PRE: A Precision-Recall-Effort Optimization Framework for Query Simulation

Sahiti Labhishetty
sahitil2@illinois.edu
University of Illinois Urbana-Champaign
Illinois, USA

ChengXiang Zhai
czhai@illinois.edu
University of Illinois Urbana-Champaign
Illinois, USA

ABSTRACT

We study how to develop an interpretable query simulation framework that can potentially explain the process a real user might have used to formulate a query and propose a novel interpretable optimization framework (PRE) for simulating query formulation and reformulation uniformly based on a user's knowledge state, where the three high-level objectives are to maximize the precision and recall of the anticipated retrieval results and minimize the user effort. We propose probabilistic models to model how a user might estimate precision and recall for a candidate query and derive multiple specific query formulation algorithms. Evaluation results show that the major assumptions made in the PRE framework appear to be reasonable, matching the observed empirical result patterns. PRE provides specific hypotheses about a user's query formulation process that can be further examined via user studies, enables simulation of meaningful variations of users without requiring extra training data, and serves as a roadmap for systematic exploration and derivation of new interpretable query simulation methods.

CCS CONCEPTS

• **Information systems** → **Users and interactive retrieval**; • **Computing methodologies** → Modeling and simulation.

KEYWORDS

Query simulation; Formal interpretable framework; Knowledge state

ACM Reference Format:

Sahiti Labhishetty and ChengXiang Zhai. 2022. PRE: A Precision-Recall-Effort Optimization Framework for Query Simulation. In *Proceedings of the 2022 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '22), July 11–12, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539813.3545136

1 INTRODUCTION

Modeling and simulating search engine users is essential for quantitative evaluation of Interactive Information Retrieval (IIR) systems. Indeed, it has been argued that simulation of users may be the only way to do any reproducible experiments with an IIR system [4, 51] since evaluation using real users (e.g., the A/B test) is inherently not reproducible or extensible to include any future IIR system in the experiments. Further, simulation will ensure control over experiments such that it can be used to evaluate the effects of different scenarios like different user types, user tasks, and "what-if" scenarios that are not frequent in real user datasets or online evaluations [4]. A formal model of a user is also essential for optimizing any IIR algorithms since the objective function to be optimized by such an algorithm must include a mathematical description of the user that it attempts to interact with. Moreover, an interpretable parameterized user simulation model where the parameters meaningfully correspond to different user behaviors can also be used as a tool to mine real user logs to identify interesting user search behavior patterns [34], in addition to simulating different user search behaviors.

However, formally modeling and simulating user interactions is very challenging for the following reasons.

First, the interaction process of a user with an IR system is an unobservable complex cognitive process where the user's knowledge and information need can also be frequently updated [9, 26]. Existing research in cognitive science and search user studies only provides a limited understanding of this process. We thus do not have a clear theoretical basis for modeling users mathematically. Second, user interaction has a lot of variance; users with the same information need may show different types of querying and clicking behaviors. Thus a simulator should also be able to vary in a meaningful way to simulate different kinds of user behaviors, such as variations in their knowledge background, patience, or trade-off between effectiveness and effort. Further, the simulator should also adapt to different information needs. Finally, evaluation of the user simulation model also poses multiple challenges [32]. While there has been some progress in evaluating user simulators empirically (e.g., the Tester-based evaluation approach [32, 33] and the multi-dimensional evaluation framework [12, 15]), there is a lack of progress in evaluating the soundness of the model behind a user simulator, which is primarily because the existing user simulators do not clearly articulate the assumed generative process that an actual user uses in generating the observed search behavior.

Due to these challenges, progress in the research of user simulation has been slow, especially in developing formal models that can provide an interpretable explanation of how users formulate queries. While many models and methods have been proposed for modeling clickthroughs (see e.g., [11, 17, 21, 23]), only a few methods have been proposed for query simulation [3, 5, 6, 13, 15, 30, 35].

There are two common deficiencies in all the existing approaches to simulating user query formulation. First, the existing approaches do not explain how or whether the simulation algorithm is based on an assumed generative process that an actual user may follow for formulating a query, making it hard to extract any meaningful

ICTIR '22, July 11–12, 2022, Madrid, Spain

ACM ISBN 978-1-4503-9412-3/22/07…$15.00
https://doi.org/10.1145/3539813.3545136

hypothesis about the whole process of a user's query formulation from such algorithms. Moreover, all the existing methods with few exceptions [14, 34, 35] do not incorporate a user knowledge state as an underlying variable for the query formulation process, even though a query formulated by a real user is clearly influenced by their knowledge. Second, although most algorithms adopted an implicit optimization framework to generate a query that is "optimal" by some criteria, the objectives to be optimized are often not explicitly articulated, nor can they be easily interpreted as meaningful objectives that a user could conceivably optimize when formulating a query, again making it hard to interpret those algorithms from the perspective of simulating real users.

One may argue that as long as the synthetic queries generated by a query simulator are similar to a real user's queries or give similar performance, the interpretability and how realistic the simulator is in terms of simulating the actual query generation phenomenon are not very critical, thus developing a highly interpretable simulator may be only theoretically interesting. However, we would argue that an interpretable simulator is not only theoretically interesting but also is of practical importance. When a simulator only has a high empirical validity but is not theoretically sound, it would have limited value in applications because its generalization capacity of generating realistic queries unseen in the training data set is questionable; yet, being able to generate unseen realistic queries is precisely why we need a simulator in the first place. Further, the best way to evaluate simulated queries is already an open challenge raising the question of the reliability of pure empirical validation.

In this paper, we address the above limitations of the existing work by proposing a novel interpretable Precision-Recall-Effort (PRE) optimization framework for simulating query formulation and reformulation. In the PRE framework, we make the following explicit assumptions (hypotheses) about how a user formulates a query: 1) A user is assumed to generate a query to maximize both the recall and precision of the anticipated retrieval results while minimizing the effort of making the query. 2) A user would attempt to estimate the recall and precision of the anticipated retrieval results based on the (current) knowledge state of the user. The knowledge of a user includes, as a minimum, knowledge about how the search engine would process a query (e.g., retrieving documents matching keywords in the query), and knowledge about the relevant and non-relevant (distracting) information (e.g., terms that may occur in relevant or non-relevant information items). 3) A user's knowledge state would be updated throughout a search session as the user learns during the search session; thus, reformulation of queries can be assumed to follow the same initial query formulation process but with an updated knowledge state.

The precision and recall are estimated using a Model of Precision (Prec) and a Model of Recall (Rec); both can be defined based on probabilistic models conditioned on the knowledge state of the user.

As an interpretable optimization framework for query formulation with explicit objectives, the proposed PRE framework has the following benefits that cannot be offered by existing approaches to query formulation simulation:

1. Explicit assumptions and hypotheses: PRE can be evaluated analytically to assess its soundness since all the assumptions made about a user would be not only explicit but also formally described. Experimental studies with real users can potentially help validate or further improve those assumptions, advancing our understanding of how users formulate a query.

2. Simulation of variable behaviors of users: The interpretability of the framework means that the parameters introduced in PRE can be interpreted as representing meaningful user variations, thus naturally tackling the challenge of modeling variable behaviors of users without requiring training data from all the users. For example, the influence of the three objectives (Precision, Recall, Effort) in the optimization objective function can be controlled flexibly by parameters; varying these parameters enables to simulate plausible different querying strategies that users may adopt without requiring additional user data for training.

3. Roadmap for systematic exploration of query formulation simulators: PRE can serve as a roadmap for systematic exploration of specific query simulation methods. Different ways to instantiate the framework would lead to many new query simulation methods. It further enables examination and comparison of the proposed query simulation methods in existing literature analytically as PRE covers existing approaches as specific instantiations.

4. Extensibility with other user modeling components: PRE can be extended by plugging in any existing models for modeling a user's knowledge state [14] into the PRE framework, incorporating other interpretable models related to a user's query formulation (e.g., economics models [2]), and combining it with other models of search user actions like click models.

As mentioned above, the PRE framework can be instantiated in many ways resulting in different query simulation methods for simulating queries. While a full exploration of this potential of PRE is out of the scope of this paper, we explore and study some basic strategies to instantiate the individual components of the PRE framework to derive multiple query simulation algorithms. These simulation algorithms are used to study the soundness of the design of PRE, that is, the high-level objectives to optimize and the uniform modeling of both initial and subsequent query formulations. Our experiment results show that optimizing both precision and recall when generating a query is indeed necessary in that optimizing only one of them has consistently resulted in lower performance compared to optimizing both together. This suggests that the objective function of PRE is reasonable, and conceptually the query formulation problem can be reduced to choosing a query to maximize both precision and recall. The results also show that the performance patterns of different query formulation methods in initial query formulation are similar to their patterns observed for query reformulation, suggesting that the uniform query formulation mechanism adopted in PRE is reasonable and it is possible to improve query formulation in a general way to impact both initial and subsequent query formulations. However, we observed that improving precision and improving recall does not have an additive effect on overall performance improvement, suggesting that there is a potential interaction between precision and recall models that needs to be further studied.

The main contribution of this paper is the introduction and study of a novel interpretable optimization framework PRE, which is based on hypotheses explaining how real users formulate their queries, formally connects the query formulation process with the knowledge state of a user, simulates both initial query formulation and subsequent reformulation in a uniform manner, and serve as a

roadmap for systematic exploration of many new specific query simulation models and algorithms. Query simulation algorithms can be used as part of a user simulation model for evaluating or optimizing IIR system, for generating synthetic data collections [15, 36, 40] and also individually for evaluating query suggestion algorithms [47].

2 RELATED WORK

User simulation has been studied in the past from different perspectives. For example, early studies focused on using user simulation for IR evaluation [18, 45]. It also has been used for analysing user search behavior patterns [34], training reinforcement learning algorithms [43], and recently for evaluating conversational search systems [42, 50]. While many click modeling approaches are proposed [11, 17, 21, 23], less work has been done on query modeling [3, 5, 6, 13, 15, 30, 35]. Some of the previous works attempted to simulate all user actions (e.g., [15, 34]).

Multiple query simulation methods have been proposed in the previous works, but no previous work has attempted to propose a query simulation method based on an explicit hypothesis about the process that a user may have used to formulate a query (especially the initial query), which is achieved in our proposed PRE framework. For example, query modification strategies have been studied and utilized in many methods (see, e.g., [5, 7, 13, 29, 30, 35, 47]). While those query modification strategies are realistic user strategies, the previous work has not attempted to explain why a user has chosen a particular strategy which may also be different at different times. The PRE framework offers a potential explanation of such behavior from the perspective of a user's attempt to optimize precision and/or recall while minimizing effort (editing a previous query involves less effort than generating a new query from scratch). Most existing query simulation algorithms are based on a language model of information needs (e.g., [3, 12, 15, 29, 35]). Most early simulators [3] are not parameterized and thus cannot simulate variable user behavior or preferences. A recent work [12] overcame this limitation by adapting the query change model [22, 49] for query simulation and enabling some limited user variation by introducing parameters to denote user preferences like preference to retain previous query terms or preference to stick to the topic. Almost all of these methods can be covered as special instantiations of the PRE framework. Large scale user search logs are also used for learning query simulation methods to rewrite [24] or reformulate queries [25, 28], but these methods rely on the availability of such data. In the PRE framework, the model of recall and precision can either be computed using TREC collections as pursued in our experiments or be trained using user search log data if available.

Multiple studies have investigated how learning occurs during the search process [10, 19, 37] and how different user characteristics may have an impact on learning [39, 48] including knowledge of the users. However, all the query simulation methods, with only a few exceptions ([14, 15, 35]) have failed to capture the impact of a user's knowledge state on query formulations. The PRE framework established a general probabilistic model to connect query formulation with knowledge state, enabling uniform modeling of initial query formulation and all subsequent query reformulations. The specific user knowledge state and update model we have explored in this study is similar to the strategy proposed by Maxwell et al. [35].

A significant limitation of all the existing work on query simulation is that the simulation methods are not grounded on any explicit hypothesis regarding how the real users actually formulate queries. Also, although the query scoring is optimized in most of the existing approaches, it is not explicitly explained what objectives are optimized, making it extremely hard to assess whether the objectives optimized by the existing algorithms actually reflect what a real user might aim to optimize while formulating a query. There are descriptive models of search behavior and search process [8, 9, 20, 27, 31, 41] but they are not mathematical models that can be applied for simulating user behavior or generating queries. For example, the microeconomic theory has been used to study the effort and effectiveness of querying and browsing [1, 2]. Our framework is also grounded on similar principles where we optimize the effectiveness and effort of queries for query formulation, and the microeconomic models can be potentially incorporated into our framework as additional constraints for optimization.

3 THE PRECISION-RECALL-EFFORT QUERY SIMULATION FRAMEWORK

3.1 Maximization of recall and precision

To develop an interpretable framework for query simulation, we must consider how a user might formulate a query. Logically, a user would want to generate a query that can retrieve relevant documents without retrieving any non-relevant ones, i.e., optimize the quality of the anticipated retrieval results. As Recall and Precision are two basic meaningful measures of quality of retrieval results from a user's perspective, it is reasonable to assume that a user would choose a query to maximize the expected recall and precision. However, how does a user estimate the expected recall and precision of a query?

To address this question, let's consider an example of a real query "collecting old US coins" from the TREC Session Track dataset [16], where the information need (IN) is to "Obtain information on how to start collecting old US coins." It is instructive to analyze how a user might have come up with such a query.

First, it is natural for the user to think about "US coins" as it is the general topic of the IN. We note that "US coins" might be the most popular term in the relevant documents, and a user's tendency to use such a popular term reflects the desire to match as many relevant documents as possible (i.e., maximize recall). Thus to maximize recall, a query is chosen such that it would "match" as many relevant documents as possible, where "match" indicates that the document is likely retrieved for the query by a search engine, and we will use the word "match" in this notion throughout the paper.

Formally, let $Match \in \{0, 1\}$ be a binary variable that denotes whether there is a match ($Match = 1$) or not ($Match = 0$) between a query q and a document d in the collection C. We use $p(Match = 1|q, d)$ to denote the probability that there is a match between a query q and document d. Let $\mathcal{R} \subset C$ be the subset of relevant documents. To maximize the recall of the anticipated results, a user can be reasonably assumed to come up with a query q that would maximize $p(Match = 1|q, d)$ for **all** relevant documents, which can be formally denoted by the following Model of Recall (Rec) component in the objective function

$$\textbf{Model of Recall}: \mathbf{Rec}(q,\mathcal{R}): \prod_{d\in\mathcal{R}} p(Match=1|q,d). \quad (1)$$

Note that the conjunctive expression here (instead of a disjunctive expression) encodes the objective of finding a q that can match every relevant document in $\mathcal{R}$.

However, maximizing recall is unlikely the only objective in a user's mind since such a query also tends to match "too many" documents, including non-relevant ones. Taking the previous example of the TREC real user query discussed earlier, the shorter query "US coins" alone might match many documents about current US coins that don't have information about old US coins nor about how to collect them. This explains why the user has further added the phrase "collecting old" which helps make the query more specific, resulting in the final query "collecting old US coins", which can be expected to have higher precision than the previous candidate "US coins" due to its discrimination against non-relevant documents. This example shows that in addition to maximizing recall by choosing a query that can match all relevant documents, a user may also attempt to maximize the precision of the retrieval results by choosing a query that does not match any non-relevant document.

Formally, to avoid matching non-relevant documents means to minimize the probability $p(Match = 1|q, d)$ for every non-relevant document $d \in C-\mathcal{R}$, which conceptually is equivalent to maximization of precision since the precision reaches a maximum when we do not retrieve any non-relevant documents. Thus, we assume that when a user composes a query, the user would also attempt to maximize the precision captured by the following Model of Precision (Prec) component in the objective function

$$\textbf{Model of Precision}: \mathbf{Prec}(q,\mathcal{R}): \prod_{d\in C-\mathcal{R}} (1-p(Match=1|q,d)), \quad (2)$$

where $C-\mathcal{R}$ is the set of non-relevant documents and $1-p(Match = 1|q, d)$ is the probability that q does not match d. Once again, the conjunctive, instead of disjunctive, relation here captures the goal of not matching any of the non-relevant document.

The Rec and Prec can be combined naturally into one single objective function to capture the objective of optimizing both recall and precision. Indeed, the product $\mathbf{Rec}(q,\mathcal{R})\mathbf{Prec}(q,\mathcal{R})$ is precisely the probability that query q matches every relevant document but does not match any non-relevant ones. Despite the theoretical attractiveness of using this product directly as an objective function, in reality, we often need to accommodate the inevitable tradeoff between recall and precision. Indeed, maximizing precision often means sacrificing recall and vice versa. For example, consider again the information need in the previous example with an extension, "Obtain information about old US coins, how to start collecting and selling them?". While the query "US coins" might have a higher recall but lower precision, the query "collecting and selling old US coins" may be the opposite and too specific to retrieve sufficiently many relevant documents. In some cases, increasing precision may miss to retrieve relevant results or lead to zero retrieved results if the document does not match the query completely or only matches part of the query. Thus recall and precision should be balanced to obtain satisfactory search results. The optimal tradeoff between them may depend on multiple factors, including user preference or specific information needs. For example, recall might be more important in the case of finding all literature articles to write a comprehensive survey, while precision may be more important if a user would simply want to know the major events today by finding a few relevant news articles. Thus a general query formulation framework must have a precision-recall weighting parameter to enable it to be sufficiently flexible to accommodate variable tradeoffs between precision and recall. In the proposed PRE framework, we will thus choose a query to maximize the following objective of a weighted combination of Rec and Prec

$$g(q,\mathcal{R},\alpha) = \alpha \log \mathbf{Rec}(q,\mathcal{R}) + (1-\alpha)\log \mathbf{Prec}(q,\mathcal{R}), \quad (3)$$

where $\alpha \in (0,1)$ is an interpretable weighting parameter to indicate the importance of recall relative to precision, which we can vary to simulate different user behaviors or user needs.

3.2 Modeling effort

While optimizing the query quality, the user also naturally wants to minimize the effort, which constitutes time and cognitive load of the user to make a query. In general, the effort required for formulating a query q, which is denoted by $\mathbf{E}(q)$, can be modelled by utilizing two types of work involved in querying.

1. Cognitive Effort: The first is the cognitive effort required to think about the query words; a query with more rare or difficult terms/words may be assumed to require more cognitive load and time from the user and thus have a higher effort.

2. Physical Effort: The second is the effort needed to physically communicate the query to the search engine, e.g., the effort required for typing in the query; shorter queries generally take less time/effort, and longer queries take more time/effort. The effort spent to type in the query can be based on the length of the query, which can be measured in terms of both the words and the letters in the query.

The effort function $\mathbf{E}(q)$ can be used either as an objective to be minimized or as a constraint like a max limit on query length ($|q| < l$), which is again equivalent to an objective function which is 0 if $|q| < l$ and ∞ if $|q| >= l$. An effective query often requires more effort (e.g., the query "collecting old US coins" is longer and takes more effort than "US coins", but it is more effective) thus, there is a tradeoff between maximizing the query quality (as reflected by the models of recall and precision) and minimizing effort. A user might spend more time/effort to make an optimal query or settle on a sub-optimal query which is easier and takes less effort. Thus a general framework for query formulation should also accommodate the above tradeoff, which we achieve by introducing a weight on the effort to control the tradeoff.

Thus, in general, the query formulation process is a multi-objective optimization process involving three (potentially conflicting) objectives: (1) Model of Recall: $\mathbf{Rec}(q,\mathcal{R})$; (2) Model of Precision: $\mathbf{Prec}(q,\mathcal{R})$; and (3) User Effort: $\mathbf{E}(q)$, leading to the following general Precision-Recall-Effort (PRE) optimization framework for query formulation and reformulation

$$\begin{aligned} q^* &= \arg\max_q g(q,\mathcal{R},\alpha) - \lambda \mathbf{E}(q) \\ &= \arg\max_q \alpha \log \mathbf{Rec}(q,\mathcal{R}) + (1-\alpha)\log \mathbf{Prec}(q,\mathcal{R}) - \lambda \mathbf{E}(q), \end{aligned} \quad (4)$$

where $\lambda > 0$ is an interpretable parameter controlling the tradeoff between query quality and user effort.

The PRE framework provides a general theoretical framework for simulating how a user formulates a query based on the assumption that the user has the following knowledge: (1) Knowledge about the collection of information items C; (2) Knowledge about relevant

item set $\mathcal{R} \subset C$; (3) Knowledge about how to estimate $p(Match = 1|q, d)$, i.e., how a search engine works. In reality, the users may not have accurate knowledge about any of these (if they did, they would be able to formulate a perfect query), especially in the initial stage of the search. As the user interacts with a search engine more, the user may gain more knowledge in all the three areas above. The PRE framework enables us to model the cognition process of a user during the search process by accommodating updating of any of the knowledge over time. In this way, the framework models the initial query formulation and subsequent reformulations in a uniform way with the difference only in the assumed knowledge of the user at the time of formulating a query. Next, we discuss how to model a user's knowledge state in detail.

3.3 Knowledge state

When we apply the PRE framework to simulate a user, the whole optimization problem must be framed in the context of a user and a user's knowledge, which we denote by K. While $\mathbf{Rec}(q, \mathcal{R})$ and $\mathbf{Prec}(q, \mathcal{R})$ capture user's objective to optimize, they are computationally complex. It is unlikely that a user would be able to keep track of all the relevant and non-relevant documents cognitively and follow the exact formulas to do the computation. Since how exactly a user stores knowledge about relevance is unknown, as an initial exploration of PRE, we start with the simplest model of a user's knowledge state, where we assume that the user would accumulate and summarize the knowledge about relevant documents in $\mathcal{R}$ with an aggregated prototype relevant document $\mathcal{R}_k$ and that about non-relevant documents in $C - \mathcal{R}$ with an aggregated prototype non-relevant document $\bar{\mathcal{R}}_K$; this way, the user would only need to keep track of these two prototype documents for representing relevant and non-relevant information, respectively, and the knowledge state K is mainly composed of two prototype documents $\mathcal{R}_K$ and $\bar{\mathcal{R}}_K$, i.e., $K = \{\mathcal{R}_K, \bar{\mathcal{R}}_K\}$.

Under such a prototype document assumption, we have $\mathcal{R} = \{\mathcal{R}_K\}$ and $C - \mathcal{R} = \{\bar{\mathcal{R}}_K\}$, both containing just one (prototype) document, thus the product is no longer needed in the definitions of Rec and Prec, leading to the following knowledge state-dependent models of recall and precision

$$\begin{aligned} \mathbf{Rec}(q, \mathcal{R}_K) &: p(Match = 1|q, \mathcal{R}_K), \\ \mathbf{Prec}(q, \bar{\mathcal{R}}_K) &: 1 - p(Match = 1|q, \bar{\mathcal{R}}_K), \end{aligned} \tag{5}$$

Adding the effort model $\mathbf{E}(q)$, we obtain the PRE framework,

$$q^* = \arg\max_q \alpha \log \mathbf{Rec}(q, \mathcal{R}_K) + (1 - \alpha) \log \mathbf{Prec}(q, \bar{\mathcal{R}}_K) - \lambda \mathbf{E}(q). \tag{6}$$

It is reasonable to assume that initially, a user's knowledge about $\mathcal{R}_K$ is mainly based on a brief description of the information need since the user has not yet seen any relevant document, and a user's knowledge about $\bar{\mathcal{R}}_K$ can be assumed to be based on a general sense about the content in a collection C. As a user interacts more with a search engine, the user would be able to see more examples of relevant and non-relevant documents and thus update the user's knowledge by adding more information about relevant documents to $\mathcal{R}_K$ and more information about non-relevant documents to $\bar{\mathcal{R}}_K$. The exact form of updating depends on how exactly the user stores the knowledge about $\mathcal{R}_K$ and $\bar{\mathcal{R}}_K$. We will further discuss this issue in Section 4 under the assumption that the user would store the knowledge in the form of a unigram language model, i.e., the probability that a word w is seen in a relevant prototype document ($p(w|\mathcal{R}_K)$) or in a non-relevant prototype document ($p(w|\bar{\mathcal{R}}_K)$). The updating of a user's knowledge increases the user's capacity to potentially formulate a better query, and the PRE framework naturally captures this by simply using a more enriched knowledge state of the user for formulating a query.

The explicit connection of the query formulation (Rec and Prec) with a user's knowledge state (K) in PRE not only enables the modeling of initial query formulation and subsequent reformulations in a uniform way (as it should be), but also allows for meaningful variations of the simulated users by simulating different knowledge backgrounds of users; for example, the novice users vs. expert users can be simulated by varying how $\mathcal{R}_K$ is initialized. It also enables a natural integration of modeling query formulation with modeling the cognition of users during the search process.

So far, we have explained all the major elements in PRE, but we have not yet explained how a user might estimate $p(Match = 1|q, d)$, which is the basis for computing the models of both recall and precision in the objective function. This has to do with a user's knowledge about how a search engine works and can be potentially instantiated in many ways, which we will elaborate in the next section. The existing query formulation methods can generally be interpreted as special instantiations of the PRE framework.

4 INSTANTIATION OF THE FRAMEWORK

An important benefit of PRE is that by instantiating each component in PRE in different ways, we can systematically explore and study many new query simulation algorithms in the same unified framework. As an initial step, we propose some basic instantiations of PRE using statistical language models (LMs), leaving a thorough exploration as future work.

4.1 Conjunctive vs. Disjunctive matching

The major component that we need to instantiate is the matching likelihood $p(Match = 1|q, d)$, which models a user's assessment of whether a document d matches a query q. Without any additional knowledge about the user, we propose and study two complementary basic interpretations of matching:

Conjunctive matching: In this interpretation, we assume that a user's notion of "matching" is that document d matches all the words in query q, i.e., $p(Match = 1|q, d) = \prod_{w \in q} p(w|d)$. With this interpretation, the model of recall can be refined as follows,

$$\mathbf{Rec}(q, \mathcal{R}_K) = \prod_{w \in q} p(w|\mathcal{R}_K). \tag{7}$$

We note that such a conjunctive interpretation has an inherent bias toward favoring short queries since adding a word to a query would cause the product to be smaller. Intuitively, this bias makes sense since it would be easier to match all the words in a shorter query than in a longer query. However, we want to consider variable lengths of the query when finding an optimal query; thus, we need to normalize the product using the query length. We thus add an exponent $1/|q|$ to the product, where $|q|$ is the total number of words in the query (query length), giving an interpretation of the "per-word" query probability in the conjunctive recall model (Cr).

$$Cr : \mathbf{Rec}(\mathbf{q}, \mathcal{R}_\mathbf{K}) = (\prod_{w \in q} p(w|\mathcal{R}_K))^{1/|q|}. \tag{8}$$

We can similarly refine the model of precision to obtain the following conjunctive precision model (Cp)

$$Cp : \mathbf{Prec}(\mathbf{q}, \mathcal{R}_\mathbf{K}) = 1 - (\prod_{w \in q} p(w|\bar{\mathcal{R}}_K))^{1/|q|}. \quad (9)$$

Disjunctive matching: Alternatively, we can also interpret matching as the document matching any one of the query words, i.e., $p(Match = 1|q, d) = \frac{1}{|q|} \sum_{w \in q} p(w|d)$. Applying this interpretation, we can obtain the following disjunctive recall model (Dr) and disjunctive precision model (Dp)

$$Dr : \mathbf{Rec}(q, \mathcal{R}_K) = \frac{1}{|q|} \sum_{w \in q} p(w|\mathcal{R}_K), \quad (10)$$

$$Dp : \mathbf{Prec}(q, \bar{\mathcal{R}}_K) = 1 - \frac{1}{|q|} \sum_{w \in q} .p(w|\bar{\mathcal{R}}_K) \quad (11)$$

Note that the disjunctive interpretation is naturally normalized without any length bias and the conjunctive and disjunctive interpretations can be combined, leading to four different instantiations of the PRE framework. Moreover, we can use a higher-order n-gram language model (e.g., a bigram language model) to replace the unigram language model in either of the two interpretations above to potentially achieve more accurate modeling and generate more variations. Thus PRE can serve as a roadmap for us to explore better models for query formulation by systematically improving each component model.

4.2 Instantiation of Effort

We instantiate the effort function as a constraint such that only queries of length less than a threshold L would be allowed, i.e., $\mathbf{E}(q)$: $|q| <= L$. As $\mathbf{E}(q)$ is a constraint, it is equivalent to taking $\mathbf{E}(q)$ as an indicator function and setting λ parameter (weight of $\mathbf{E}(q)$ in Eq. 4) to a sufficiently large constant such that the length boundary L would in effect act as the weight parameter influencing the importance of effort.

4.3 Solving optimization problem

Even with a restricted length, the complexity in solving the optimization problem is still exponential as the query can contain any words in any order. To address this problem, following previous work [2, 3, 15], we use a modified greedy algorithm to first generate candidate queries and then find the optimal query. Using the vocabulary of words in the prototype document $\mathcal{R}_K$, we first enumerate all the one-word and two-word queries as an initial set of candidate queries. The two-word candidate queries are then expanded greedily by adding a word that maximizes the whole query score, creating increasingly longer queries until L-word candidate queries are created. All the candidate queries (with variable lengths) are finally ranked according to their optimization function score resulting in a ranked list of candidate queries. The top query is taken as the simulated query. During reformulation, the previously simulated queries are ignored from the candidate query list so as not to duplicate previous queries.

4.4 Knowledge state update

As described in Section 3.3, a user's knowledge K is $\{\mathcal{R}_K, \bar{\mathcal{R}}_K\}$. We assume that the initial knowledge of the user regarding relevant and non-relevant information is based on information need description (s) and collection (C), respectively. That is, $\mathcal{R}_K = s$ (s is the only relevant document) and $\bar{\mathcal{R}}_K = C$ (all documents in the collection are non-relevant and can be concatenated into one single prototype non-relevant document). This assumption is appropriate for building (dynamic) user simulators based on static TREC collections [15].

During a search session, the user can scan through the search results to learn new information. In this process, the user would be exposed to examples of both relevant and non-relevant documents, which can then be used to update $\mathcal{R}_K$ and $\bar{\mathcal{R}}_K$, respectively, to enrich the representation of relevant and non-relevant information.

With the unigram language model instantiations the simulated user would only need to store the knowledge about relevance in the form of two unigram LMs, i.e., the relevance LM $p(w|\mathcal{R}_K)$ and the non-relevance LM $p(w|\bar{\mathcal{R}}_K)$, which model the probability that word w occurs in a prototype relevant document and non-relevant document, respectively. With such a knowledge storage model, updating of knowledge boils down to updating these two LMs by aggregating word counts from the newly acquired examples of both relevant and non-relevant documents as follows

$$p(w|\mathcal{R}_K) = \frac{c(w, s) + \sum_{sr} c(w, sr) p(R = 1|sr, s)}{\sum_{w'} c(w', s) + \sum_{sr} c(w', sr) p(R = 1|sr, s)}, \quad (12)$$

$$p(w|\bar{\mathcal{R}}_K) = \frac{c(w, C) + \sum_{sr} c(w, sr) p(R = 0|sr, s)}{\sum_{w'} c(w', C) + \sum_{sr} c(w', sr) p(R = 0|sr, s)}, \quad (13)$$

where $c(w, s)$ gives count of the word w in s, $c(w, C)$ is the count of the w in the collection C, sr is a snippet of a retrieval result, and $p(R = 1|sr, s)$ and $p(R = 0|sr, s)$ are the probability that snippet sr is relevant or non-relevant, respectively, estimated based on known relevance judgments of the corresponding documents. Once the relevance and non-relevance LMs are updated, they can be used to reformulate a query using the PRE framework. As in the case of refining the models of recall and precision, more sophisticated language models and knowledge updating mechanisms can be easily plugged into the PRE framework.

5 EXPERIMENT DESIGN

The purpose of our experiments is to study the soundness and benefit of the proposed framework and answer the following research questions: **RQ1:** Do the empirical results support the design of the objective function of PRE, i.e., maximization of both precision and recall? **RQ2:** Do the improvement of the precision and recall instantiations have an additive effect on the overall performance improvement, or is there a complex interaction between precision and recall models? **RQ3:** Does the relative performance of different query simulation methods follow a similar trend for both initial and reformulated queries? In the rest of this section, we describe how we design our experiments to answer these questions.

Dataset: We use TREC Session Track 2012, 2013 and 2014 data sets [16] because they are among the very few data sets with the initial and reformulated queries formulated by real users, which we need for evaluation. Each topic is an information need with a title and description. Each topic's description is used as an information need description s to perform query simulation. We used unigram word frequencies derived from Google Web Trillion Word Corpus [38] available for the top 333,333 words as the collection language model $p(w|C)$. We used indri and pyndri [46] for indexing the ClueWeb datasets. For evaluating the simulated queries, we compared them with the real user queries of the same information need obtained from the sessions in the Session track (St for short)

datasets. St 2012 and 2013 only have 200 and 400 user queries respectively, compared with 3600 user queries in St 2014. Thus we can expect that the evaluation with St 2014 is more consistent and robust. Further, St 2012 has completely different topics compared to St 2013, 2014 whereas St 2013 and St 2014 share some topics.

Query similarity measurement: To assess the quality of a simulated query, for every simulated query, we computed its maximum Jaccard similarity with any of the real user queries of the same information need. We use Jaccard similarity because the TREC topic descriptions used by the real users and our simulation model are the same, so it is likely that the users have used most of the words from this topic description. The average of these similarities for all simulated queries of all TREC topics in the dataset is computed as average Jaccard similarity (*Avg_jsim*). *Avg_jsim* can be computed with *top-k* simulated queries from the ranked list generated by *PRE*. The reason to compute maximum similarity is that the simulated query can be similar to any one of the real user queries and need not be close to all.

$$Avg\_jsim@k = \frac{1}{|T|}\sum_{t\in T}\frac{\sum_{Q\in smt_{t,k}}\max_{q\in act_t}(jsim(Q,q))}{|smt_{t,k}|} \quad (14)$$

where $jsim$ is Jaccard similarity, act_t is the set of real user queries, $smt_{t,k}$ is the set of top-k queries generated for that topic and T is the set of all topics (information needs).

In addition to Jaccard similarly, we also compute F-measure score of the simulated queries. For each simulated query, we compute how many words of the real query are covered (recall) and how many words in simulated query are actually in the real query (precision). We use average recall and average precision to compute the final F-measure score. Similar to Jaccard similarity, we compute maximum recall and maximum precision score for a simulated query when compared to a real user query.

$$Avg\_prec@k = \frac{1}{|T|}\sum_{t\in T}\frac{\sum_{Q\in smt_{t,k}}\max_{q\in act_t}(prec(Q,q))}{|smt_{t,k}|}$$
$$Avg\_recall@k = \frac{1}{|T|}\sum_{t\in T}\frac{\sum_{Q\in smt_{t,k}}\max_{q\in act_t}(recall(Q,q))}{|smt_{t,k}|} \quad (15)$$

where $prec(Q,q) = \frac{|Q\cap q|}{|Q|}$, $recall(Q,q) = \frac{|Q\cap q|}{|q|}$.

Parameter estimation and setting: We estimate parameter α in Eq. 4 using four fold cross-validation; it can vary from 0 to 1 and we used grid search with 0.1 step. We set the effort parameter, which is the threshold of the effort function or the maximum query length L to 6. We have also studied the effect of varying the parameters α and L on the performance results.

To establish statistical significance between the performance of two different methods, we perform an independent two-sample t-test with a p-value of 0.05; the p-value is chosen as 0.05 as there are very few instances, just the number of TREC topics, of *Avg_jsim*.

6 EXPERIMENT RESULTS

We first compare different instantiations of Rec and Prec and the four combinations of Rec and Prec instantiations, which we refer to as combination methods, in Table 1 and Table 2. Table 1 and Table 2 show the average Jaccard similarity and F-measure scores of all the instantiations. We can infer from Table 1, that combination methods are always better than individual Rec or Prec methods

Table 1: *Avg_jsim*@5 for Rec, Prec combination methods using L=6 for Session Track 2012-2014 data.

Jaccard similarity	**Session track datasets**		
Methods	**St 2012**	**St 2013**	**St 2014**
CrCp	0.2536 (0.05)	0.3612 (0.475)	0.4431 (0.125)
CrDp	0.267 (0.1)	0.3539 (0.1)	0.436 (0.1)
DrCp	**0.253 (0.125)***	**0.3651 (0.85)**	**0.4504 (0.7)**$^{*,!}$
DrDp	**0.273 (0.1)**	0.3533 (0.1)	0.4436 (0.1)
Cr	**0.2529**	**0.3435**	**0.4124**
Dr	0.2407	0.3312	0.4045
Dp	0.2038	0.1453	0.313
Cp	0.2538	0.2108	0.3573
QS3+ (baseline)	0.2696	0.2368	0.4049

Table 2: $F-measure$@5 for Rec, Prec combination methods using L=6 for Session Track 2012-2014 data.

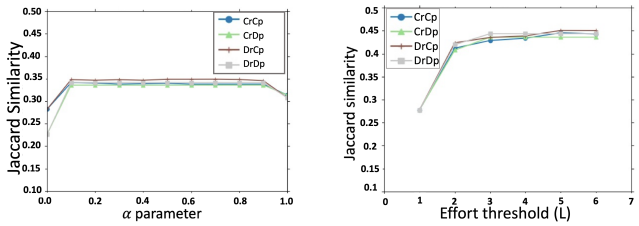
F-measure	**Session track datasets**		
Methods	**St 2012**	**St 2013**	**St 2014**
CrCp	0.3893 (0.05)	0.5324 (0.275)	0.6373 (0.15)
CrDp	0.3991 (0.1)	0.5169 (0.1)	0.6081 (0.1)
DrCp	0.3876 (0.125)	**0.5344 (0.875)***	**0.6474 (0.625)**$^{*,!}$
DrDp	**0.4071 (0.1)**	0.5167 (0.1)	0.616 (0.1)
Cr	**0.3855**	**0.4958**	**0.5895**
Dr	0.3663	0.4822	0.5854
Dp	0.3161	0.2436	0.4702
Cp	0.3836	0.3459	**0.5569**
QS3+	0.3886	0.3726	0.5664

(which are special cases when we set α to either 1 or 0), i.e., *CrCp*, *CrDp*, *DrCp*, *DrDp* always perform better than *Cr*, *Dp*, *Cp*, *Dp* in all datasets. Similar trends can also be observed in Table 2. In Table 1 and 2, ! indicates statistically significant difference between *DrCp* and *Cr* which are best methods among combination methods and individual component methods respectively. This shows that both Rec and Prec are important in the query formulation process; using recall or precision alone always performs less than combining them together. The result provides some empirical justification for the design of the objective function of the PRE framework to optimize both precision and recall, thereby answering **RQ1**.

Among individual components, *Cr* always performs better than *Dr* and *Cp* is performing better than *Dp*. However, among the combination methods, we observe that combining *Dr* with Prec methods performs better than combining *Cr* with Prec methods. *DrCp* performs better than *CrCp* and *DrDp* also performs better *CrDp*. But in the case of precision, *Cp* mostly performs better than *Dp* even after combining with a recall model. This implies that combining the best individual methods in Rec and Prec may not always result in the best method overall, indicating that there may be a complex interaction between the Rec and Prec models, which has to be further studied to optimize the component models of PRE (addressing **RQ2**).

As a reference point, we also include the performance of a state-of-the-art baseline model QS3+ [35] in Table1 and Table2. We see that even the very basic instantiations we have studied in this paper can already outperform this baseline method, $*$ indicates a statistically significant difference between *DrCp* and *QS3+* in Table 1 and 2, suggesting great potential for using PRE to further

Figure 1: *Avg_jsim*@5 with varying α and L



(a) varying α **(b) varying effort threshold (L)**

Table 3: *Avg_jsim*@5 of reformulated queries

Methods	St 2012	St 2013	St 2014
CrCp	0.2520	0.3639	0.4333
CrDp	0.2606	0.3575	0.4264
DrCp	0.2523	**0.3743**	**0.4383**
DrDp	**0.2670**	0.3616	0.4360

develop more effective query simulation methods. Among all the PRE instantiations, *DrCp*, the disjunctive instantiation of recall (*Dr*) combined with conjunctive instantiation of precision (*Cp*) appears to be most effective for query simulation except in St 2012 dataset. However, as these are only basic instantiations of PRE, we expect to be able to achieve better performance in the future as we further explore more sophisticated ways to instantiate PRE (e.g., higher-order n-gram LMs).

Table 1 and Table 2 also show the optimal parameter value of α obtained through cross-validation in the parenthesis for each method. For *Cr*, *Cp*, $\alpha = 1.0$ and for *Dr*, *Dp* $\alpha = 0.0$ as they are recall only and precision only methods respectively. For the remaining methods, we observe that the optimal α is mostly 0.1, with the exception of *DrCp*, which has $\alpha = 0.85$ and $\alpha = 0.7$ as the best parameter. Overall, the optimal parameter is never 0.0 or 1.0, which again shows that considering both recall and precision gives the best performance for the instantiations. To address **RQ3**, we now perform query reformulation using search results of the initial query with the knowledge state update method described in Section 4 to see if some conclusions we have made on initial query formulation also hold for reformulated queries. In Table 3, *Avg_jsim*@5 is computed by comparing the reformulated queries with actual queries. We observe similar trends in performance in Table 1 and Table 3. The reformulation method further ensures that *DrCp* is the best method in most cases, and similarly *DrDp* performs best on St 2012. One reason why St 2012 results are consistently different from other datasets could be the small size of the dataset and the different topic set compared to the remaining datasets. Similar trends are also observed for the F-measure score. These results suggest that the uniform modeling of initial formulation and subsequent reformulation is reasonable and also facilitates optimization of a query simulator by optimizing knowledge state updating and query generation separately.

6.1 Analysis of parameters

We analyzed the sensitivity of all the combination methods with respect to parameter α as shown in Figure 1a. Figure 1a shows *Avg_jsim*@5 for different α's overall the folds in cross-validation. The performance is low for $\alpha = 0.0$ and 1.0, and mostly the same for α between 0.1 to 0.9, and a similar trend is observed for both Jaccard similarity and F-measure. Thus, it can be concluded that as long as both recall and precision are considered in the optimization framework, the performance of different instantiations is not highly affected by α. These results also indicate that the PRE instantiations consistently outperform baseline model QS3+ irrespective of the parameter α as long as both recall and precision are considered.

We have studied the distribution of best α for different users in the St 2014 dataset as each session is associated with a *user id*. For each user, we have estimated the best α that optimizes *Avg_jsim*@5 of the simulated queries by grouping sessions of that user. We could obtain two interesting observations from these results. First, the performance variation by varying α for each user is similar to that of the average variation in Figure 1a, further supporting our conclusion that considering both recall and precision is necessary. That is, $\alpha = 0.0$ and 1.0 are non-optimal not only for the overall average performance with all sessions but also for each individual user's sessions. Second, we examined the best α for those users where the performance is most sensitive to α, and found that the best α varies to a great extent between each user, which indicates that real user behavior varies with different tradeoffs between recall and precision; thus, not only should we maximize both recall and precision, but we also must have the parameter α in PRE in order to accurately simulate individual users' variable tradeoff between precision and recall.

As described in Section 5, we have set maximum length threshold (L) to 6, $\mathbf{E}(q) : |q| < 6$, for obtaining results in Table 1, 2, 3. To study the sensitivity of this parameter, we have varied the threshold L from 1 to 6 and the performance of different combination methods are shown in Figure 1b, we observe that the performance of all the methods increases in the beginning from $L = 1$ to 3 and then is mostly the same for $L = 3$ to 6. Indeed, when the threshold is 6, we observed that most of the optimal queries are of length 2 or 3 and the average query length of the simulated queries is 1.97 for *St 2014*. This shows that the PRE framework cuts down the query length automatically for the optimal query even with higher threshold parameter. At the same time, it can also simulate long queries (even of length five) as shown by the example simulated queries in Table 4. Therefore, unlike many existing simulation approaches which choose a constant query length, PRE adaptively simulates long or short queries depending on the information need and setting L to 6 will allow for simulating a longer query whenever it is optimal.

6.2 Qualitative analysis

In Table 4, we provide a few examples of the simulated queries from top-10 queries generated by the *DrCp* method and similar real user queries for those topics. From the results, it can be observed that the quality of the queries highly depends on the information need description as it is the only information used. If the important words representing the information need are frequent in the description, then simulated queries are close to the user queries and vice versa; for example, as "swahili dish", "hydropower" appear many times in the topic description so the simulated queries are close to user queries for the first and third topics in Table 4, whereas the important keyword "world cup" appears only once in the last example topic which is probably the reason for poor quality simulated queries, thus this also leads to poor performance sometimes. The results also show that the query length can either be long or short depending on the information need; for example, in the second

Table 4: Examples of simulated queries

Topic description	Similar real user queries	Simulated queries
A friend from Kenya.....surprise him .... cooking a traditional Swahili dish.. learn about Swahili dishes and how to cook them. Find..about Swahili home cooking.	swahili food traditional, swahili dish wiki, swahili traditional dish, swahili dish, swahili cook.	swahili, swahili dish, dish, traditional swahili dish.
You are planning a winter vacation to the Pocono Mountains region... Where will you stay? What will you do while there? How will you get there?	pocono mountain winter activity, winter vacation in the pocono mountain, pocono mountain region.	winter pocono mountain vacation plan, pocono mountain region winter vacation plan, pocono pennsylvania winter vacation plan.
Hydropower...renewable sources of energy..replace fossil fuels. Find information about the efficiency of hydropower, the technology.. consequences building hydroelectric dams..on the environment.	hydropower efficiency, what be hydropower, hydropower damn, hydoelectric power, hydropower environment	hydropower, hydropower hydroelectric, hydropower dam, hydropower efficiency.
France won..World Cup in 1998. Find information about the reaction of French people and institutions (such as stock markets), and studies about these reactions.	1998 world cup french reaction, french world cup 1998, french reaction win world cup, stock market world cup 1998.	reaction, 1998 reaction, cup reaction, reaction institution, france reaction.

example of Table 4 the queries generated are very long, similar to real user queries which are also long.

7 DISCUSSION AND FUTURE WORK

Ideally, user simulation should be done based on a computational user model that can explain how users make various decisions (e.g., querying or viewing documents) in the search process and suggest an operational algorithm that can generate user actions in response to a search engine. However, we are currently far from such a model. For example, none of the current methods for query simulation suggests any hypothesis about the process that a user might have followed in formulating a query. This is partly due to the lack of understanding of the internal cognitive and reasoning processes in a user's mind, which requires more progress in research in cognitive science, information science, and human-computer interaction.

We can approach such an ideal computational model of users in two directions: 1) We can do user studies to sufficiently understand search users to be able to design a computational user model to simulate the understood user behavior. 2) We can design a computational user model that is as explanatory as possible of the user behavior. The proposed PRE framework can be regarded as taking the second direction and making a step toward building an interpretable and explanatory computational model for simulating a user's query formulation/reformulation process. PRE is designed based on multiple hypotheses about how exactly a user formulates a query. Our preliminary evaluation results provide some evidence to support the hypotheses, but clearly, more research is needed to further assess the validity of the hypotheses proposed in PRE, especially via designing appropriate user studies or leveraging search log data to more rigorously examine those hypotheses.

As a novel framework, PRE opens up many interesting opportunities for future research.

First, PRE can easily accommodate the incorporation of or combination with additional formal models of user behavior to further advance the development of interpretable computational models of users. For example, any click models or browsing behavior models can be combined with PRE to provide a more complete simulation model of a user.

Second, from a practical viewpoint, PRE offers three lines of immediate benefits: 1) It provides a general formal framework that we can use to compare the existing query simulation methods and systematically examine their effectiveness from the perspective of optimizing precision and recall and minimization of effort. 2) It facilitates the design of many new simulation models via different ways to instantiate each component of the PRE framework. This is a promising direction as even the very basic instantiation strategies explored in this paper already deliver comparable performance to one of the frequently used state of the art query simulation method *QS3+* [35]. For example, *DrCp* instantiation method outperforms *QS3+* for Session track 2013 and 2014 datasets. There is much potential to further improve the effectiveness of query simulation using the PRE framework by improving instantiations of the objective models of PRE. 3) Its interpretable parameters enable simulation of meaningful user variations in multiple dimensions, including, e.g., variation in the relative importance of precision and recall, variation in user's initial knowledge state, variation in effectiveness of a user's learning during search session, and variation in the tradeoff between the effort and effectiveness of the formulated query. Exploration of such directions is an important future work.

Third, PRE provides a theoretical roadmap for further exploration of new ways to potentially model how a user estimates precision and recall more accurately. We discuss two specific possibilities here. 1) As a more intuitive way to factor out precision and recall, the models of precision and recall can be defined more similarly to the precision and recall measures on retrieval results as follows,

$$\textbf{Expected\_precision (q)} = \frac{\sum_{d \in C}(p(Match{=}1|q,d)p(rel|d)}{\sum_{d \in C} p(Match{=}1|q,d)},$$

$$\textbf{Expected\_recall (q)} = \frac{\sum_{d \in C}(p(Match{=}1|q,d)p(rel|d)}{\sum_{d \in C} p(rel|d)},$$

where $p(rel|d)$ is the probability that the document d is relevant. 2) The models of recall and precision can also be computed over subtopics or facets of relevant information (like in [14]) instead of the set of relevant documents. The set of relevant documents of an information need often consists of multiple subtopics or facets and to satisfy the information need, the user may want to care more about maximizing recall and precision of the *subtopics* than those of the documents in the retrieved results. Such a new model has the potential for modeling query formulation in the context of complex retrieval tasks.

Finally, the conclusions we have drawn in our experiments can be further examined by performing more experiments and using additional evaluation measures. Our use of Jaccard similarity is justified based on the fact that the simulator model and most of the real users used the same word set from the topic descriptions, but it does not support inexact matching of semantically related words. Although whether to support inexact matching appears to be orthogonal to the hypotheses we are testing, meaning that adding inexact matching will unlikely have a significant impact on the conclusions we have drawn, it is necessary to further experiment with other evaluation measures [13, 32] including new semantic similarity measures such as BLEURT [44] in the future to further verify our conclusions.

REFERENCES

[1] Leif Azzopardi. 2011. The economics in interactive information retrieval. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 15–24.

[2] Leif Azzopardi. 2014. Modelling interaction with economic models of search. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 3–12.

[3] Leif Azzopardi, Maarten De Rijke, and Krisztian Balog. 2007. Building simulated queries for known-item topics: an analysis using six european languages. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 455–462.

[4] Krisztian Balog, David Maxwell, Paul Thomas, and Shuo Zhang. 2022. Report on the 1st simulation for information retrieval workshop (Sim4IR 2021) at SIGIR 2021. In *ACM SIGIR Forum*, Vol. 55. ACM New York, NY, USA, 1–16.

[5] F. Baskaya. 2014. Simulating Search Sessions in Interactive Information Retrieval Evaluation. PhD thesis, University of Tempere.

[6] F. Baskaya, H. Keskustalo, and K. Jarvelin. 2011. Simulating simple and fallible relevance feedback. In *Proceedings of ECIR*.

[7] Feza Baskaya, Heikki Keskustalo, and Kalervo Järvelin. 2013. Modeling behavioral factors in interactive information retrieval. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2297–2302.

[8] Marcia J Bates. 1989. The design of browsing and berrypicking techniques for the online search interface. *Online review* (1989).

[9] Nicholas J Belkin, Robert N Oddy, and Helen M Brooks. 1982. ASK for information retrieval: Part I. Background and theory. *Journal of documentation* 38, 2 (1982), 61–71.

[10] Nilavra Bhattacharya and Jacek Gwizdka. 2018. Relating eye-tracking measures with changes in knowledge on search tasks. In *Proceedings of the 2018 ACM symposium on eye tracking research & applications*. 1–5.

[11] Alexey Borisov, Ilya Markov, Maarten De Rijke, and Pavel Serdyukov. 2016. A neural click model for web search. In *Proceedings of the 25th International Conference on World Wide Web*. 531–541.

[12] Timo Breuer, Norbert Fuhr, and Philipp Schaer. 2022. Validating Simulations of User Query Variants. In *Advances in Information Retrieval*, Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty (Eds.). Springer International Publishing, Cham, 80–94.

[13] Timo Breuer, Norbert Fuhr, and Philipp Schaer. 2022. Validating Simulations of User Query Variants. *arXiv preprint arXiv:2201.07620* (2022).

[14] Arthur Câmara, David Maxwell, and Claudia Hauff. 2022. Searching, Learning, and Subtopic Ordering: A Simulation-based Analysis. *arXiv preprint arXiv:2201.11181* (2022).

[15] Ben Carterette, Ashraf Bah, and Mustafa Zengin. 2015. Dynamic test collections for retrieval evaluation. In *Proceedings of the 2015 international conference on the theory of information retrieval*. ACM, 91–100.

[16] Ben Carterette, Evangelos Kanoulas, Mark Hall, and Paul Clough. 2014. *Overview of the TREC 2014 session track*. Technical Report. DELAWARE UNIV NEWARK DEPT OF COMPUTER AND INFORMATION SCIENCES.

[17] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 7, 3 (2015), 1–115.

[18] Michael D Cooper. 1973. A simulation model of an information retrieval system. *Information Storage and Retrieval* 9, 1 (1973), 13–32.

[19] Carsten Eickhoff, Jaime Teevan, Ryen White, and Susan Dumais. 2014. Lessons from the journey: a query log analysis of within-session learning. In *Proceedings of the 7th ACM international conference on Web search and data mining*. 223–232.

[20] David Ellis. 1993. Modeling the information-seeking patterns of academic researchers: A grounded theory approach. *The Library Quarterly* 63, 4 (1993), 469–486.

[21] Artem Grotov, Aleksandr Chuklin, Ilya Markov, Luka Stout, Finde Xumara, and Maarten de Rijke. 2015. A comparative study of click models for web search. In *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, 78–90.

[22] Dongyi Guan, Sicong Zhang, and Hui Yang. 2013. Utilizing query change for session search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 453–462.

[23] Fan Guo, Chao Liu, and Yi Min Wang. 2009. Efficient multiple-click models in web search. In *Proceedings of the second acm international conference on web search and data mining*. 124–131.

[24] Yunlong He, Jiliang Tang, Hua Ouyang, Changsung Kang, Dawei Yin, and Yi Chang. 2016. Learning to rewrite queries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 1443–1452.

[25] Amaç Herdagdelen, Massimiliano Ciaramita, Daniel Mahler, Maria Holmqvist, Keith Hall, Stefan Riezler, and Enrique Alfonseca. 2010. Generalized syntactic and semantic models of query reformulation. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. 283–290.

[26] Peter Ingwersen. 1996. Cognitive perspectives of information retrieval interaction: elements of a cognitive IR theory. *Journal of documentation* 52, 1 (1996), 3–50.

[27] Peter Ingwersen. 2005. *Integrative framework for information seeking and interactive information retrieval*. na.

[28] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*. 387–396.

[29] Chris Jordan, Carolyn Watters, and Qigang Gao. 2006. Using controlled query generation to evaluate blind relevance feedback algorithms. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*. 286–295.

[30] Heikki Keskustalo, Kalervo Järvelin, Ari Pirkola, Tarun Sharma, and Marianne Lykke. 2009. Test collection-based IR evaluation needs extension toward sessions–a case of extremely short queries. In *Asia Information Retrieval Symposium*. Springer, 63–74.

[31] Carol Collier Kuhlthau. 1988. Developing a model of the library search process: Cognitive and affective aspects. *Rq* (1988), 232–242.

[32] Sahiti Labhishetty and Chengxiang Zhai. 2021. An Exploration of Tester-based Evaluation of User Simulators for Comparing Interactive Retrieval Systems.. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1598–1602.

[33] Sahiti Labhishetty and ChengXiang Zhai. 2022. RATE: A Reliability-Aware Tester-Based Evaluation Framework of User Simulators. In *European Conference on Information Retrieval*. Springer, 336–350.

[34] Sahiti Labhishetty, Chengxiang Zhai, Suhas Ranganath, and Pradeep Ranganathan. 2020. A Cognitive User Model for E-Commerce Search. In *Proceedings of the Data Science for Retail and E-Commerce Workshop*.

[35] David Maxwell and Leif Azzopardi. 2016. Agents, simulated users and humans: An analysis of performance and behaviour. In *Proceedings of the 25th ACM international on conference on information and knowledge management*. ACM, 731–740.

[36] David Maxwell and Leif Azzopardi. 2016. Simulating interactive information retrieval: Simiir: A framework for the simulation of interaction. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 1141–1144.

[37] Felipe Moraes, Sindunuraga Rikarno Putra, and Claudia Hauff. 2018. Contrasting search as a learning activity with instructor-designed learning. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 167–176.

[38] Peter Norvig. 2008. *Natural Language Corpus Data: Beautiful Data*. http://norvig.com/ngrams/

[39] Heather L O'Brien, Andrea Kampen, Amelia W Cole, and Kathleen Brennan. 2020. The role of domain knowledge in search as learning. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*. 313–317.

[40] Teemu Pääkkönen, Jaana Kekäläinen, Heikki Keskustalo, Leif Azzopardi, David Maxwell, and Kalervo Järvelin. 2017. Validating simulated interaction for retrieval evaluation. *Information Retrieval Journal* 20, 4 (2017), 338–362.

[41] Peter Pirolli and Stuart Card. 1999. Information foraging. *Psychological review* 106, 4 (1999), 643.

[42] Alexandre Salle, Shervin Malmasi, Oleg Rokhlenko, and Eugene Agichtein. 2021. Studying the Effectiveness of Conversational Search Refinement Through User Simulation. In *European Conference on Information Retrieval*. Springer, 587–602.

[43] Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*. 149–152.

[44] Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. BLEURT: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696* (2020).

[45] Jean Tague, Michael Nelson, and Harry Wu. 1980. Problems in the simulation of bibliographic retrieval systems. In *Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*. 236–255.

[46] Christophe Van Gysel, Evangelos Kanoulas, and Maarten de Rijke. 2017. Pyndri: a Python Interface to the Indri Search Engine. In *ECIR*, Vol. 2017. Springer.

[47] Suzan Verberne, Maya Sappelli, Kalervo Järvelin, and Wessel Kraaij. 2015. User simulations for interactive search: Evaluating personalized query suggestion. In *European Conference on Information Retrieval*. Springer, 678–690.

[48] Barbara M Wildemuth. 2004. The effects of domain knowledge on search tactic formulation. *Journal of the american society for information science and technology* 55, 3 (2004), 246–258.

[49] Hui Yang, Dongyi Guan, and Sicong Zhang. 2015. The query change model: Modeling session search as a markov decision process. *ACM Transactions on Information Systems (TOIS)* 33, 4 (2015), 1–33.

[50] Shuo Zhang and Krisztian Balog. 2020. Evaluating conversational recommender systems via user simulation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1512–1520.

[51] Yinan Zhang, Xueqing Liu, and ChengXiang Zhai. 2017. Information retrieval evaluation as search simulation: A general formal framework for ir evaluation. In *ACM ICTIR*. ACM, 193–200.