A is for Adele: An Offline Evaluation Metric for Instant Search

Negar Arabzadeh
University of Waterloo
narabzad@uwaterloo.ca

Oleksandra Kmet
University of Waterloo
okmet@uwaterloo.ca

Ben Carterette
Spotify
benjaminc@spotify.com

Charles L. A. Clarke
University of Waterloo
charles.clarke@uwaterloo.ca

Claudia Hauff
Spotify
claudiah@spotify.com

Praveen Chandar
Spotify
praveenr@spotify.com

ABSTRACT

Instant search has emerged as the dominant search paradigm in entity-focused search applications, including search on Apple Music, LinkedIn, and Spotify. Unlike the traditional search paradigm, in which users fully issue their query and then the system performs a retrieval round, instant search delivers a new result page with every keystroke. Despite the increasing prevalence of instant search, evaluation methodologies for instant search have not been fully developed and validated. As a result, we have no established evaluation metrics to measure improvements to instant search, and instant search systems still share offline evaluation metrics with traditional search systems. In this work, we first highlight critical differences between traditional search and instant search from an evaluation perspective. We then consider the difficulties of employing offline evaluation metrics designed for the traditional search paradigm to assess the effectiveness of instant search. Finally, we propose a new offline evaluation metric based on the unique characteristics of instant search. To demonstrate the utility of our metric, we conduct experiments across two very different platforms employing instant search: A commercial audio streaming platform and Wikipedia.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Users and interactive retrieval**; **Collaborative search**; *Specialized information retrieval*; **Retrieval effectiveness**.

KEYWORDS

Evaluation, Retrieval Effectiveness, Instant Search

ACM Reference Format:
Negar Arabzadeh, Oleksandra Kmet, Ben Carterette, Charles L. A. Clarke, Claudia Hauff, and Praveen Chandar. 2023. A is for Adele: An Offline Evaluation Metric for Instant Search. In *Proceedings of the 2023 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '23), July 23, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3578337.3605115

ICTIR '23, July 23, 2023, Taipei, Taiwan

ACM ISBN 979-8-4007-0073-6/23/07…$15.00
https://doi.org/10.1145/3578337.3605115

1 INTRODUCTION

In *instant search*, a searcher is provided with a complete result list immediately after each character they enter. As soon as they enter the first character ("A", for example), the searcher receives search results that contain "Amazon", "Amazon Prime", "Airbnb", and so on. As soon as they enter a second character ("d"), the result list switches to "Adobe", "Adidas shoes", etc. Finally, upon entering a third character ("e") the searcher receives search results that include the video for Adele's *Easy on Me* as the sixth result, which was the target of their search. The searcher clicks to watch the video, and the search ends. A growing number of commercial search services now feature instant search in essentially this form, including Apple Music, Kayak, Netflix [21], LinkedIn [36], and Spotify [6]. Instant search services tend to focus on entity search (e.g. apps, games, music, people, podcasts, and videos) often with a single target for the search. Traditional search services often provide *query suggestions* or *autocompletions* as the searcher types, which appear superficially similar to instant search. However, these mechanisms are intended to guide the user in formulating a complete query, rather than directly providing results [4, 11, 22, 27, 34].

Traditional IR metrics, such as MRR and nDCG [17] assume a *complete query* as indicated by a user's click on a "Search" button or an equivalent action, as well as a single Search Engine Results Page (SERP) which the user inspects top-down until they locate the target(s) of their search or give up browsing and reformulate their query [31, 38]. These metrics operate on a single ranked list, summing over ranks. Applying these metrics independently to the individual query/SERP pairs of a multi-character search sequence in instant search means that most result lists will have a zero score.

Our work focuses on instant search and its evaluation in an offline setting. After examining the characteristics and requirements of instant search from an evaluation perspective, we demonstrate that traditional metrics such as nDCG [17] do not satisfy all these requirements. *Based on these insights, we propose and experimentally validate a new offline evaluation metric for instant search that fulfills the requirements.* This new metric, which we called "2d-Gain", considers both the rank of the target and the length of the character sequence required to reach it. We demonstrate the utility of our metric in the context of two distinct platforms employing instant search: (i) A commercial audio streaming platform, and, (ii) Wikipedia. While our evaluation metric could be applied more generally, we focus our experiments on searches that have a single entity as a target—a common setting of instant search systems. Post-search engagement with entities (e.g., downloading an app,

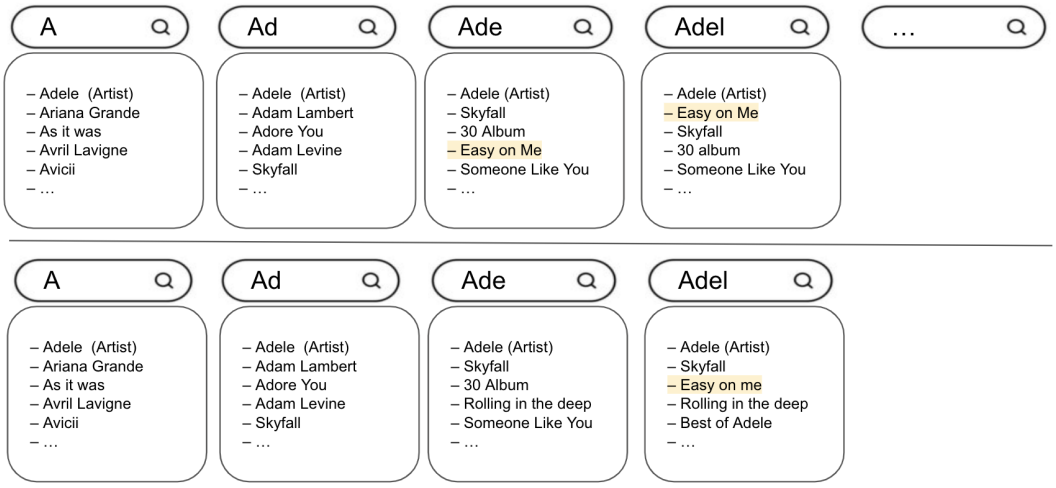

Figure 1: Two examples of search sequences where the user is looking for the song "Easy on me" by "Adele". In the top row, the search sequence has more than one occurrence of the successful entity, both on the third and fourth SERP. In the bottom row, the user stops once they see the relevant result i.e., the successful item only occurs once in the sequence.

listening to a song, or watching a video) provide a high level of confidence that the search was successful and the target was found.

While entity popularity represents a major ranking feature for instant search, practically speaking, an effective instant search surface must also support a high level of personalization. On an audio streaming platform, a short prefix may be insufficient to disambiguate entities without considering the listening habits, language, location, and other features associated with the searcher. For some searchers, Adele's *Easy on Me* should be the top result for the query "A", while for others AC/DC's *Thunderstruck* will be best. On a general web search surface, "A" is for Apple or Amazon. Through our experiments on data from a commercial audio streaming platform, we demonstrate that our metric can provide insights into the trade-off between popularity and personalization. Recognizing that Wikipedia's search is essentially instant search, we repeat a version of our experiments on Wikipedia, providing an additional demonstration of our approach, and suggesting that Wikipedia may benefit from exploring further popularity features for its ranker.

After a brief review of related work, the remainder of the paper is organized as follows: §3 discusses the unique characteristics of instant search, provides details of terminology, explains assumptions regarding searcher behavior; §4 provides a description of our 2d-Gain framework. We present two case studies of 2d-Gain. The first case study (§5) reports an experimental comparison between a personalized full-featured ranker and a simple popularity-based ranker on a commercial audio streaming search surface. The second case study (§6) reports a similar comparison on a dataset derived from Wikipedia. Finally, we discuss limitations of 2d-Gain, as well as future work (§7).

Our main contributions include 1) We explore the differences between instant search and traditional search, exposing the limitations of traditional evaluation metrics in the context of instant search. 2) We propose a 2d-Gain metric, which considers both the rank of the target entity and the length character sequence entered, arguing for its suitability as a metric for the offline evaluation of instant search. 3) We experimentally explore three different instantiations of 2d-Gain with different discount factors: (i) estimated from production data; (ii) based on exponential decay, inspired by RBP [31]; (iii) based on the nDCG discount function. 4) We compare a personalized full-featured ranker and a simple popularity-based ranker on a commercial audio streaming platform and separately on a Wikipedia dataset, showing that 2d-Gain can provide beneficial insights into trade-offs between popularity and personalization.

2 BACKGROUND AND RELATED WORK

While online evaluation has shown its merits, especially when considering user engagement, offline evaluation of search systems has always had more advocates because it is cheaper and more generalizable [14, 32]. Therefore, we build our work on a long history of offline evaluation metrics with a focus on metrics targeted at web search and image search [37, 40].

2.1 Evaluation in context

An evaluation metric should consider the context of the retrieval and the interface through which the searcher interacts with the results. For example, Xie et al. [37] argue that the modeling of searcher behavior should differ between web search and image search due to the differences between their interfaces. The use of a two-dimensional grid in image search requires different evaluation metrics than those designed for a single ranked list. Other factors that have a significant impact on how searcher behavior should be modeled — and how the search engine should be evaluated —include the number of items that will be shown to the user on each SERP and the options for pagination vs. scrolling. Overall, evaluation methodologies must appropriately reflect a search engine's interface and interaction characteristics. Each search application, interface, and context may require a different evaluation metric. There is no single standard metrics that can fit all circumstances. As shown in prior work, different evaluation metrics are required in different settings, including web search [1, 7–10, 31], image search [16, 37, 40] and conversational search [13, 24–26]. Like these applications, instant search must also be evaluated on the basis of its interface and interaction characteristics.

2.2 Offline evaluation metrics

Offline evaluations metrics, such as nDCG [17], are typically computed over a single SERP. While multiple alternatives to nDCG have been proposed, these alternatives generally focus on different models of interaction between the searcher and the SERP. For example, Moffat and Zobel [31] propose the position-based Ranked Bias Precision (RBP) evaluation metric, intended to model a searcher's persistence in scanning the SERP. However, RBP does not consider the possibility that a searcher may stop scanning once they see a relevant item. ERR [7] addresses this issue, modeling user behavior as varying depending on the relevance of each item. In this paper, we base discount models for 2d-Gain on inspiration provided by the discount models employed in nDCG, RBP, and ERR. In addition, the stopping model of ERR directly inspired our stopping criteria for 2d-Gain. On the importance of offline effectiveness metrics in evaluating information retrieval systems, the C/W/L framework has been introduced, which separates user actions from the benefit users derive as they exit the ranking. C/W/L and C/W/L/A allow for the systematic categorization of current effectiveness metrics and enables novel combinations to be considered [2, 28–30] In terms of SERP-level evaluations, Sakai and Zeng [33] evaluate ranked retrieval evaluation measures based on how well they match users' SERP preferences. The study compares traditional and preference-based evaluation measures based on SERP relevance and diversity preferences. The results suggest that measures such as nDCG perform best for traditional search, while diversified search measures based on the SERP diversity preferences are most reliable for diversified search. In addition, they found out that document preference-based measures do not align as well with users' SERP preferences and are not recommended over traditional measures. All in all, Sakai and Zeng [33] emphasize that the evaluation metric should be aligned with the user's SERP preferences and no single metric can provide a surrogate for user preferences in all cases. Similarly, Zhang et al. [39] explored the consistency between evaluation metrics and user satisfaction in the batch evaluation of IR systems. Their study primarily focused on whether metrics calibrated with user behavior data can perform as well as those calibrated with user satisfaction feedback in estimating user satisfaction.

2.3 Query autocompletion & session search

While query autocompletion [4, 11, 22, 27, 34] shares with instant search the goal of providing immediate feedback to the searcher, the evaluation of query autocompletion focuses on the suggestions themselves, which may be evaluated in terms of the SERP retrieved by each autocompletion suggestion [4, 23]. In contrast, we view instant search as a *two-dimensional process*, one dimension associated with character entry and the other dimension associated with the rank of each item on the SERP. Evaluation metrics and methods for instant search should reflect the two-dimensional nature of the interaction. Session search has also been modeled as a two-dimensional process [18, 19]. However, we differentiate instant search from session search in several ways. In instant search, every keystroke leads to a new SERP and the set of all keystrokes forms a sequence whereas in session search, each *complete* query results in a new SERP and a set of complete queries forms a session. As such, it is more likely for information need drift to occur in session search compared to instant search since each query is complete for each SERP and might point to a different aspect or subtopic related to the information need. In contrast, since each successive query in an instant search session (i.e., a sequence of SERPs) only differs from the previous one by a single character, we assume that the probability of information need drift is minimal.

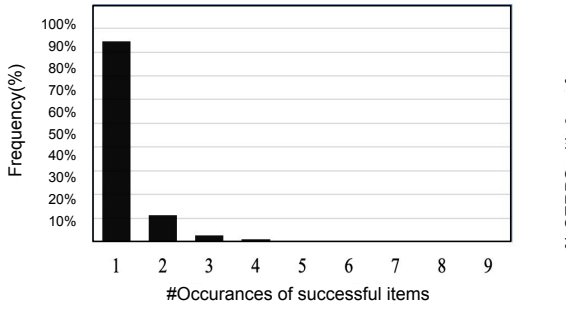

Figure 2: Histogram of the number of occurrences of successful entities in search sequences of $\texttt{LOG}_{\texttt{instant-audio}}$. The y-axis indicates the percentage of sequences that the successful items appeared in N times.

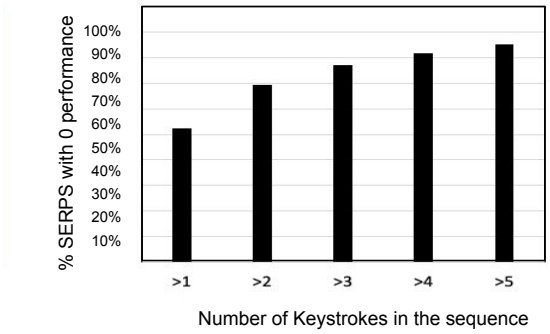

Figure 3: Percentage of SERPs with 0 nDCG score based on minimum number of keystrokes in the search sequences of $\texttt{LOG}_{\texttt{instant-audio}}$.

3 CHARACTERISTICS OF INSTANT SEARCH

In this paper, we argue that traditional evaluation metrics are not capable of capturing important aspects of how searchers interact with instant search, particularly the dependency between successive SERPs in a sequence, since the searcher experience in instant search starts by pressing a key and continues with their typing until either they find the desired entity or give up. The complete sequence of SERPs should be considered together for evaluation purposes instead of treating each individual SERP independently. In this paper, we focus on proposing an evaluation framework which is compatible with instant search. We now define the terminology we use throughout this paper. We also describe and justify our assumptions about user behavior in an instant search system.

3.1 Search sequences

A search sequence starts when a user presses any key in the search bar. An example of a keystroke could be adding a character or deleting one. Each keystroke results in a new SERP in the sequence. A sequence can be abandoned or terminated for a number of reasons including when a user is not able to find what they were looking

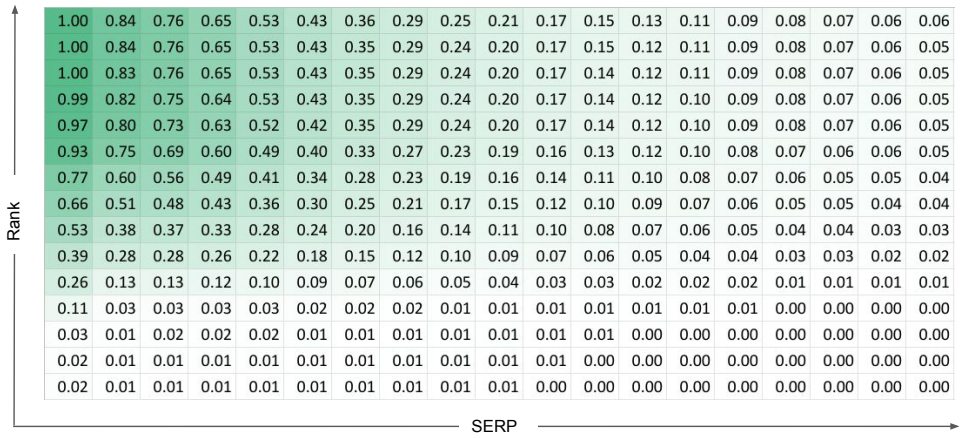

Figure 4: Estimated survival probability of entities at different rank and SERP levels derived from $\texttt{LOG}_{\texttt{instant-audio}}$. Darker cells indicate a relatively higher survival probability.

for, inactivity, or by clearing the search bar. We intentionally avoid using the term "session" since here the sequence mostly comprises *incomplete* queries. Incomplete queries are naturally similar to each other since they only differ by a single keystroke.

3.2 SERP rank and SERP level

A SERP contains the top retrieved entities for a given query, where the query can either be complete or incomplete (i.e. a *prefix* of the complete query). Each SERP is a ranked list of retrieved entities which start at position 1 of the ranking and can have a different number of ranked entities depending on different factors such as the user's screen size or whether the user has scrolled down on the SERP. We assign a *SERP level* to each SERP in the sequence which starts with SERP level 1, and each keystroke adds one level to it.

3.3 Successful sequence and entities.

A successful sequence is one that is terminated due to *search success*, e.g. streaming a song for longer than a defined threshold. For offline evaluation purpose, each query must have a known *successful entity* or *target entity*. On a commercial search platform (§5) the target entity is typically inferred from searcher behavior through business-specific rules. In the case of our synthetic experiments (§6) target entities are derived from lists of the most popular entities over a selected time period. For the experiments reported in this paper, we exclude searches that were not successful. Successful searches provide a target entry that allows to search sequences to be re-used for offline evaluation. Extending our work to accommodate abandonment is left for the future.

3.4 Searcher behavior

Offline evaluation metrics are essentially models of searcher behavior [5]. A metric such as nDCG can be viewed as the searcher scanning a search result list receiving some gain or utility from each relevant item seen. Gain accumulates as the searcher scans down the list. Other metrics, such as ERR [7], make the additional assumption that once a searcher has seen a relevant item, they are less likely to continue scanning, so that gain is discounted on future relevant items as each relevant item is seen. In the case of entity search, we adopt an extreme version of this assumption, that once the searcher has seen the target entity, they stop scanning. An example of our assumed search behavior is shown in Figure 1, where the searcher's intention is to find the song "*Easy on Me*" by Adele. The top and bottom rows represent different search sequences. On the top row, the target entity "*Easy on Me*" first appears in the third SERP, when the query consists of only three characters. However, the user does not notice the target entity at SERP level 2 and continues entering their query, adding a fourth character (i.e. "adel"). In the bottom row, we illustrate a sequence where the user stops as soon as they see the target entity i.e., the sequence terminates once they see the "Easy on Me" song in the forth SERP. We empirically explore how often these different behaviors occur among one million randomly sampled search log sequences—each with exactly one successful entity—from a commercial audio streaming platform. This data was collected throughout a single day in March 2022 for no specific reason. We refer to this query log as $\texttt{LOG}_{\texttt{instant-audio}}$ throughout this paper. This same log, along with data from the following day also forms the basis for the experiments reported in §5. We plot how often the target entity occurs in a search sequence in Figure 2. In more than 84% of sequences, the target entity was displayed only once in the sequence. In most cases, searchers continue the sequence only until they *first* retrieve what they are looking for. Only in 11% of the sequences target entities repeated twice and in less than 5% of the sequences are they repeated three or more times. These results suggest that searchers pay close attention to entities shown in each SERP as they type. The fact that successful items do not repeat supports the view that a better ranker is one that retrieves the relevant items in the earliest possible SERP since users are indeed paying attention to the items on each SERP. This analysis also suggests that searchers expect instant search systems to retrieve target entities with incomplete queries.

4 2D-GAIN

In §3 we highlighted differences between instant search and traditional search, such as web search. Due to these differences, instant search should be evaluated with a metric that reflects its unique characteristics. We now introduce our metric—2d-Gain.

A common approach for measuring the effectiveness of a ranked list involves summing over the product of a *discount function* of ranks and a *gain function* mapping relevance assessments and rank

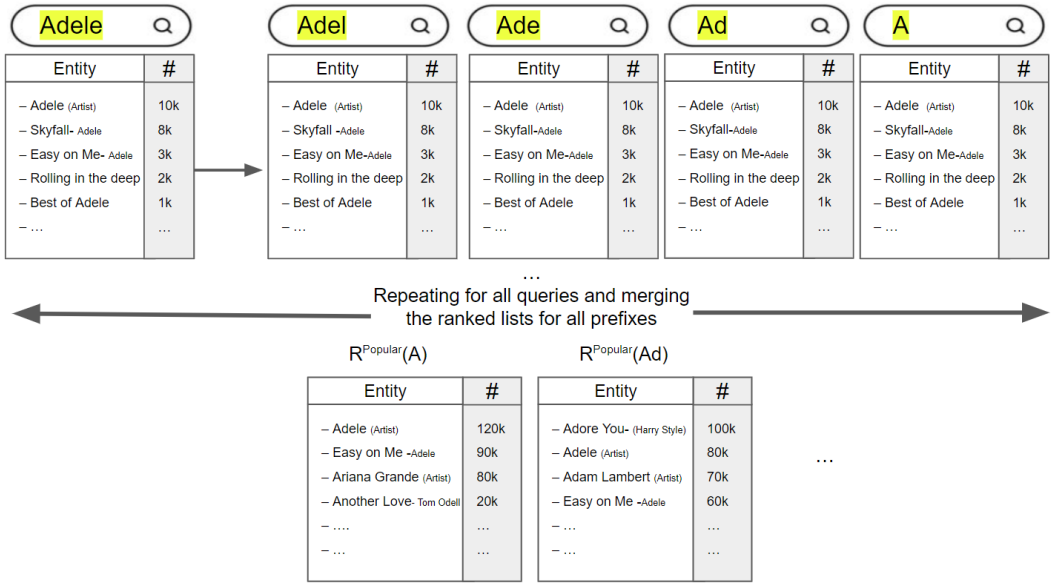

Figure 5: Example illustrating the creation of a popularity based ranker (R^P) from historic query logs.

to a scalar value, indicating the overall utility of the ranked list[5]. The measures are usually proposed based on underlying browsing model, a model of document utility, and a utility accumulation model. These models can be independently modified, allowing for the evaluation of different combinations of models in information retrieval systems. The discount function under these user models is commonly interpreted as representing a user who becomes increasingly disinterested as they scan down a ranked list, while the gain function represents the user's perception of the value of each document. However, this interpretation oversimplifies the many options available for constructing these measures. Different approaches can be taken, such as using a probability density function as the discount or applying dynamic or static discounts based on relevance, which can lead to nuanced variations in the user model. Under such an approach, an evaluation metric M is defined over $D = [d_i | i \leq K]$, a ranked list of top-k retrieved items d_i, as:

$$M(D) = \sum_{i=1}^{K} gain\,(rel(d_i)) \times discount(i) \tag{1}$$

where $rel(d_i)$ refers to the relevance level of the item which is ranked at position i and $discount(i)$ can be interpreted as a *survival probability*, i.e. the probability that the searcher will continue scanning to rank i.

Many metrics such as DCG, RBP and ERR follow the formulation shown in Equation 1, with the ERR gain function also considering the gain of the items ranked above d_i. These metrics are defined on the SERP level and do not explicitly consider the relationship between the SERPs in a sequence. To apply them to a sequence, they would need to be applied independently to each individual {query$_j$,SERP} pair of a multi-character search sequence in instant search. However, SERPs at the start of the search sequence are less likely to contain a relevant item, therefore have a zero score. It is only after the target entity first appears that SERPs will have a non-zero score, at which point the searcher will likely complete their search by clicking on the entity. Metrics for instant search much take this dependence between SERPs into account.

If we imagine applying nDCG independently to each SERP in a sequence, most SERPs will receive a score of zero. Figure 3 shows the percentage of SERPs with zero nDCG scores when applying nDCG to every SERP in the sequence. We calculate nDCG independently on each SERP, with the target entity as the sole relevant item. Among all successful queries that have at least 5 keystrokes (i.e. > 4 on the x-axis in Figure 3) in our dataset, over 81% of SERPs have zero nDCG scores.

Based on shortcomings of traditional search evaluation metrics for assessing instant search, we define the 2d-Gain of a sequence S_k^n of instant search results with n SERP levels and each SERP containing the top-k retrieved items as follows:

$$2d\text{-}Gain(S_k^n) = \sum_{j=1}^{n} \sum_{i=1}^{k} G_{ij} \times D(SERP_j, rank_i) \tag{2}$$

Here, G_{ij} is the gain of an item positioned at $rank_i$ in the j^{th} SERP of sequence S_k^n. In traditional IR frameworks [7, 17, 31], G_{ij} could be interpreted as a function of relevance of the entity positioned in $SERP_j$ of the sequence S_k^n at rank i. Moreover, $D(SERP_j, rank_i)$ indicates the *discount function* at each position in the sequence. The most important difference between this evaluation framework and typical evaluation metrics (i.e. Equation 3 vs. Equation 1) is that the discount factor is not only dependent on the rank, but considers both the SERP level and the entity rank. 2d-Gain also maintains the advantage of traditional frameworks with the flexibility to derive variations of the 2d-Gain evaluation metric by adapting different gain and discount functions according to the application under investigation and the goal of the search system [41]. For instance, any traditional IR metric can fit into our framework with appropriate changes to the gain and discount functions.

Equation 2 provides a general formulation of our metric, emphasizing its nature as a generalization of Equation 1. However, in light

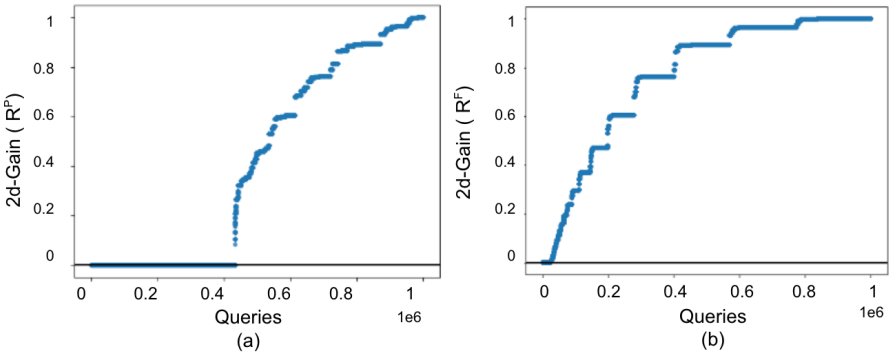

Figure 6: Performance of 2d-Gain using estimated discount rates from data ($\texttt{LOG}_{\texttt{instant-audio}}$) on (a) popularity-based ranker and (b) personalized full-featured ranker.

of the characteristics of instant search described in §3, we assume that the searcher will only receive benefit or gain from the target entity the first time they encounter it, and there is no partial gain, so G_{ij} is never less than 1. Therefore, in the case of instant search, $G_{ij} = 1$ the first time the searcher encounters the target entity; $G_{ij} = 0$ otherwise. As a result, we simplify equation 2 to recognize gain at the rank and level where $D(SERP_j, rank_i)$ is maximized:

$$2d\text{-}Gain(S_k^n) = \max_{i,j} \left(target_{ij} \times D(SERP_j, rank_i) \right), \tag{3}$$

where $target_{ij} = 1$ if and only if the target entity appears at rank i of SERP j. If the target appears more than once in a SERP sequence, we score it only at the position where the searcher is most likely to see it. As a result, $0 \leq 2d\text{-}Gain(S_k^n) \leq 1$ in this equation.

We could, of course, have made other assumptions. Perhaps a person searching for *Easy on Me* would be happy with *Someone Like You*, providing partial gain, but we leave the exploration of this and other ideas for future work. While equation 2 represents the general form of 2d-Gain, we use equation 3 for the experiments reported in the remainder of the paper.

We now propose three instantiations of the discount function in equation 3: (i) a data-driven instantiation; (ii) an instantiation using exponential decay, inspired by RBP and; (iii) an instantiation based on the nDCG discount function.

4.1 Approximating discounts from data

We interpret the discount function of our 2d-Gain framework as reflecting the probability of each item on different SERP levels and ranks being seen by the searcher. In instant search, we assume the first ranked entity of the first SERP is seen once the sequence is initiated. Depending on whether a searcher has already found what they are looking for in the first SERP, the searcher might continue typing and issuing keystrokes and perhaps move to a second SERP, third SERP, and so on. As the SERP level increases, there is a greater chance that the searcher has already found what they are looking for and is satisfied. Therefore, the items in earlier SERPs of a sequence have a greater chance to be visible to users compared to items in later SERPs. Similarly, the items at higher ranks of each SERP have a higher probability to be seen compared to entities ranked lower, which might even require scrolling to see them. To adapt these assumptions to the 2d-Gain discount function, we first study empirically how persistent searchers are in completing and refining their queries in sequences. To this end, from $\texttt{LOG}_{\texttt{instant-audio}}$ we estimate the survival probability of an entity at $[SERP_x, \text{rank}_y]$, i.e., its probability of being visible to the searcher. We adopt these estimates as approximating the discount function, although we recognize that this approximation does not consider abandoned SERPs and other factors.

We define the survival probability of entity E_{ij} which is ranked at SERP j (or d_i as in Equation 1), rank i as $P_s(E_{ij})$ which is the probability of an entity positioned at SERP j and rank i being visible to the user. Figure 4 demonstrates the normalized survival probability plot based on $\texttt{LOG}_{\texttt{instant-audio}}$. We limit both the number of the SERP and rank levels to 15. Darker cells indicate a relatively higher survival probability. We leverage the survival probability at each position, treating it as an approximation of the discount rate from data in our 2d-Gain formulation, i.e., in Equation 3 we replace $D(SERP_j, rank_i)$ with $P_s(E_{ij})$ where the probabilities are derived from the empirical survival probability estimates.

4.2 Exponentially decaying discount

As an alternative to the data-driven derivation of the discount function as seen in the previous section, we also consider a model-based approach. Inspired by traditional evaluation metrics, particularly RBP [31], we adapt an exponential decay function. Unlike nDCG, RBP's discount function is informed by click information, similar to our approach in the previous section. Concretely, we define $D(SERP_j, rank_i)$ as follows with $\alpha, \beta \in [0, 1]$:

$$D(SERP_j, rank_i) = exp\left(-(\alpha \times SERP_j + \beta \times rank_i)\right). \tag{4}$$

An advantage of this approach, compared to estimating the discount rate from data, is that adapting an exponential decay coefficient gives us the flexibility to balance the trade-off between the SERP and the rank level. This flexibility may lead to a more accurate evaluation of instant search for application scenarios where estimates of survival probabilities are not available.

4.3 Adapting the nDCG discount

Other traditional evaluation metrics can be adapted into the 2d-Gain metric by adapting the discount function to consider SERP level, as well as rank. As an example,we can fit the discounted cumulative gain (DCG) into 2d-Gain as follows, defining G_{ij} as:

$$G_{ij} = 2^{rel_{ij}} - 1 \tag{5}$$

where rel_{ij} indicates the relevance of the ith item on the jth SERP, and defining $D(SERP_j, rank_i)$ as:

$$D(SERP_j, rank_i) = \frac{1}{log(i+j)}. \quad (6)$$

This definition equates ranks and SERP levels, so that changing the rank of the target entity by one or changing the SERP level by one, changes the discount by the same amount. We can also normalize by ideal gain to yield nDCG. In the case of the experiments reported in our work, we have a single target entity and we count the gain only once, so that normalization is not required. In the experiments that follow, we replace the discount in equation 3 with equation 6.

5 CASE STUDY I: AUDIO STREAMING

Relying on data of an audio streaming platform, we illustrate how 2d-Gain can provide insights into the impact of personalization, which is an important aspect of many large-scale streaming services, and we highlight where personalization may fail. Recall that traditional evaluations metrics are not capable of reflecting sequence-level differences because they only consider a single ranking at a time. Moreover, if applied individually to each SERP in a sequence, they consider all SERPs in the sequence equally and do not reward rankers that place target items earlier in the sequence. For illustration purposes, we compare two specific rankers with 2d-Gain and show how leveraging such an offline evaluation framework can be useful in practice for assessing instant search systems. We note that in a production setting, we would directly compare alternatives of personalized full-featured rankers before deployment rather than compare against this simple baseline.

5.1 Rankers: R^F and R^P

Concretely, we compare a personalized full-featured ranker R^F and a popularity-based ranker R^P:

Personalized full-featured ranker: The ranker R^F makes use of multiple feature groups, including personalization features, as well as lexical and semantic matching signals. In terms of features and ranking model, this ranker is similar to a ranker that might be deployed in production.

Popularity-based ranker: The idea behind R^P is that for each query, we retrieve items based on the inferred targets of people who previously entered that query. The more popular an entity is for a given query q, the higher it is ranked by R^P. This simple setup captures the notion of a personalization-free ranker, where ranking is based solely on popularity, effectively omitting the effect of personalization from the full-featured ranker. To build R^P, we follow the following five steps:

(1) For each query q, we define the ranked list of top-k most popular successful items during a period of time T, as $S(q) = [(E_q^1, F(E_q^1)), (E_q^2, F(E_q^2)), ..., (E_q^k, F(E_q^k))]$ where E_q^1 is the most popular entity for q and E_q^k is the k^{th} popular item for q during that time period T. In addition, $F(E_q^k)$ indicates the number of sequences where E_q^k was selected as target entity for query q during period T. Since $S(q)$ is a ranked list, if $i > j$ then $F(E_q^j) \geq F(E_q^i)$.

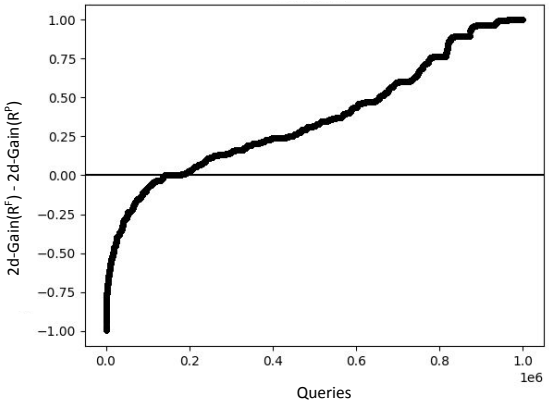

Figure 7: Difference between the 2d-Gain of R^P and R^F, using a discount function derived from log data (§4.1). Negative values indicate that R^P outperforms R^F.

(2) We determine the set of all prefixes $Pre(q)$ for query q (including q itself). For example, for the query `adele`, $Pre(\texttt{adele}) = \{\texttt{a}, \texttt{ad}, \texttt{ade}, \texttt{adel}, \texttt{adele}\}$.
(3) We assume that every successful entity for q (i.e., the set $S(q)$) was also a successful entity for *all its prefixes* because the ranker had the chance to retrieve the relevant item earlier when fewer keystrokes had been issued. Thus, for each prefix $p \in Pre(q)$, we obtain $S(p) = S(q)$. We thus obtain the top-k popular items for a query and its prefixes.
(4) Then, we aggregate the ranked list of popular entities for all queries and their prefixes. We aggregate $S(q)$ for all queries in our dataset $q \in Q$ as follows, if U is a set of all unique prefixes in Q i.e., $U = \{Pre(q) | q \in Q\}$:

$$S_{agg}(Q) = \{[(E_u, \sum_{q \in Q} \sum_{p' \in pre(q)} F(E'_p) | p' = u] \forall u \in U\} \quad (7)$$

For every prefix $u \in U$, we sum the frequency of the successful entities over all the queries.
(5) Now, for each unique query q or their prefixes $p \in Pre(q)$, we have a ranked list $S(q)$ or $S(p)$ from $S_{agg}(Q)$. We consider these ranked lists as the output of R^P for each query.

For clarity, in Figure 5 provides an example of the popularity-based ranker construction procedure. We repeat this procedure for all queries in $\texttt{LOG}_{\texttt{instant-audio}}$ — a single day of data. For testing we apply R^P to queries issued on the following day. Since we train on one day and test on the next, the ranker could theoretically be deployed in production, with a new ranker created each day.

5.2 Ranker Evaluation with 2d-Gain

In Figure 6 we compare the effectiveness of the two rankers using 2d-Gain. Given the availability of log data, we use the survival probability in the discount function which is estimated from $\texttt{LOG}_{\texttt{instant-audio}}$. As anticipated—also since the relevant labels are coming from the personalized full-featured ranker—the personalized full-featured ranker outperforms R^P. R^P fails to retrieve any successful (relevant) entities for more than 40% of the queries (over

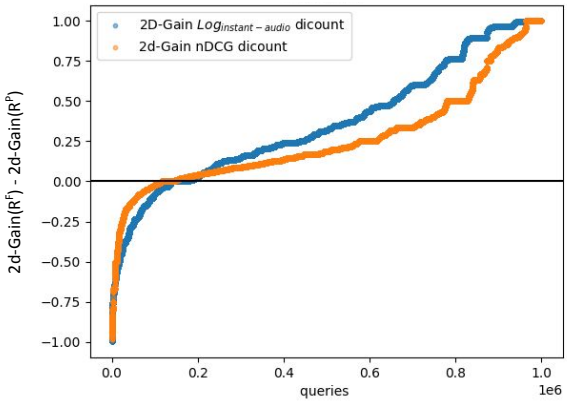

Figure 8: Comparison between the relative effectiveness of R^P and R^F using two different discount functions for 2d-Gain, one based on a discount on Figure 4 and one based on the nDCG discount of Equation 6. Negative values indicate R^P outperforms R^F; the reverse holds for positive values.

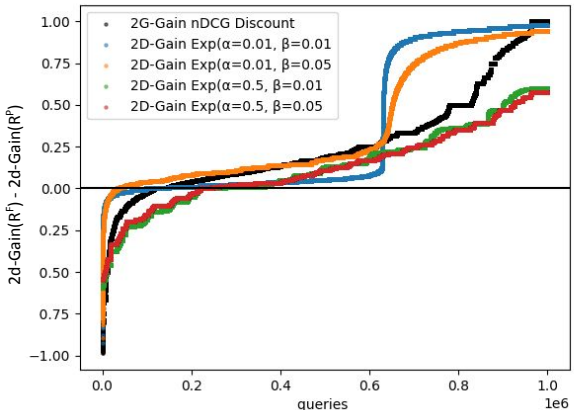

Figure 9: Comparison between the relative effectiveness of R^P and R^F using two different discount functions for 2d-Gain, one based on the exponential discount of Equation 4, with different values for α and β, and one based on the nDCG discount of Equation 6. Negative values indicate R^P outperforms R^F; the reverse holds for positive values.

400K queries). While R^F, with personalization, outperforms R^P on average, on specific queries R^P outperforms R^F. Figure 7 shows the difference between R^P and R^F on individual queries, ordered by increasing difference. For a subset of queries our popularity-based ranking outperforms R^F. These are queries for which personalization appears to hurt effectiveness. This outcome reflects the findings of [12, 35], who found personalization to harm some web queries.

Figure 8 and Figure 9 compare 2d-Gain under various discount functions. In addition, while there is no offline evaluation metric in the literature specifically designed for instant search, it is possible to leverage the proposed 2d-Gain framework in §4 and adapted the nDCG-inspired discount function to be applied to the results. Figure 8 shows that the data-based discount and the nDCG-inspired discount identify roughly the same number of queries for which the popularity-based ranker (R^P) outperforms the full-featured ranker (R^F). Additionally, Figure 9 shows the trade-off between SERP rank and SERP level as we vary α and β.

6 CASE STUDY II: WIKIPEDIA SEARCH

To provide an additional illustration of our approach, we apply it to the Wikipedia search box, which provides instant search over the entities in Wikipedia (Figure 10). As the searcher types, this search box lists matching entities and allows direct navigation to them[1]. While Wikipedia search also provides a full range of other search features — including content matching, proximity operators, and regular expressions — its core search feature closely matches the instant search paradigm.

To illustrate our metric, we also require a stream of queries reflecting the relative popularity of entities. While Wikipedia query logs are not public, we can approximate these logs with publicly available data that indicates the popularity of entities over various time periods[2]. By taking the top k queries over a specific time period of a day, week, month, etc., we can construct an artificial query stream, reflecting entities of interest during that period. In constructing this stream, we assume the searcher types characters

[1]https://en.wikipedia.org/wiki/Help:Searching
[2]https://pageviews.wmcloud.org/topviews

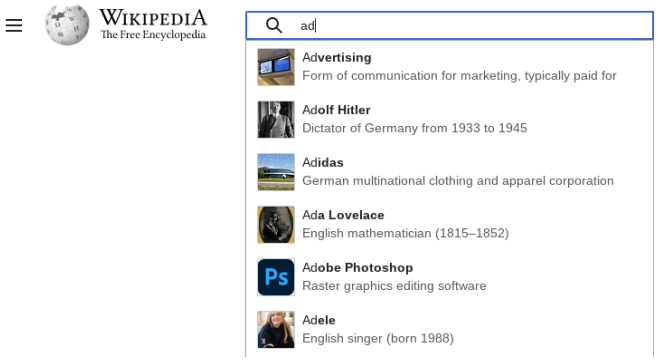

Figure 10: Example of Wikipedia's instant search.

of a target entity one-by-one until the target entity appears in the search results. We then compute 2d-Gain over the entirety of the sequence. The artificial query stream constructed for the experiments reported in this section is derived from the top-500 most popular entities for the month of December 2022, where the number of entities and time period are chosen arbitrarily for illustrative purposes. Due to the artificial nature of this query stream, we do not consider the experiments reported in this section to be a fully realistic and meaningful evaluation of Wikipedia search. However, the approach itself is realistic. If we had direct access to the Wikipedia query logs, this approach could be used to test and tune rankers. We also considered applying our approach to other entity-related test collections [3] but these are designed to test different entity-related tasks and do not fit the instant search paradigm.

6.1 Rankers: $R^{Wiki\text{-}F}$ and $R^{Wiki\text{-}P}$

To illustrate our approach, we compare two rankers, replicating as closely as possible the comparison in §5.

Full-featured ranker We take the Wikipedia's full-featured production ranker as it existed in early 2023 ($R^{Wiki\text{-}F}$). Using a Selenium script, we entered each query from our artificial query

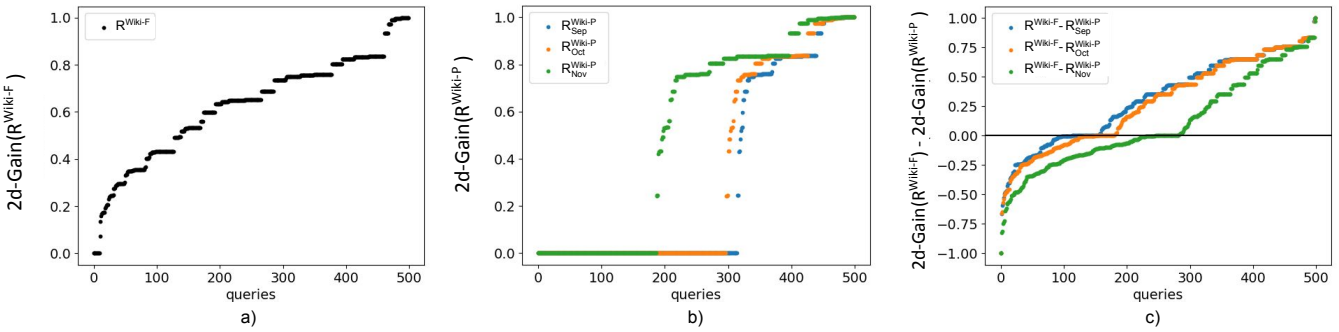

Figure 11: a) Performance of Wikipedia's production ranker (R^{Wiki-F}). b) Performance of three popularity-based rankers (R^{Wiki-P}) based on most viewed queries of September (Blue), October (Orange), and November(Green) 2022. c) Difference between the performance of Wikipedia ranker with three popularity rankers. The test queries in all subfigures comprise the top-500 most popular entities of Wikipedia from December 2022.

stream, one character at a time, capturing the output of the ranker after each character. In order to avoid blocking by bot detection systems, we gathered this data slowly over the course of several days. During this period the production ranker may have changed, and we have been subjected to A/B testing. We do not know if we are testing a consistent ranker, but the results are genuine in the sense that they would have been seen by some searcher typing an entity slowly into the search box during that period.

Popularity-based ranker As our second ranker, we construct a purely popularity-based ranker (R^{Wiki-P}) from the same public data source used to construct our artificial query stream, but from the previous months (September to November 2022). We instantiate a different version of this ranker for each month (R^{Wiki-P}_{Sep}, R^{Wiki-P}_{Oct}, R^{Wiki-P}_{Nov}). We base our rankers on months, rather than hours, days, or weeks, since we expect to see greater variations in popularity of entities from month to month, providing a clearer illustration of our approach. As each query is entered one character at a time, we match the prefix against the list of top 1000 entities from that month. If an entity does not appear in the top-1000 entities, it can never be returned by the ranker, where we choose 1000 entirely arbitrarily for illustrative purposes. Since it can only return entities from the top 1000, this ranker is obviously not deployable in production. Nonetheless, it serves as a simple baseline for comparison against the production ranker (R^{Wiki-F}) in order to illustrate our approach.

6.2 Ranker Evaluation with 2d-Gain

The absolute performance of the production ranker (R^{Wiki-F}) appears in Figure 11 as plot a), and the results of our comparison appear as plots b) and c). Part a) of the figure shows that there may be room for improvement in the production ranker over our artificial stream of the top 500 queries from December, with less than 3% of these queries receiving a 2d-Gain of 1. Part b) shows a steady decrease in performance as we move to earlier months. For all rankers, at least 35% of queries receive an 2d-Gain score of zero. Part c) of Figure 11 presents the 2d-Gain difference between R^{Wiki-F} and R^{Wiki-P}, ordered by increasing difference. Positive numbers indicate that R^{Wiki-F} outperformed R^{Wiki-P}, while negative numbers indicate that R^{Wiki-P} outperformed R^{Wiki-F}. As illustrated in the Figure, the November ranker (R^{Wiki-P}_{Nov}) outperforms the production ranker on this artificial query stream, with more negative difference than positive differences. The September ranker (R^{Wiki-P}_{Sep}) and October ranker (R^{Wiki-P}_{Oct}) perform less well, with the October ranker only slight better than the September ranker. While nothing definite can be concluded from these results — due to the artificial nature of the query stream and the simplicity of our popularity ranker — it may be that additional popularity features would improve the effectiveness of the Wikipedia ranker. If such features were proposed, our approach would serve as a suitable offline evaluation method.

7 CONCLUSIONS

Instant search provides the searcher with a complete SERP after each character they type, so that the searcher need only type a prefix of their query to find their target entity. Reflecting the relationship between queries in this sequence, an offline evaluation metric for instant search must operate over the sequence of SERPs taken as a whole, rewarding a ranker for placing the target entity higher on a SERP and earlier in the sequence of SERPs. To address this requirement, we define a general 2d-Gain evaluation metric for instant search, and describe several instantiations of the metric employing different discount functions. We provide experimental illustrations of our metrics on a commercial audio streaming platform, based on query logs from that platform, and Wikipedia, based on public data from that site. Popularity and personalization provide key features for instants search, and 2d-Gain allows us to explore the trade-off between these features, identifying queries which may be harmed by excessive personalization. Many search engines deploying instant search augment it with query suggestions and other aspects of traditional search, so that after a particular query prefix, the search results may consist of a blend of suggestions and entities. Clicking on the suggestion returns the corresponding SERP, while clicking on an entity takes the searcher to that entity. In future, we hope to extend our metric to this case, allowing us to trade-off a mixture of query suggestions and entities. We also aim at applying the metrics to other platforms and entity types [15], as well as extensions that accommodate query abandonment [20]. We hope to explore additional discount functions to determine the discount function that best models searcher satisfaction.

REFERENCES

[1] Negar Arabzadeh, Alexandra Vtyurina, Xinyi Yan, and Charles LA Clarke. 2022. Shallow pooling for sparse labels. *Information Retrieval Journal* 25, 4 (2022), 365–385.

[2] 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 *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 231–237.

[3] Krisztian Balog, Pavel Serdyukov, and Arjen P. de Vries. 2011. Overview of the TREC 2011 Entity Track. In *20th Text REtrieval Conference*. Gaithersburg, Maryland.

[4] Fei Cai, Maarten De Rijke, et al. 2016. A survey of query auto completion in information retrieval. *Foundations and Trends® in Information Retrieval* 10, 4 (2016), 273–363.

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

[6] Praveen Chandar, Jean Garcia-Gathright, Christine Hosey, Brian St. Thomas, and Jennifer Thom. 2019. Developing evaluation metrics for instant search using mixed methods methods. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 925–928.

[7] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 621–630.

[8] Charles LA Clarke, Nick Craswell, Ian Soboroff, and Azin Ashkan. 2011. A comparative analysis of cascade measures for novelty and diversity. In *Proceedings of the fourth ACM international conference on Web search and data mining*. 75–84.

[9] Charles LA Clarke, Fernando Diaz, and Negar Arabzadeh. 2023. Preference-Based Offline Evaluation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 1248–1251.

[10] William S Cooper. 1968. Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American documentation* 19, 1 (1968), 30–41.

[11] Giovanni Di Santo, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2015. Comparing Approaches for Query Autocompletion. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 775–778.

[12] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. 2007. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international conference on World Wide Web*. 581–590.

[13] Guglielmo Faggioli, Marco Ferrante, Nicola Ferro, Raffaele Perego, and Nicola Tonellotto. 2021. Hierarchical dependence-aware evaluation measures for conversational search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1935–1939.

[14] Jacek Gwizdka. 2010. Distribution of cognitive load in web search. *Journal of the American Society for Information Science and Technology* 61, 11 (2010), 2167–2187.

[15] Helia Hashemi, Aasish Pappu, Mi Tian, Praveen Chandar, Mounia Lalmas, and Benjamin Carterette. 2021. Neural instant search for music and podcast. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2984–2992.

[16] Enamul Hoque, Orland Hoeber, and Minglun Gong. 2011. Evaluating the trade-offs between diversity and precision for Web image search using concept-based query expansion. In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Vol. 3. IEEE, 130–133.

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

[18] Kalervo Järvelin, Susan L Price, Lois ML Delcambre, and Marianne Lykke Nielsen. 2008. Discounted cumulated gain based evaluation of multiple-query IR sessions. In *European Conference on Information Retrieval*. Springer, 4–15.

[19] Evangelos Kanoulas, Ben Carterette, Paul D. Clough, and Mark Sanderson. 2011. Evaluating Multi-Query Sessions. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1053–1062.

[20] Madian Khabsa, Aidan Crook, Ahmed Hassan Awadallah, Imed Zitouni, Tasos Anastasakos, and Kyle Williams. 2016. Learning to account for good abandonment in search success metrics. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 1893–1896.

[21] Sudarshan Lamkhede and Sudeep Das. 2019. Challenges in search on streaming services: Netflix case study. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1371–1374.

[22] Liangda Li, Hongbo Deng, and Yi Chang. *Query Auto-Completion*. Springer, Cham, 145–170.

[23] Liangda Li, Hongbo Deng, Anlei Dong, Yi Chang, Hongyuan Zha, and Ricardo Baeza-Yates. 2015. Analyzing user's sequential behavior in query auto-completion via markov processes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 123–132.

[24] Aldo Lipani, Ben Carterette, and Emine Yilmaz. 2021. How am I doing?: Evaluating conversational search systems offline. *ACM Transactions on Information Systems (TOIS)* 39, 4 (2021), 1–22.

[25] Zeyang Liu, Ke Zhou, and Max L Wilson. 2021. Meta-evaluation of conversational search evaluation metrics. *ACM Transactions on Information Systems (TOIS)* 39, 4 (2021), 1–42.

[26] Chuan Meng, Negar Arabzadeh, Mohammad Aliannejadi, and Maarten de Rijke. 2023. Query Performance Prediction: From Ad-hoc to Conversational Search. *arXiv preprint arXiv:2305.10923* (2023).

[27] Bhaskar Mitra and Nick Craswell. 2015. Query auto-completion for rare prefixes. In *Proceedings of the 24th ACM international on conference on information and knowledge management*. 1755–1758.

[28] 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 (TOIS)* 35, 3 (2017), 1–38.

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

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

[31] Alistair Moffat and Justin Zobel. 2008. Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems* 27, 1 (December 2008), 2:1–2:27.

[32] Heather L O'Brien and Elaine G Toms. 2008. What is user engagement? A conceptual framework for defining user engagement with technology. *Journal of the American society for Information Science and Technology* 59, 6 (2008), 938–955.

[33] Tetsuya Sakai and Zhaohao Zeng. 2020. Retrieval evaluation measures that agree with users' SERP preferences: Traditional, preference-based, and diversity measures. *ACM Transactions on Information Systems (TOIS)* 39, 2 (2020), 1–35.

[34] Milad Shokouhi. 2013. Learning to personalize query auto-completion. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 103–112.

[35] Jaime Teevan, Susan T Dumais, and Daniel J Liebling. 2008. To personalize or not to personalize: modeling queries with variation in user intent. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 163–170.

[36] Ganesh Venkataraman, Abhimanyu Lad, Lin Guo, and Shakti Sinha. 2016. Fast, lenient and accurate: Building personalized instant search experience at linkedin. In *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 1502–1511.

[37] Xiaohui Xie, Jiaxin Mao, Yiqun Liu, Maarten de Rijke, Yunqiu Shao, Zixin Ye, Min Zhang, and Shaoping Ma. 2019. Grid-based evaluation metrics for web image search. In *The world wide web conference*. 2103–2114.

[38] Emine Yilmaz, Milad Shokouhi, Nick Craswell, and Stephen Robertson. 2010. Expected browsing utility for web search evaluation. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. 1561–1564.

[39] Fan Zhang, Jiaxin Mao, Yiqun Liu, Xiaohui Xie, Weizhi Ma, Min Zhang, and Shaoping Ma. 2020. Models versus satisfaction: Towards a better understanding of evaluation metrics. In *Proceedings of the 43rd international acm sigir conference on research and development in information retrieval*. 379–388.

[40] Fan Zhang, Ke Zhou, Yunqiu Shao, Cheng Luo, Min Zhang, and Shaoping Ma. 2018. How Well do Offline and Online Evaluation Metrics Measure User Satisfaction in Web Image Search?. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 615–624.

[41] Ke Zhou, Hongyuan Zha, Yi Chang, and Gui-Rong Xue. 2012. Learning the gain values and discount factors of discounted cumulative gains. *IEEE Transactions on Knowledge and Data Engineering* 26, 2 (2012), 391–404.