. They dominate the runs based on traditional methods. Under NRG however, we see a clear split between the traditional and neural methods. Within each group, NRG generally increases with increasing NDCG. However the high values of NRG seen in the BASELINE methods suggest that neural runs may be ranking passages differently than traditional methods. Many of the best BASELINE runs under NRG incorporate pseudo-relevance feedback [34, 37], which may be a source of unique relevant passages. The best non-neural run (`srchvrs_ps_run3`) has an NDCG@10 slightly higher than the best BASELINE run but an NRG lower than all of them. It employs a combination of traditional methods [5], but notably does not include pseudo-relevance feedback[4].

The plot includes other difference that may be worth investigating. The best method by NDCG@10 (`idst_bert_p1`) employs neural methods to expand each passage, creates a traditional inverted index

[4] Personal communication with L. Boytsov, as there is no TREC proceedings paper.

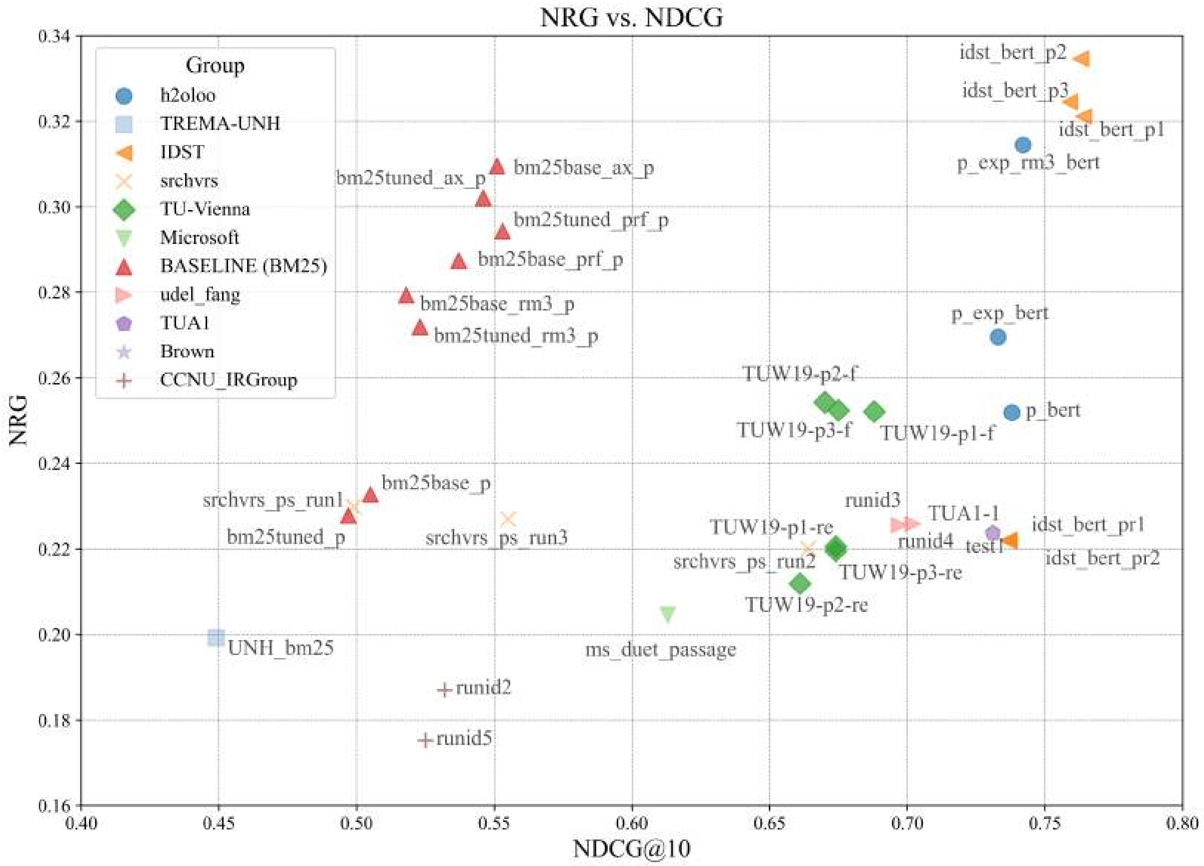

Figure 2: NDCG@10 vs. NRG for all experimental runs submitted to the passage-retrieval task of the TREC 2019 Deep Learning Track. For each run, the prior set for computing NRG consists of the best run submitted from each other group. Higher NDCG generally correlates with higher NRG, except for the non-neural BM25-based runs, especially those from the "BASELINE" grouping. Despite having lower NDCG@10 values, these runs exhibit higher values of NRG.

from the expanded passages, and applies BM25 [33]. Another run from the same group (`idst_bert_pr1`) includes a final re-ranking step, which slightly harms NDCG@10, but drops NRG below all BASELINE runs. Notably, this re-ranker is trained on MS MARCO, as are many of the other top methods. Due to this training, the re-ranking may favor the same passages as other methods trained on MS MARCO.

5 CONCLUDING DISCUSSION

We introduce and experimentally validate the Normalized Residual Gain (NRG) metric. NRG measures the effectiveness of a ranked list relative to the performance of a prior set of ranked lists. For example, A ranker returning many relevant items not appearing the prior set will receive a higher NRG score than a ranker returning equally relevant items that repeatedly appear in the prior set. Informally, NRG provides an indication of the difference between a ranking and the rankings in the prior set.

NRG was directly inspired by the work of Türkmen et al. [30] and Arabzadeh et al. [1], but builds on a theoretically sounder foundation by extending the browsing models of establishes metrics. Neither Türkmen et al. nor Arabzdeh et al. account for the positions at which items appears in prior rankings. Türkmen et al. compute rarity as a linear function of the number of systems that return a item, regardless of the rank at which it is returned. Arabzdeh et al. directly aggregate effectiveness scores, and do not consider individual items at all. Figure 1(a) can be directly compared with Figure 3 in Arabzdeh et al, and Figure 1(b) can be directly compared with Figure 4(c) and 4(d) in Arabzdeh et al. Neither Türkmen et al. nor Arabzdeh et al. show clear differences between distinct ranking approaches, e.g., neural vs. traditional[5].

The results shown in Figure 2 and discussed in Section 4 illustrate insights provided by NRG, including an ability to identify rankers that have unique characteristics. Novel retrieval approaches may not immediately perform at the state-of-the-art, even if they show initial promise, and may be discarded. If future evaluation efforts report NRG (similar to Figure 2) it may allow these approaches to demonstrate their potential, even if they do not place in the top few runs by NDCG, or other traditional measures.

[5] Appendix C provides a further experimental comparison between NRG and the metrics of Türkmen et al. [30] and Arabzadeh et al. [1].

REFERENCES

[1] Negar Arabzadeh, Amin Bigdeli, Radin Hamidi Rad, and Ebrahim Bagheri. 2023. Quantifying ranker coverage of different query subspaces. In *46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2298–2302.

[2] Negar Arabzadeh, Oleksandra Kmet, Ben Carterette, Charles LA Clarke, Claudia Hauff, and Praveen Chandar. 2023. A is for Adele: An offline evaluation metric for instant search. In *ACM SIGIR International Conference on Theory of Information Retrieval*. 3–12.

[3] Leif Azzopardi, Joel Mackenzie, and Alistair Moffat. 2021. ERR is not C/W/L: Exploring the relationship between expected reciprocal rank and other metrics. In *ACM SIGIR International Conference on Theory of Information Retrieval*. 231–237.

[4] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. (2018). arXiv:cs.CL/1611.09268 https://arxiv.org/abs/1611.09268

[5] Leonid Boytsov. 2020. Traditional IR rivals neural models on the MS~MARCO Document Ranking Leaderboard. *CoRR* abs/2012.08020 (2020).

[6] Ben Carterette. 2011. System effectiveness, user models, and user utility: A conceptual framework for investigation. In *34th international ACM SIGIR conference on Research and development in information retrieval*. 903–912.

[7] Ben Carterette, Evangelos Kanoulas, and Emine Yilmaz. 2011. Simulating simple user behavior for system effectiveness evaluation. In *20th ACM International Conference on Information and Knowledge Management*. 611–620.

[8] Ben Carterette, Evangelos Kanoulas, and Emine Yilmaz. 2012. Incorporating variability in user behavior into systems based evaluation. In *21st ACM International Conference on Information and Knowledge Management*. 135–144.

[9] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *18th ACM Conference on Information and Knowledge Management*. 621–630.

[10] Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 659–666.

[11] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 758–759.

[12] Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. 1998. Efficient construction of large test collections. In *21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 282—-289.

[13] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the TREC 2020 deep learning track. In *Text REtrieval Conference*. Gaithersburg, Maryland.

[14] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2019. Overview of the TREC 2019 deep learning track. In *Text REtrieval Conference*. Gaithersburg, Maryland.

[15] Donna Harman Ellen M. Voorhees. 1998. Overview of the Seventh Text REtrieval Conference. In *7th Text REtrieval Conference*. Gaithersburg, Maryland.

[16] Donna Harman Ellen M. Voorhees. 1999. Overview of the Eighth Text REtrieval Conference. In *8th Text REtrieval Conference*. Gaithersburg, Maryland.

[17] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse lexical and expansion model for first stage ranking. In *44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2288–2292.

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

[19] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *43rd International ACM SIGIR conference on research and development in Information Retrieval*. 39–48.

[20] Jimmy Lin and Xueguang Ma. 2021. A few brief notes on DeepImpact, COIL, and a conceptual framework for information retrieval techniques. *arXiv preprint arXiv:2106.14807* (2021).

[21] Aldo Lipani, Ben Carterette, and Emine Yilmaz. 2019. From a user model for query sessions to session rank biased precision (sRBP). In *ACM SIGIR International Conference on Theory of Information Retrieval*. 109–116.

[22] David E. Losada, Javier Parapar, and Álvaro Barreiro. 2016. Feeling lucky? Multi-armed bandits for ordering judgements in pooling-based evaluation. In *31st Annual ACM Symposium on Applied Computing*. 1027–1034.

[23] Alistair Moffat, Peter Bailey, Falk Scholer, and Paul Thomas. 2017. Incorporating user expectations and behavior into the measurement of search effectiveness. *ACM Transactions on Information Systems* 35, 3 (2017), 1–38.

[24] Alistair Moffat, Joel Mackenzie, Paul Thomas, and Leif Azzopardi. 2022. A flexible framework for offline effectiveness metrics. In *45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 578–587.

[25] Alistair Moffat, Paul Thomas, and Falk Scholer. 2013. Users versus models: What observation tells us about effectiveness metrics. In *22nd ACM International Conference on Information & Knowledge Management*. 659–668.

[26] Alistair Moffat and Justin Zobel. 2008. Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems* 27, 1 (2008), 1–27.

[27] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *arXiv preprint arXiv:1908.10084* (2019).

[28] Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 232–241.

[29] Tetsuya Sakai and Noriko Kando. 2008. On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Information Retrieval* (October 2008), 447–470.

[30] Mehmet Deniz Türkmen, Matthew Lease, and Mucahid Kutlu. 2023. New metrics to encourage innovation and diversity in information retrieval approaches. In *European Conference on Information Retrieval*. 239–254.

[31] Ellen M Voorhees, Ian Soboroff, and Jimmy Lin. 2022. Can old TREC collections reliably evaluate modern neural retrieval models? *arXiv preprint arXiv:2201.11086* (2022).

[32] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[33] Ming Yana, Chenliang Li, Chen Wu, Bin Bi, Wei Wang, Jiangnan Xia, and Luo Si. 2019. IDST at TREC 2019 Deep Learning Track: Deep cascade ranking with generation-based document expansion and pre-trained language modeling. In *28th Text REtrieval Conference*. Gaithersburg, Maryland.

[34] Peilin Yang and Jimmy Lin. 2019. Reproducing and generalizing semantic term matching in axiomatic information retrieval. In *European Conference on Information Retrieval*. 369–381.

[35] Emine Yilmaz, Milad Shokouhi, Nick Craswell, and Stephen Robertson. 2009. Incorporating User Behavior Information in IR Evaluation. In *SIGIR 2009 Workshop on Understanding the User: Logging and Interpreting User Interactions in Information Retrieval*.

[36] Emine Yilmaz, Milad Shokouhi, Nick Craswell, and Stephen Robertson. 2010. Expected Browsing Utility for Web Search Evaluation. In *19th ACM International Conference on Information and Knowledge Management*.

[37] Zhaohao Zeng and Tetsuya Sakai. 2019. BM25 Pseudo Relevance Feedback Using Anserini at Waseda University. In *Proceedings of the Open-Source IR Replicability Challenge (OSIRRC) Workshop*.

[38] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. RepBERT: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498* (2020).

A COMPARISON WITH C/W/L AND C/W/L/A

Reviews for this paper asked about theoretical connections with the C/W/L framework [23–25], suggesting that C/W/L might provide a more general starting point for the theoretical development of NRG than Equation 1. While we might reasonably have started with C/W/L, it does not provide a more general starting point. If we substitute Equation 3 of Moffat et al. [24] into Equation 4 of that paper, the result corresponds to Equation 1 of this paper, with $G(d_i) = r_i$, $\text{seen}(i) = V(i)$, $\mathcal{N} = V^+$, and $K \to \infty$. With appropriate instantiations of $G(d_i)$, $\text{seen}(i)$, and $\mathcal{N}$, Equation 1 encompasses the same range of metrics as C/W/L, including precision@K, rank biased precision [26], and DCG, while providing more explicit support for NDCG via the normalization factor $\mathcal{N}$.

Reviews also asked about theoretical connections with the extended C/W/L/A framework described by Moffat et al. [24], which adds an additional aggregation component to the C/W/L framework (the "A"). This additional component accommodates evaluation metrics such as *expected reciprocal rank* (ERR) [3, 9], which discounts the relevance of the item at rank i according to the relevance of items appearing at ranks less than i. ERR models a searcher scanning down a ranked list by assuming that the probability they will stop scanning increases with the amount of relevant material they have

												Prior set			
	Ranking										NDCG@10	R1	R2	R3	Other two R1-3
R1	**A**	B	C	D	**E**	**F**	G	H	I	**J**	0.7933	-	0.7361	0.8277	0.8417
R2	**E**	D	C	B	**A**	**F**	G	H	I	**J**	0.7933	0.7361	-	0.7988	0.8316
R3	**J**	I	H	G	**F**	**E**	D	C	B	**A**	0.7933	0.8277	0.7988	-	0.8681

Table 1: Impact of rank order on NRG. All three rankings include the same items in different orders. Relevant items are bolded. While NDCG@10 is the same for all rankings, NRG depends on the prior ranking or rankings. See Appendix B for further discussion.

seen, an assumption confirmed by experience in commercial web search environments [9]. The original C/W/L framework assumes that the relevance of a item is independent of the items that appear above it in a ranked list. With a few exceptions [9, 10] this independence assumption is shared by other evaluation metrics, including NRG. The C/W/L/A extension provides a mechanism to address this independence assumption, as well as providing other benefits.

NRG addresses the same independence assumption, but in a manner that is orthogonal to C/W/L/A and ERR. NRG adjusts the relevance of an item (i.e., its gain value) according to a context defined by a set of prior rankings, rather than by the relevance of the items that appear above it in the current ranking. While it should be possible to combine these ideas — making adjustments for both prior rankings and relevant material in higher ranks — we leave the exploration of this idea for future work.

B UNIQUE CONTRIBUTIONS

Reviews for this paper suggested an analysis of NRG as a tool for the development of more robust test collections. Past work on test collection development — especially collections developed through TREC – have encouraged diverse retrieval approaches in order to increase the set of judged-relevant documents. For example, the overview papers for TREC 7 [15] and TREC-8 [16] report "unique contributions to the relevant set", noting that manual runs tend to make more unique contributions than automatic runs.

TREC test collections are typically constructed using a pooling process that starts with the experimental runs submitted by participating groups [13–16]. To create a pool of documents for judging, TREC organizers take the top-K documents from one or more runs per group, based on priorities assigned by the group. These documents are de-duplicated, randomized, and presented to assessors for judging. This process is known as "pooling to depth K". If a document appears in the final set of judged relevant documents only because it appears in the top-K of a single assessed run, that run is given credit for contributing a unique document to the pool.

NRG generalizes this "unique contributions to the relevant set" metric. In Equation 1 let $G(d_i) = 1$, if d_i is relevant, 0 otherwise; $\mathrm{seen}(i) = 1$, if $i \leq K$, 0 otherwise; $\mathcal{N} = 1$; and K equal to the pooling depth. If we take the other runs contributing to the pool as the prior set, the resulting instantiation of NRG is exactly the number unique contributions a ranking makes to the pool, averaged over all queries.

Figure 3 plots this "unique contributions" instantiation of NRG vs. NDCG@10. Since none of the reports or data for the TREC 2019 Deep Learning Track indicate which runs contributed to the pool — at least as far as we can see — we take as the prior set as the top run by NDCG@10 from each other group. Consistent with Figure 2, traditional retrieval methods exhibit high NRG relative to NDCG@10. Under this variant of NRG, many neural methods make relatively few or no unique contributions, while some traditional methods outperform all neural methods.

Variants of NRG based on NDCG, MRR and other metrics generalize the notion of "unique contributions" to account for the depth that items appear. However, the value of NRG does not depend solely on the presence of unique items in a ranking. Table 1 provides an example. Each of the three rankings (R1, R2, and R3) contain the same items. At different pooling depths each would contribute a different number of unique items to a hypothetical pool, but at depth 10 none make a unique contribution. Each has a relevant item at rank 1, 5, 6, and 10, which we assume to have equal relevance at the highest grade, a grade of 4. If we know nothing else about these items, there is no basis for an evaluation metric to assign them different scores. NRG values, on the other hand, reflect the different orderings of the items. R1 and R2, place A at the top and J at the bottom, while R3 places J at the top and A at the bottom.

Reviews for this paper suggested that the only application of NRG might be to inform test collection development and improve the robustness of pools. While improving robustness of pools represents one application of NRG, it is not the only application. Informally, NRG measures the extent to which ranking methods behave "like" some other methods or "differently" than those other methods. While Figure 2 and 3 suggest that traditional ranking methods — especially those employing pseudo-relevance feedback — should be encouraged for test collection development, the figures also suggest that a traditional method might be valuable in an ensemble ranker that includes several neural methods. In general, NRG may have application in any circumstance where we wish to measure diversity in a set of rankings.

Reviews expressed concerns about the impact of missing judgments on NRG. Like most standard evaluation metrics, NRG is computed *post hoc*, after items have been judged. Like most standard evaluation metrics, NRG may be impacted by unjudged items, which are often assumed to be non-relevant. Sakai and Kando [29] discuss approaches to address this problem that could be adapted to NRG. We leave this effort to future work.

Developing robust test collections may also require us to do more than just encourage a diversity of methods. Cormack et al. [12] and Losada et al. [22] both describe dynamic pooling methods that improve upon fixed-depth pooling, with the goal of increasing the ratio of relevant items found per judgment. These methods tend to favor runs that have many relevant items in high ranks, i.e., runs that will ultimately have high NDCG. It is possible that NRG could be adapted to created a dynamic pooling method that favors diversity. We leave this effort to future work.

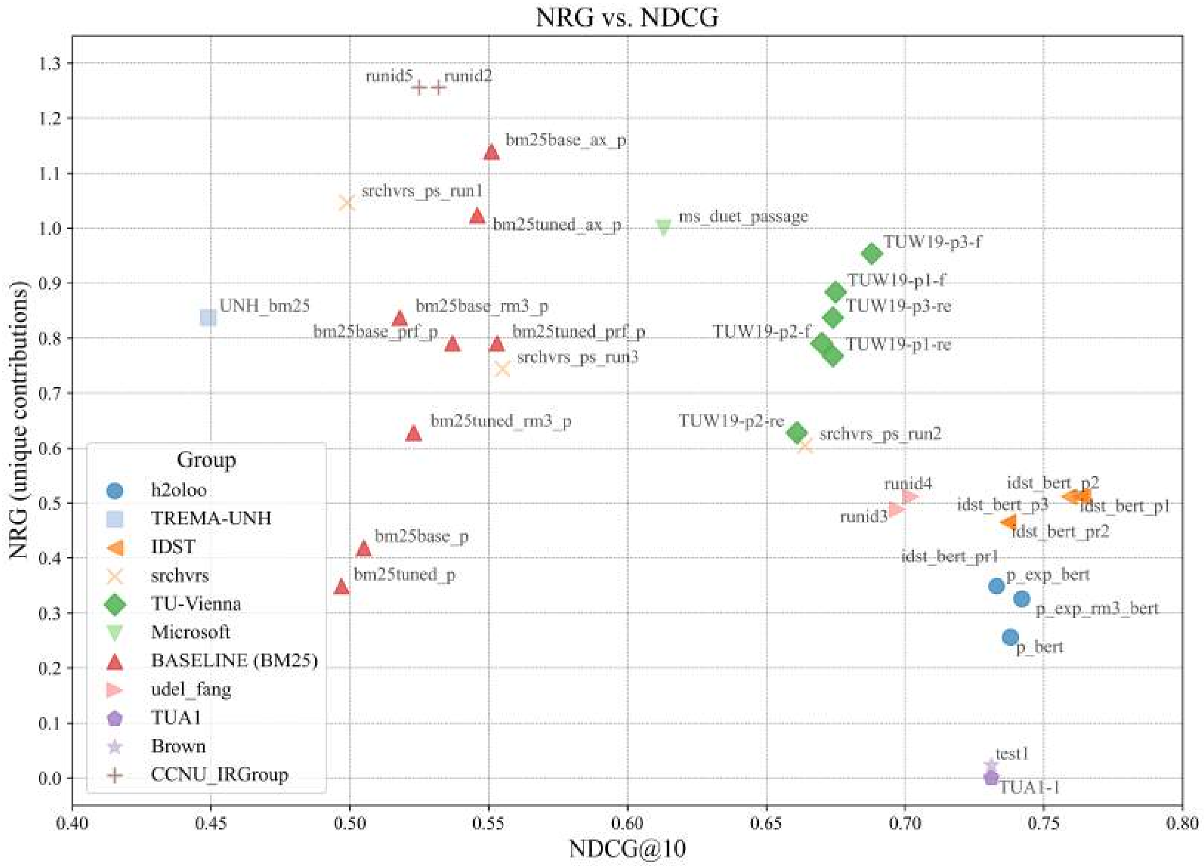

Figure 3: NDCG@10 vs. NRG@10 for all runs submitted to the passage-retrieval task of the TREC 2019 Deep Learning Track. In this plot, NRG is derived from precision@10, where 10 is the pooling depth for the track. For each run, the prior set for computing NRG@10 consists of the best run submitted from each other group. Appendix B explains the relationship between this metric and unique contributions to the relevant set.

C FURTHER COMPARISONS

As a central contribution of this paper, we place the ideas of Türkmen et al. [30] and Arabzadeh et al. [1] on a more solid theoretical foundation. We are not claiming that the work in those papers is incorrect, or that different conclusions would always be reached if we apply NRG instead of the metrics in those paper. We have already noted that Figure 1 can be directly compared with plots in Arabzadeh et al. [1]. Türkmen et al. [30] have no notion of a prior set — they compute rareness in terms of all available rankings — so we cannot compute the equivalent figure for their metric without modification, which was the starting point for the work in this paper.

Neither Türkmen et al. [30] nor Arabzadeh et al. [1] provide a software release. In addition to an implementation of our NRG metric, our software release includes an implementation of rareness-based precision-at-k [30] and TaSC [1][6]. Using these implementations we have repeated the experiment of Section 4.

The results are plotted in Figure 4 and Figure 5. These plots may be compared to Figure 2 and Figure 3. Both rareness-based precision@10 and TaSC are correlated with NDCG@10. Neither metric suggests any distinction between traditional and neural methods. Since rareness-based precision@10 [30] does not have an notion of a prior set, we follow the definition of that paper and compute the metric over all runs. While we considered modifying the metric to incorporate the notion of a prior set, for the purposes of this comparison we adhere as closely as possible to the definition in the paper.

Other comparisons might prove interesting. In particular, it might be interesting to apply NRG to older TREC experiments, where manual runs were more prevalent [15, 16]. It might also be interesting to explore various re-ranking and other methods in terms of their impact on NRG. For example, we might expect ensemble methods such as reciprocal rank fusion [11] to lower NRG, since it gives higher scores to items that appear in more rankings. If NRG proves to be a useful tool, these comparisons can be considered for future work.

[6] https://anonymous.4open.science/r/Normalized-Residual-Gain-0B4A/

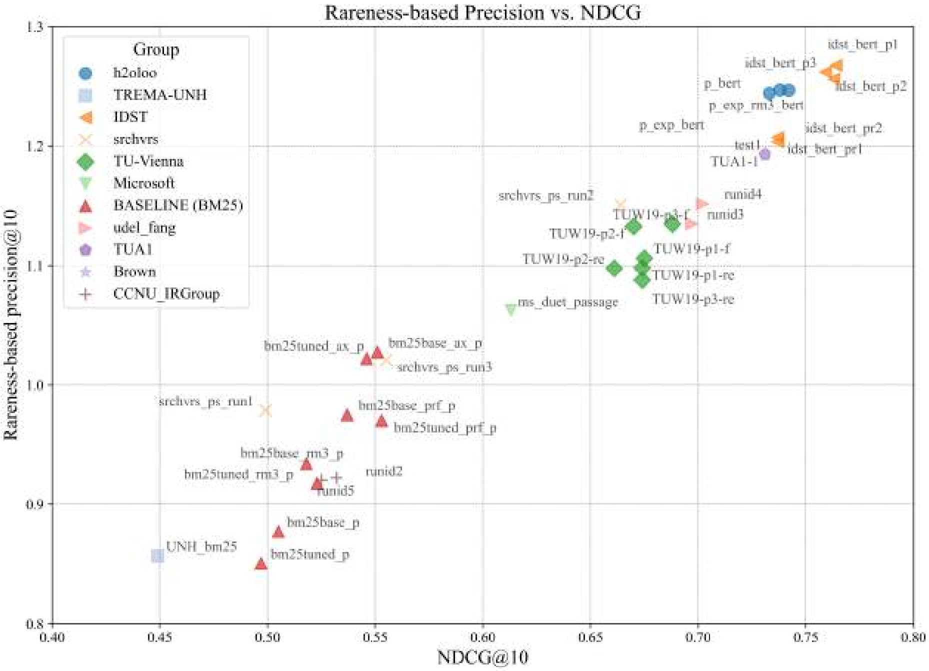

Figure 4: NDCG@10 vs. rareness-based precision@10 for all runs submitted to the passage-retrieval task of the TREC 2019 Deep Learning Track. For each run, the rarity of documents within top-10 ranked list of documents is calculated based on all the runs including the target run.

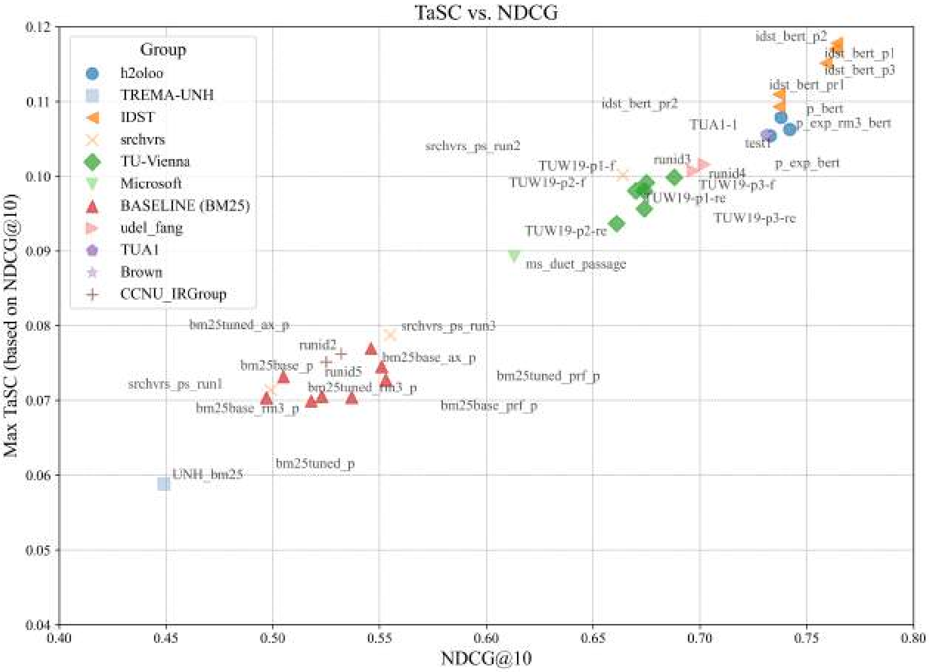

Figure 5: NDCG@10 vs. TaSC for all runs submitted to the passage-retrieval task of the TREC 2019 Deep Learning Track. For each run, the prior set for computing MAX TaSC consists of the best run submitted from each other group.