Learning to Rank Relevant and Novel Documents through User Feedback

Abhimanyu Lad
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, USA
alad@cs.cmu.edu

Yiming Yang
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, USA
yiming@cs.cmu.edu

ABSTRACT

We consider the problem of learning to rank relevant and novel documents so as to directly maximize a performance metric called Expected Global Utility (EGU), which has several desirable properties: (i) It measures retrieval performance in terms of relevant as well as novel information, (ii) gives more importance to top ranks to reflect common browsing behavior of users, as opposed to existing objective functions based on set-coverage, (iii) accommodates different levels of tolerance towards redundancy, which is not taken into account by existing evaluation measures, and (iv) extends naturally to the evaluation of session-based retrieval comprising multiple ranked lists. Our ground truth is defined in terms of “information nuggets”, which are obviously not known to the retrieval system when processing a new user query. Therefore, our approach uses observable query and document features (words and named entities) as surrogates for nuggets, whose weights are learned based on user feedback in an iterative search session. The ranked list is produced to maximize the weighted coverage of these surrogate nuggets. The optimization of such coverage-based metrics is known to be NP-hard. Therefore, we use a greedy algorithm and show that it guarantees good performance due to the submodularity of the objective function. Our experiments on Topic Detection and Tracking data show that the proposed approach represents an efficient and effective retrieval strategy for maximizing EGU, as compared to a purely-relevance based ranking approach that uses Indri, as well as a MMR-based approach for non-redundant ranking.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Relevance Feedback

General Terms

Algorithms, Experimentation, Measurement, Performance

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0099-5/10/10 ...$5.00.

Keywords

novelty-based ranked retrieval, nuggets, user feedback

1. INTRODUCTION

There has been growing interest in building and optimizing retrieval systems with respect to multiple criteria like relevance, novelty, and diversity of information [30, 6, 22, 29]. However, each of the current approaches is based on its own objective function that does not fully capture all the factors that are essential for realistic evaluation and optimization of systems with respect to relevance and novelty.

One of the most common modes of interaction with retrieval systems (*e.g.*, search engines) is ranked retrieval, where the system produces a list of documents ordered by decreasing relevance. However, so far novelty detection has not received much attention in a ranked retrieval setting. For instance, the TREC novelty track [24], which is representative of the research on novelty detection, assumed a fixed chronological order of documents, and the system's task was to merely detect the relevant and novelty sentences, without re-ordering them. While such a setting helped in isolating the novelty detection task and greatly simplified the creation of ground truth (*i.e.*, pre-judged relevant and novel sentences), it obviously does not reflect how users interact with today's retrieval systems. Similarly, diversity-based retrieval has been treated as a set retrieval problem: The objective function is reduced to the set-covering problem [30, 1], which does not differentiate between orderings of documents. This leads to an undesirable gap between set-based objective functions and the ranked-based evaluation metrics that are of ultimate interest.

Moreover, users often have different tolerances towards redundancy, as was also noted by [1, 4, 8]. While some users only want to see previously unseen documents, other users might desire a certain level of redundancy in the ranked list for various reasons like corroboration of information, or assessing the consensus or opinions on a single topic or product based on different news sources, reviewers, or blogs. However, none of the existing approaches to novelty or diversity-based ranking take this into account.

Accounting for redundancy in a principled manner is especially important when dealing with multiple ranked lists, for example, in an interactive session with a Web search engine, where the user goes through multiple rounds of query reformulation and feedback. It is not clear how to model novelty across multiple ranked lists. Simple strategies exist, *e.g.*, penalizing or removing documents that were already presented, or are similar to already-presented documents. Such an approach is reasonable in an adaptive filtering setting [32], but not for ranked retrieval: Users generally do not read all documents presented to them in a ranked list (especially in long ranked lists produced by search engines). They are more likely to read the top-ranked documents, and stop at some position based on their patience or satisfaction. A possible solution would be to assume a fixed position where users stop in each ranked list, *e.g.*, assuming that all users read the top ten documents in each ranked list [14]. However, any single stopping position would be a crude approximation of the dynamic behavior of real users and would completely ignore the documents below the cut-off rank for the purpose of evaluation. Instead, a probabilistic user model is desirable to model the browsing behavior of a population of users: A document that was presented at a very low rank in a previous ranked list has a smaller likelihood of being read by the user, and hence, should be discounted appropriately for the purpose of estimating the novelty of subsequent documents, thus leading to a probabilistic notion of novelty that extends naturally to multiple ranked lists in a search session.

Finally, the non-independent nature of novelty raises new challenges for learning from user feedback, which might be available in explicit (*e.g.*, "like"/"dislike" buttons) or implicit (*e.g.*, clicks) form. Since the utility of each document depends on other documents shown to the user, the feedback provided by the user should also be interpreted with respect to previously seen documents. In other words, user feedback is an indicator of the *marginal* utility of documents, instead of its absolute usefulness with respect to the user's information need. However, the latter assumption has been commonly used for learning from relevance feedback, *e.g.*, through regression [16, 28] or language modeling [31].

To summarize, accurate evaluation and optimization of retrieval systems must be based on a performance measure with the following properties: (i) It should take both relevance and novelty into account, (ii) give more importance to top ranks to reflect common browsing behavior of users, (iii) accommodate different levels of tolerance towards redundancy, and (iv) extend naturally to the evaluation of session-based retrieval comprising multiple ranked lists. In this paper, we develop a retrieval strategy for optimizing a recently proposed performance measure called Expected Global Utility (EGU) [27] that satisfies the above-mentioned criteria. We also propose a logistic regression based approach for learning from user feedback that takes the non-independent nature of document utility into account.

2. PROPOSED APPROACH

Our approach is based on maximizing a recently-proposed metric called Expected Global Utility (EGU) [27] that combines all the above-mentioned criteria in a principled manner. Relevance and novelty are modeled in terms of "information nuggets". The gain received from relevant nuggets follows a diminishing returns property to account for the reduced utility of seeing repeated information. However, how to directly optimize a retrieval system with respect to such a metric remains an open challenge, which we aim to address in this paper.

"Information nuggets", or simply "nuggets", is a concept borrowed from question answering evaluation [10]. A nugget is an atomic piece of information that is either present or absent from a given document. Thus, the answer keys for each user query can be defined in terms of the nuggets that satisfy the query. For example, the query "BP oil spill" would have the following nuggets: "fire started on April 20, 2010", "nine crew members and two engineers died", "top hat attempt", "top kill attempt", and so on[1]. A document's utility (*i.e.*, relevance and novelty) depends on the nuggets it contains, and whether these nuggets have been seen by the user in previously displayed documents. The goal of a retrieval system is to generate ranked lists that would maximize the utility received by the user.

There are several advantages to using nuggets as retrieval units. Nuggets can be used as answer keys to create reusable test collections for relevance and novelty-based retrieval evaluations. Previous novelty-based evaluations (*e.g.*, TREC [24] and TDT [2]) used documents or sentences as retrieval units and depended on a fixed (chronological) order of retrieval, which is clearly unrealistic for ranked retrieval. Moreover, the use of nuggets allows a more natural definition of novelty where a given document can be deemed as redundant based on two or more previously seen documents, instead of a single near-duplicate document seen by the user in the past. Since entire documents are rarely redundant with respect to each other, nuggets provide a finer granularity for relevance and novelty-based evaluation.

Next, we describe the Expected Global Utility (EGU) in detail, followed by our proposed approach for optimizing retrieval systems with respect to this performance measure.

2.1 Optimization Criterion: Expected Global Utility

Utility. EGU is based on the notion of "utility" of each document returned by the system, defined as the difference between the gain and cost that would be accrued by the user when he or she reads that document.

$$\mathbb{U}(d_i) = \mathbb{G}(d_i) - \mathbb{C}(d_i) \tag{1}$$

The cost of reading the document represents the time and effort expended by the user in going through the system's output. The simplest definition would be unit cost per document, which explicitly penalizes longer ranked lists[2]. More sophisticated definitions of cost can be based on length of the document, its language, and so on.

The gain from reading the document is defined in terms of its nuggets, whose contribution is discounted based on how many times they have been seen by the user previously:

$$\mathbb{G}(d_i) = \sum_{\delta \in d_i} \mathbf{w}_\delta \gamma^{\eta(\delta, i-1)} \tag{2}$$

where δ is a nugget, $\mathbf{w}_\delta$ is the weight of δ, $\eta(\delta, i-1)$ is the number of times δ appears in the ranked list up to rank $i-1$. γ is a pre-specified parameter that reflects the user's tolerance for redundancy. Thus, the gain received from each

[1] Note that nuggets are defined at a conceptual level, *e.g.*, a nugget like "Barack Obama" acts as a placeholder for all possible ways of referring to the same person, *e.g.*, "Obama", "President Obama", etc. Therefore, nuggets must be somehow matched to their various possible surface-level manifestations. To solve this problem, we use the dataset and nugget-matching rules that are made publicly available by [28]. (See Section 4.1 for description of the dataset).

[2] Such a penalization can be used to evaluate retrieval systems that are expected to limit the amount of information shown to the user, but is irrelevant for evaluating search engines, which produce very long ranked lists that are never fully browsed by the user.

successive presentation of the same nugget is discounted by a factor γ. If $\gamma = 1$, no discounting takes place: The user is assumed to be fully tolerant to redundancy, and the evaluation reduces to be relevance-based only. At the other extreme of $\gamma = 0$, reading a nugget after the first time is assumed to be completely useless to the user (interpreting $0^n = 1$ if $n = 0$, and 0 for $n > 0$).

The gain of an entire ranked list can thus be defined as:

$$\mathbb{G}(L|q) = \sum_{\delta \in \Delta_q} \mathbf{w}_\delta \frac{1 - \gamma^{\eta_\delta(L)}}{1 - \gamma} \tag{3}$$

where $\eta_\delta(L)$ is the number of times nugget δ appears in the ranked list L, and Δ_q is the set of all nuggets that are relevant to the query q.

Expected Utility. The current definition of gain of a ranked list does not favor early retrieval of useful information. We must take the typical browsing behavior of users into account: Users are more likely to browse ranked lists in a top-down manner and stop at some position due to various reasons like satisfaction or frustration. To capture this behavior, the utility of a ranked list is interpreted as a function of the user's stopping position s, which is assumed to be a random variable. By defining a probability distribution over s, say, $\Pr(s|p)$ with parameter p, we can obtain the *expected* utility of a ranked list:

$$\begin{aligned} \mathbb{E}_s[\mathbb{U}(s)] &= \sum_{s=1}^{\ell} \Pr(s|p)\mathbb{U}(s) && (4) \\ &= \sum_{s=1}^{\ell} \Pr(s|p)(\mathbb{G}(s) - \mathbb{C}(s)) && (5) \end{aligned}$$

We assume $\Pr(s)$ to be a geometric distribution with parameter p to model the common observation that users are more likely to stop at early positions in a ranked list.

Multiple Ranked Lists The above definition extends naturally to session-based retrieval with multiple ranked lists by extending the definition of stopping distribution as well as utility to multiple ranked lists[3]. Specifically, let $\Pr(s) = \Pr(s_1)\Pr(s_1)...\Pr(s_K)$, where $s_1, ...s_K$ represent the respecting stopping positions in the K ranked lists. Similarly, utility can be defined as a function of stopping positions in all ranked lists by considering only the nuggets that appear in the respective top $s_1..s_K$ documents in each ranked list. Since the expectation is calculated by summing over all stopping positions in each ranked list, we can obtain a more accurate estimate of the utility of multiple interrelated ranked lists without having to assume a hard reading cut-off in each ranked list[4].

Next, we focus on the main problem of system optimization, *i.e.*, designing a retrieval strategy to maximize EGU.

2.2 System Optimization

As mentioned earlier, the ground truth is defined in terms of nuggets. However, a retrieval system would obviously not have access to the true nuggets for an unseen query. Therefore, it must depend on observable features of documents (*e.g.*, words) to rank them with respect to a query. Several approaches have been used for estimating the relevance of documents in terms of the query and document words, *e.g.*, hand-crafted scoring functions like BM25 [23], or probabilistic approaches like language modeling [17]. Similarly, novelty of documents has been measured in terms of word-level similarities between documents, *e.g.*, cosine similarities [4, 27]. However, such approaches would only provide an indirect way of optimizing EGU without taking into account its various nuances like user's tolerance for redundancy (γ, see Eq. 3) and browsing persistence (p, see Eq. 4). Moreover, such approaches measure relevance and novelty independently and then combine them to score each document (*e.g.*, using the MMR strategy [4]). That is, relevance and novelty are treated as compensatory: High novelty can compensate for low relevance. Hence, it is possible for such methods to favor a document that is highly novel but irrelevant to the given query. However, users generally treat relevance as a pre-condition for usefulness of information [19, 12]. Therefore, it is more appropriate to directly target the retrieval of relevant *and* novel information.

Nuggets directly capture relevant and novel information. Therefore, we argue that the system's model of relevance and novelty should also be based on nuggets. Again, since the true nuggets are unknown to the system, it must use observable features (*e.g.*, words or named entities) as surrogates for the true nuggets. However, the main challenge is that not all features are equally important: *e.g.*, certain frequently occurring words known as "stopwords" carry negligible information. Moreover, the importance of a feature depends on the user's query: *e.g.*, for a query like "BP oil spill", the system should focus on the coverage of words and named entities that denote the occurrence, consequences, and containment efforts related to the oil spill. Furthermore, for broad or ambiguous queries, the importance of the features might depend on the intention or focus of the particular user, which can be determined based on explicit or implicit feedback.

Next, we will illustrate our retrieval approach and describe how it addresses the above-mentioned problems.

Proposed Retrieval Approach. Given a query, ranking documents is a multi-step process:

1. Obtain a candidate set of documents using a standard retrieval approach.

2. Identify and assign weights to all features that appear in the candidate set.

3. Re-rank the documents to maximize the coverage of the features as defined by the EGU objective function.

4. Update weights of features based on user feedback. Repeat.

Let us look at each of these steps in detail.

2.2.1 *Obtaining a Candidate Set*

This step is accomplished using an off-the-shelf retrieval system, and serves to limit the number of documents that need to considered for creating the final ranking. In our experiments, we use a state-of-the-art retrieval engine, Indri [26], to retrieve the initial set of documents.

[3] Since the multiple ranked lists are assumed to be part of a single coherent session, their utility cannot be calculated independently of each other due to the non-independent nature of novelty.

[4] Summing over all stopping positions in multiple ranked lists can be computationally prohibitive; See Section A.1 for solutions.

2.2.2 *Identifying and Assigning Weights to Features*

The candidate set of documents is used to extract features that will act as surrogates for the true nuggets that the user is interested in. We use the following features as surrogates for nuggets:

- **Words:** The simplest and most straightforward choice is to use words as surrogates for nuggets. Then, our goal is to re-rank the initial ranked list so as to cover as many different words as possible, subject to an appropriate weighting scheme.

- **Named Entities:** Named entities are phrases that contain names of persons, organizations, locations, times, and quantities. They can be treated as units of information and have been used to support various natural language applications tasks like *e.g.*, retrieval [3], novelty detection [15], and question answering, where a majority of *who*-, *where*-, and *when*- questions have answers in the form of person, location, and temporal entities, respectively [21, 25].

Each nugget (*i.e.*, word or named entity) is assigned a weight based on two factors:

1. Its TF–IDF (Term Frequency–Inverse Document Frequency) value, where term frequency (TF) is defined as the number of times the nugget appears in the candidate set of documents, and inverse document frequency (IDF) is negative logarithm of the fraction of documents in the entire corpus that contain the nugget. Favoring nuggets that occur frequently in the initial result-set but are not too common in the entire corpus serves the purpose of identifying nuggets that are potentially relevant and discriminative with respect to the user's query.

2. The scores of the documents (as assigned by the initial retrieval engine—Indri in our case) that the nugget appears in. This serves the purpose of favoring those terms that appear in documents deemed more relevant by the retrieval engine. Since a single nugget can appear in multiple documents, we use the average score of such documents.

That is, the weight w_δ assigned to the nugget δ is:

$$\mathbf{w}_\delta = \bar{s}(\delta) \cdot \mathrm{TF}(\delta) \cdot \mathrm{IDF}(\delta) \tag{6}$$

where $\bar{s}(\delta)$ is the average score of the documents in which δ appears.

Note that this is only an initial assignment of weights and is updated based on user feedback, as described in Section 2.2.4.

2.2.3 *Ranking the Documents*

Once weights are assigned to features, we must rank the documents so as to maximize the weighted coverage of the features at the top ranks, so that a typical user who browses the ranked list in a top-down manner is likely to come across the most number of (surrogate) nuggets. It is already known that finding the optimal set of documents that will maximize the coverage of any discrete elements (aspects, nuggets, categories, etc.) is an NP-hard problem [30, 1, 5]. The NP-hardness also extends to the ranking problem. In Section A.2, we show that maximizing EGU can be reduced to the Maximum Coverage Problem, and that a simple greedy algorithm guarantees good performance due to the submodularity of EGU. Here, we only focus on the greedy algorithm. Algorithm 1 shows the steps involved in the greedy algorithm for ranking. At each step, the algorithm picks the document with the highest *marginal utility*, which in turn depends on the expected number of times each nugget in the document has already appeared in documents already included in the ranked list L:

$$\textsc{Marginal-Utility}(d_i, L) = \sum_{\delta \in d_i} \mathbf{w}_\delta \gamma^{\eta_\delta(L)} \tag{7}$$

Algorithm 1 Greedy algorithm (Greedy)

```
/* Input: Documents to rank $D = \{d_1, d_2, ..., d_k\}$ */
/* Output: Ranked list $L$ */
$L \leftarrow \emptyset$
for $i = 1$ to $k$ do
    $j \leftarrow$ Choose-Next-Doc($L, D$)
    $L \leftarrow L \cup d_j$
    $D \leftarrow D \setminus d_j$
end for

sub Choose-Next-Doc($L, D$) do
    /* Input: Current ranked list $L$, remaining documents $D$ */
    /* Output: Index of next document from $D$ to include in the ranked list */
    for $i = 1$ to $|D|$ do
        $s_0(i) =$ Marginal-Utility($d_i, L$)
    end for
    return $\mathrm{argmax}_i\, s_0(i)$
end sub
```

The definition of marginal utility can be extended to multiple ranked lists by replacing $\eta_\delta(L)$ by the expected number of times each nugget appeared in all previously displayed ranked lists (Eq. 14).

2.2.4 *Updating Weights based on User Feedback*

The initial weights obtained in Section 2.2.2 are further tuned to adapt to the user's information needs by leveraging feedback obtained from the user. In a deployed system, such feedback could be available in an explicit (*e.g.*, "like"/"dislike" buttons in the interface) or implicit (*e.g.*, clicks on documents) form. For the purpose of this paper, we will simulate the presence of positive and negative feedback on documents returned by the system. We aim to use this feedback to automatically infer the user's interest in particular pieces of information, *i.e.*, nuggets.

Learning from user feedback poses two main challenges: First, user feedback is generally available at the document level. However, we need to learn the concept of usefulness at a much lower granularity of nuggets. Second, unlike traditional retrieval setup, where the relevance of each document is assumed to be independent of other documents in the ranked list, the usefulness of each document depends on other documents presented in the ranked list. Therefore, the user's feedback on a document can no longer be assumed to be independent of what he or she has seen before the current document. In other words, the user's feedback is an indicator of the *marginal* utility of a document, not its absolute utility.

To solve both these problems, we use a learning approach based on logistic regression, which models the user's feedback as a function of the marginal gain provided by each document (as opposed to its absolute gain). Specifically, the log-odds of receiving a positive feedback on a document is modeled as a linear combination of the marginal gain of each nugget in the document:

$$\Pr(f_i = 1|\mathbf{w}) = \frac{1}{1+e^{-(g_i(\mathbf{w})+b)}} \qquad (8)$$

where f_i is the feedback on the i^{th} document, and $g_i(\mathbf{w})$ is the corresponding marginal gain interpreted as a function of the weights $\mathbf{w}$ of the nuggets:

$$g_i(\mathbf{w}) = \sum_{\delta\in d_i} \mathbf{w}_\delta \gamma^{\mathbb{E}n_\delta(d_{1:i-1})} \qquad (9)$$

where $\mathbf{w}_\delta$ is the weight of nugget δ, and $\mathbb{E}n_\delta(d_{1:i-1})$ is the expected number of times nugget δ has been seen by the user before the current document.

Thus, each document in a ranked list is a training instance with label equal to the user's feedback (+1 or –1), and predictors equal to the marginal gain of each nugget. The goal of logistic regression is to find the weights for nuggets that best explain the observed user feedback. The optimal weights $\mathbf{w}^*$ are found through maximum a-posteriori (MAP) estimation, using a Normal prior whose mean is equal to the current estimate of the weights:

$$\Pr(\mathbf{w}) \sim \mathcal{N}(\mathbf{w}_0, \lambda\mathbf{I}) \qquad (10)$$

where λ controls the strength of the prior. The use of a prior allows the system to adapt to the user's interests in an incremental manner by using the previous iteration's weights as the prior for the current step.

The optimal weights maximize the log-likelihood over all documents on which feedback is received:

$$\ell(\mathbf{w}) = -\sum_i \log(1+exp(-f_i g_i(\mathbf{w})-b)) - \frac{\lambda}{2}||\mathbf{w}-\mathbf{w}_0||^2 \qquad (11)$$

which can be solved efficiently using conjugate gradient ascent [18].

3. TIME COMPLEXITY

Let us estimate the time complexity of the proposed approach and compare it against the MMR strategy for novelty-based ranking. Assume an initial candidate set of documents (cf. Section 2.2.1) of size C. Since retrieval of this candidate set is common to both re-ranking approaches (and also not the focus of this paper), we will ignore it from the time complexity calculation. The goal is to create a new ranked list of length k. Assume that each document contains W words on average.

The proposed approach first processes the candidate set of documents to assign initial weights to all surrogate nuggets (*e.g.*, words), which requires O(CW) operations. Then, the ranked list is built in k steps: At each step, score the remaining O(C) documents based on their marginal utility, which requires O(W) operations per document. This step leads to a time complexity of O(kCW). Hence, the total time complexity of the proposed approach is O($CW + kCW$).

The MMR-based approach is also based on iteratively building the ranked list: At each of the k steps, score the remaining O(C) documents by computing their cosine similarity with each of the O(k) documents already selected in the ranked list. A single cosine similarity computation takes O(W) operations. This leads to a time complexity of O(k^2CW).

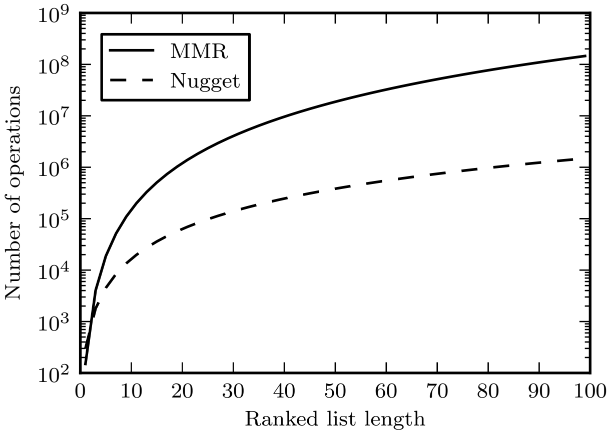

Figure 1: Time complexity of the proposed nugget-based re-ranking vs. that of MMR-based re-ranking approach.

Since MMR is based on computing similarities with all previously displayed documents, its computation is quadratic in the length of the ranked lists, which can be prohibitive for long ranked lists, or multiple ranked lists in a search session. On the other hand, the proposed approach is based on marginal utilities of documents, which can be computed using a running total of the number of times each nugget has been previously displayed to the user. This acts as a succinct representation of the user's browsing history and leads to the linear time complexity with respect to ranked list length. Figure 1 shows the difference in time complexities of the two approaches as a function of the ranked list length (k), assuming that the average document length (W) is 50 words and the initially retrieved set of candidate documents is three times the number of documents in the target ranked list, *i.e.*, $C = 3 \cdot k$.

4. EXPERIMENTS

We use the Topic Detection and Tracking dataset (described below) to compare our proposed approach against the following baselines: (i) Indri [26], which is a state-of-the-art retrieval engine, and represents a purely relevance based approach, and (ii) Indri+MMR: Indri is used to create an initial set of documents for each query, which are re-ranked according to the Maximal Marginal Relevance (MMR) [4] criterion. This represents a baseline for novelty-based ranking that uses cosine similarities between document vectors to measure redundancy, which is then combined with the relevance score in a linear fashion[5].

We include four variants of the proposed approach to un-

[5] MMR involves a parameter λ (see [4] for details) that controls the trade-off between relevance and novelty. We used a validation set to choose the best value of λ in all experiments.

derstand its behavior: (i) Indri+W, which re-ranks an initial set of documents returned by Indri by using words as surrogates for nuggets, but makes no use of feedback, (ii) Indri+W+F, which again uses words as surrogates but updates their weights based on user feedback, (iii) Indri+NE+F, which uses named entities as surrogates and leverages user feedback as above, and (iv) Indri+W+NE+F, which uses both words and named entities as surrogates and also leverages user feedback.

To assess the behavior and performance of the greedy algorithm for novelty-based ranking, we generate synthetic ranked lists and evaluate the ranked list returned by the greedy algorithm against the true ideal ranked list obtained using exhaustive search.

4.1 Data

Topic Detection and Tracking (TDT) Data. TDT4 was a benchmark corpus used in Topic Detection and Tracking (TDT2002 and TDT2003) evaluations. It consists of over 90,000 articles from various news sources published between October 2000 and January 2001. This corpus was extended for novelty-based evaluations by creating 120 queries with corresponding nuggets and nugget-matching rules as described in [13, 28]. To simulate session-based retrieval with multiple rounds of retrieval and user feedback, we divided the 4-month span of the corpus into 25 chunks, each comprising approximately 5 consecutive days. A retrieval system is expected to produce a ranked list of documents at the end of each chunk, receive feedback from the user, and then produce a new ranked list for the next chunk, and so on. This setup simulates a user who is following an evolving news event over an extended period of time—expecting the retrieval system to return a personalized ranked list of relevant and novel documents after every 5 days.

4.2 Evaluation Metric

We used EGU with three different values of γ: 0.0, 0.1, and 1.0, to simulate different tolerances towards redundancy. 0.0 corresponds to no tolerance, 0.1 corresponds to some tolerance, and 1.0 corresponds to full tolerance for redundancy, *i.e.*, the traditional relevance-only based retrieval setup. The user's stopping probability p was set to 0.1, which corresponds to an average reading length of 10 documents. Since we are mainly interested in the ability of the system to return relevant and novel documents, and do not care about the lengths of the ranked lists (as is common in ranked retrieval evaluation), we use zero cost in EGU (see Section 2.1).

4.3 Results

Main Results. Table 1 shows the performance obtained by the baselines and different settings of the proposed approaches[6]. The use of words and named entities as surrogates for re-ranking, with weights updated through user feedback (Indri+W+NE+F) performs the best. The performance of words-only with feedback (Indri+W+F) and named entities-only with feedback (Indri+NE+F) is close in all cases, but their combination leads to the best performance.

When novelty is a factor of consideration (*i.e.*, $\gamma = 0.0$ or 0.1), the proposed approach using words as surrogates but without feedback (Indri+W) performs better than the baselines, which demonstrates its ability to model novelty effectively. However, it is disappointing that the feedback-based variants do not exhibit a substantial improvement over the no-feedback variant. To some extent, this can be explained by the fact that the benefits of feedback are nullified by the demand for novelty: Through feedback, the user indicates interest in specific items, but at the same time, expects the system to not retrieve the same (or very similar) items in the future, but instead, other items that are relevant *and* novel. Our evaluation merely shows that the benefits of user feedback may be overestimated if novelty (or redundancy) is not taken into account.

Table 1: EGU scores of different systems for three values of γ.

System	EGU		
	($\gamma = 0.0$)	($\gamma = 0.1$)	($\gamma = 1.0$)
(Baselines)			
Indri	0.3360	0.4016	0.4890
Indri+MMR	0.3424	0.4202	0.4890
(Proposed)			
Indri+W	0.3501	0.4278*	0.4666
Indri+W+F	0.3656*†	0.4401*†	0.5017
Indri+NE+F	0.3683*†	0.4445*†	0.5121*†
Indri+W+NE+F	**0.3688***†	**0.4473***†	**0.5192***†

On the other hand, when the user has full tolerance for redundancy (*i.e.*, $\gamma = 1.0$), the proposed approach without any feedback (Indri+W) performs worse than the baseline, which shows that the proposed re-ranking approach is less effective for relevance-based ranking unless feedback is provided. Hence, there is an opportunity to improve the re-ranking approach itself (possibly through the use of more sophisticated surrogates for nuggets), which would lead to further improvements across all settings. When feedback is provided, the proposed method indeed performs substantially better.

For $\gamma = 0.0$ and 0.1, MMR-based retrieval (Indri+MMR) performs better than the baseline of relevance-only ranking (Indri), as expected. Although the improvement in performance of the proposed approaches over MMR is not substantial (especially if feedback is not considered), the proposed approaches provide a computationally efficient alternative without requiring parameter tuning. In the MMR approach, the parameter λ must be re-tuned for different tolerances for redundancy (γ), whereas the proposed approaches directly take this into account through the definition of marginal utility (Eq. 7), which can enable a user to dynamically change his or her redundancy tolerance in a deployed system. Moreover, as mentioned in Section 3, the proposed approach can be much more computationally efficient compared to the MMR approach for longer (or multiple) ranked lists. For $\gamma = 1.0$, the MMR approach obtains the same performance as the Indri baseline because the optimal value of λ (see Section 4) was found to be zero as expected, *i.e.*, no novelty component in the document scoring.

Performance of the greedy algorithm. An important step in our approach is the use of the greedy algorithm

[6] The symbols * and † indicate statistically significant differences ($p < 0.01$ for sign test with paired queries) with respect to the two baselines, respectively.

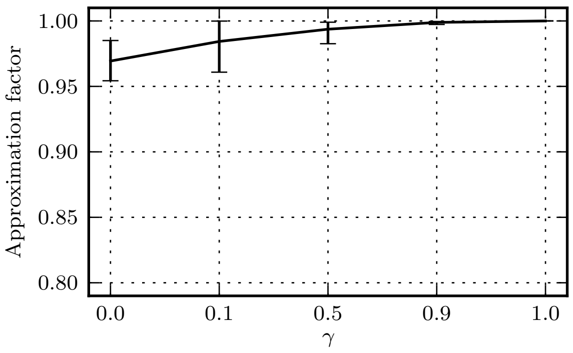

Figure 2: Approximation factors achieved by the greedy algorithm on synthetic ranked lists for different redundancy tolerances γ.

to rank the documents so as to maximize the coverage of nuggets at the top ranks. Therefore, we wish to understand the performance characteristics of the greedy algorithm, and assess whether its use is justified for optimizing EGU. While the greedy algorithm has been shown to perform well for solving MAX-COVER and SET-COVER problems [5], EGU is a more general form of MAX-COVER due to the expectation taken over all possible rank positions (thus, leading to a probabilistic version of MAX-COVER) as well as the notion of diminishing returns adjustable using the γ parameter (thus, leading to a notion of "soft coverage").

We use 1000 randomly generated ranked lists to assess the effect of γ (user's redundancy tolerance) on the behavior of the greedy algorithm. Figure 2 shows the performance obtained by the greedy algorithm for different values of γ. We have plotted the mean (flanked by minimum and maximum) approximation factors[7], *i.e.*, the score of the greedy algorithm divided by the best possible score obtained using exhaustive search. Notice that the approximation factor tends to improve (*i.e.*, gets close to 1.0) as the value of γ increases. This is intuitive, since $\gamma = 1$ corresponds to relevance-based ranking, which does not penalize redundancy, and hence, a simple greedy approach of ranking by decreasing number of nuggets is provably optimal. At the other extreme, $\gamma = 0$ corresponds to the "hard" notion of coverage where every nugget is only counted once, which corresponds to the standard MAX-COVER problem.

5. RELATED WORK

One of the earliest works on novelty and diversity-based ranking is the Maximal Marginal Relevance (MMR) method [4], which proposed a greedy algorithm that incrementally builds the ranked list by choosing the next document with the highest "marginal" relevance, *i.e.*, high relevance to the query, and low similarity to already selected documents in the ranked list. However, as mentioned in Section 2.2, MMR can lead to sub-optimal performance due to the independent treatment of relevance and novelty. Our proposed approach is based on a unified model of relevance and novelty in terms of nuggets.

Zhai et al. [30] proposed an approach for diversity-based retrieval, where diversity is defined in terms of the number of sub-topics covered by the retrieved documents for a given search topic. To measure the quality of diversity-based rankings, they extended the traditional measures of recall and precision for sub-topic retrieval and defined two new measures: S-recall and S-precision. However, the authors point out the challenges in defining a single summary measure that combines relevance and novelty. As a compromise, they measure aspect coverage at a few arbitrarily defined recall levels. For system optimization, the authors use language modeling based scores for relevance and novelty in an MMR-like formulation. However, it is evident from the definition of S-recall and S-precision that their framework does not allow different tolerances towards redundancy: A sub-topic is either covered or uncovered, and subsequent presentations or the same sub-topic receive no credit. In other words, $\mathcal{S}$*-recall* and $\mathcal{S}$*-precision* are inflexible in that they assume no tolerance towards redundancy for all users.

Agrawal et al. [1] proposed an approach for maximizing the coverage of different categories of documents in the ranked list to deal with the inherent ambiguity associated with certain user queries. Their goal was to maximize the probability that the user will find at least one document relevant to his or her true intent. However, the authors admit that such a criterion might be too conservative when serving users who desire a certain level of redundancy. Moreover, their objective function is set-based, *i.e.*, it does not differentiate between different permutations of the selected documents. In reality, the ranking of documents plays an important role in the perceived utility of the system. The top-down browsing behavior of users is not explicitly modeled by the objective function, and only manifests as a side-effect of the greedy ranking algorithm.

In the above-mentioned approaches, there is a disconnect between the evaluation metric (e.g., S-recall, S-precision in [30], and IA-NDCG and IA-MAP in [1]), and the objective function used by the system. In contrast, this paper represents the first framework where the evaluation metric and objective function coincide for the optimization of relevance and novelty-based retrieval.

Clarke at el. [8] proposed α-NDCG as a variation of NDCG to model relevance and novelty in terms of nuggets. However, α-NDCG is not based on an explicit model of user browsing behavior, *i.e.*, the likelihood of user stopping at various ranks. Therefore, it does not naturally extend to multiple ranked lists since it is not clear which nuggets in documents from previous ranked lists would be deemed as read by the user for the purpose of evaluating the current ranked list. On the other hand, EGU is based on a probabilistic model of user behavior, and hence, is a more general optimization criterion for retrieval performance over one or more ranked lists.

El-Arini et al. [11] proposed an approach for learning from user's feedback to provide a personalized set of diverse blogs to the user. They address the problem of non-independent feedback. However, their objective function is set-based, and hence, does not take ranking performance into account. Also, similar to the other approaches for diversity-based retrieval, different tolerances towards redundancy are not taken into account.

Radlinski et al. [22] proposed the use of click-through data to improve the document rankings produced by the retrieval

[7] Only those ranked lists where the greedy algorithm led to sub-optimal performance were included.

system. Since real users implicitly take all pertinent factors (relevance and novelty with respect to previously seen documents) into account when clicking on documents, such an approach can optimize for novelty without explicitly modeling it. However, such approaches are expensive since they require multiple interactions with real users to collect click-through patterns on different variations of ranked lists for each query. Therefore, such approaches do not provide an efficient means for offline evaluation and tuning of new and potentially risky algorithms.

6. CONCLUSIONS

Expected Global Utility (EGU) is a recently proposed metric that has several desirable characteristics for measuring the performance of novelty-based ranking systems. We present the first approach for directly optimizing retrieval systems with respect to this performance measure. EGU models relevance and novelty in terms of "nuggets"; Since a retrieval system does not have access to the true nuggets for a given query, our approach is based on the use of observable features like words and named entities as surrogates for the true nuggets, whose weights are updated based on user feedback in an iterative search session. We show that the ranking problem is NP-hard, and use mathematical as well as empirical analysis to demonstrate that a simple greedy algorithm achieves good performance with respect to EGU. Our experiments on a nugget-based data collection indicate that the proposed approach can successfully optimize the performance in terms of EGU, compared to a purely relevance based approach (Indri) as well as an MMR-based approach, which is computationally expensive for longer (or multiple) ranked lists.

7. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14. ACM, 2009.

[2] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA broadcast news transcription and understanding workshop*, volume 1998. Citeseer, 1998.

[3] A. Caputo, P. Basile, and G. Semeraro. Boosting a Semantic Search Engine by Named Entities. In *Proceedings of the 18th International Symposium on Foundations of Intelligent Systems*, page 250. Springer, 2009.

[4] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM New York, NY, USA, 1998.

[5] B. Carterette. An Analysis of NP-Completeness in Novelty and Diversity Ranking. In *Proceedings of the 2nd International Conference on Theory of Information Retrieval: Advances in Information Retrieval Theory*, page 211. Springer, 2009.

[6] H. Chen and D. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, page 436. ACM, 2006.

[7] R. Church and C. ReVelle. The maximal covering location problem. *Papers in regional science*, 32(1):101–118, 1974.

[8] C. Clarke, M. Kolla, G. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666. ACM New York, NY, USA, 2008.

[9] G. Cornuejols, M. Fisher, and G. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810, 1977.

[10] H. Dang, J. Lin, and D. Kelly. Overview of the TREC 2006 question answering track. *TREC 2006*, 2006.

[11] K. El-Arini, G. Veda, D. Shahaf, and C. Guestrin. Turning down the noise in the blogosphere. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–298. ACM New York, NY, USA, 2009.

[12] H. Greisdorf. Relevance thresholds: a multi-stage predictive model of how users evaluate information. *Information Processing & Management*, 39(3):403–423, 2003.

[13] D. He, P. Brusilovsky, J. Ahn, J. Grady, R. Farzan, Y. Peng, Y. Yang, and M. Rogati. An evaluation of adaptive filtering in the context of realistic task-based information exploration. *Information Processing and Management*, 44(2):511–533, 2008.

[14] K. Järvelin, S. Price, L. Delcambre, and M. Nielsen. Discounted cumulated gain based evaluation of multiple-query IR sessions. In *Proceedings of the 30th European Conference on Information Retrieval*, 2008.

[15] G. Kumaran and J. Allan. Text classification and named entities for new event detection. *Proceedings of the 27th annual international conference on Research and developement in information retrieval*, pages 297–304, 2004.

[16] A. Lad and Y. Yang. Generalizing from relevance feedback using named entity wildcards. *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 721–730, 2007.

[17] J. Lafferty and C. Zhai. Probabilistic relevance models based on document and query generation. *Language modeling for information retrieval*, 13:1–10, 2003.

[18] T. Minka. A comparison of numerical optimizers for logistic regression. *Unpublished draft*, 2003.

[19] S. Mizzaro. Relevance: The whole history. *Historical studies in information science*, pages 221–244, 1998.

[20] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

[21] J. Prager, E. Brown, A. Coden, and D. Radev. Question answering using predictive annotation. *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in*

Information Retrieval (SIGIR2000), pages 184–191, 2000.

[22] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, pages 784–791. ACM New York, NY, USA, 2008.

[23] S. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241. Springer-Verlag New York, Inc. New York, NY, USA, 1994.

[24] I. Soboroff. Overview of the TREC 2004 novelty track. In *Proceedings of the 13th Text REtrieval Conference (TREC 2004)*. Citeseer, 2004.

[25] R. Srihari and W. Li. A question answering system supported by information extraction. *Proceedings of the sixth conference on Applied natural language processing*, pages 166–172, 2000.

[26] T. Strohman, D. Metzler, H. Turtle, and W. Croft. Indri: A language model-based serach engine for complex queries. *Proceedings of the International Conference on Intelligence Analysis*, 2004.

[27] Y. Yang and A. Lad. Modeling Expected Utility of Multi-session Information Distillation. In *Proceedings of the 2nd International Conference on Theory of Information Retrieval: Advances in Information Retrieval Theory*, page 175. Springer, 2009.

[28] Y. Yang, A. Lad, N. Lao, A. Harpale, B. Kisiel, and M. Rogati. Utility-based information distillation over temporally sequenced documents. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 31–38, 2007.

[29] Y. Yue and T. Joachims. Predicting diverse subsets using structural SVMs. In *Proceedings of the 25th international conference on Machine learning*, pages 1224–1231. ACM New York, NY, USA, 2008.

[30] C. Zhai, W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 10–17, 2003.

[31] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410. ACM New York, NY, USA, 2001.

[32] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 81–88. ACM New York, NY, USA, 2002.

APPENDIX

A. MISCELLANEOUS DETAILS

A.1 Efficient Calculation of EGU for Multiple Ranked Lists

The computation of `EGU` becomes intractable with increasing number and lengths of ranked lists. Specifically, the computation of expected gain requires summing over all combinations of stopping positions in all ranked lists. Expected cost is easy to calculate for the linear definition of cost used in this paper. Therefore, we will only focus on the computation of expected gain over multiple ranked lists. First, let us re-write expected utility as the sum of expected gain obtained for each nugget:

$$\texttt{EGU} = \sum_{\delta} \mathbf{w}_{\delta} \mathbb{E}\left[\frac{1-\gamma^{\eta_\delta(s)}}{1-\gamma}\right] \tag{12}$$

where nugget counts $\eta_\delta(s)$ depend on the stopping positions $s = s_1 ... s_K$ in the K ranked lists, respectively. We can approximate `EGU` by moving the expectation operator inside:

$$\texttt{EGU} \approx \sum_{\delta} \mathbf{w}_{\delta} \frac{1-\gamma^{\mathbb{E}[\eta_\delta(s)]}}{1-\gamma} \tag{13}$$

That is, instead of calculating the expected gain with respect to different browsing patterns, we compute the gain obtained by the expected number of times each nugget will be read from all ranked lists, *i.e.*, $\mathbb{E}\left[\eta(s)\right]$. This quantity can be efficiently calculated by summing over the expected nugget counts in each ranked list:

$$\mathbb{E}\left[\eta_\delta(s)\right] = \sum_{k=1}^{K} \sum_{s_k=1}^{|L_k|} \Pr(s_k) \eta_\delta(s_k) \tag{14}$$

Thus, the approximate computation requires a sum over $O(|L_1| + |L_2| + ... + |L_K|)$ terms, instead of the $O(|L_1| \times |L_2| \times ... \times |L_K|)$ terms in the original calculation, which must consider all combinations of stopping positions in the K ranked lists.

A.2 NP-Hardness of EGU

Let us focus on a particular parameterization of $\gamma = 0$ *i.e.*, no tolerance towards redundancy, and $P(s = k) = 1$, *i.e.*, the user reads all documents from the top down and stops at a given rank, say k. Given a set of documents, all nuggets that appear in at least one of these documents are said to be *covered* by the set of documents. Then, finding a set of k documents that cover the most number of nuggets is exactly equivalent to the *Maximum Coverage Problem*, which is known to be NP-hard [7].

Maximum Coverage Problem (Max-Cover)**:** Given a collection of sets $S = S_1, S_2, ..., S_m$, each containing a subset of elements, *i.e.*, $S_i \subseteq \{e_1, e_2, ..., e_n\}$, find the subset $S^* \subseteq S$ of size K such that the number of covered elements is maximized:

$$\begin{aligned} &\underset{S^* \subseteq S}{\text{argmax}} \quad \left| \bigcup_{S_i \in S*} S_i \right| \\ &\text{s.t.} \quad |S^*| = k \end{aligned} \tag{15}$$

Our ranking problem can be reduced to Max-Cover by mapping documents to sets and nuggets to elements.

Submodularity. Our objective function admits additional structure that allows approximation algorithms to guarantee good performance. Specifically, the gain function of EGU is *submodular*. Submodularity formalizes the intuitive property of diminishing returns, and is defined as follows [20]: A set function F is called submodular if and only if for all $A \subseteq B \subseteq V$ and $s \in V \setminus B$ it holds that $F(A \cup \{s\}) - F(A) \geq F(B \cup \{s\}) - F(B)$.

The submodularity of the gain function follows directly from its concavity with respect to nugget counts. Intuitively, the increase in gain obtained by adding a document d_i to a ranked list L can never be larger than the increase obtained by adding d_i to a subset of L.

A classic result shows that the simple greedy algorithm of incrementally building the list of documents based on decreasing marginal utilities guarantees a constant approximation ratio: For any monotonic submodular function, the greedy algorithm achieves an approximation ratio of $(1-1/e)$ [20].

However, this lower-bound can be further improved by taking the special structure of EGU into account, as we show in Section A.3.

A.3 Improved Bound for Greedy Algorithm

Here, we develop a tighter bound on the performance of the greedy algorithm for the optimization of EGU. The lower-bound for MAX-COVER is $1 - 1/e$, which is approximately 0.63. The main idea for deriving the new bound is that the lower bound of $1 - 1/e$ is too conservative: It is guaranteed irrespective of the size k of the covering problem, whereas EGU involves an expectation of MAX-COVER problems of size $k = 1, 2, ...$, *i.e.*, all stopping positions.

The lower bound for MAX-COVER, as a function of k is [9]:

$$\frac{g_k}{I_k} \geq 1 - (1 - 1/k)^k \tag{16}$$

where g_k is the greedy solution and I_k is the ideal solution. The bound of $1 - 1/e$ arises because:

$$(1 - 1/k)^k < 1/e \tag{17}$$

which approaches equality for large k. However, for smaller values of k, the gap is large. For instance, for $k = 1$, the expression $(1 - 1/k)^k$ is equal to zero, which corresponds to the fact that the solution of size 1 is always optimal. In other words, approximation factors better than 0.63 can be guaranteed for covering problems of smaller size. Since the total gain, say G, is calculated by taking an expectation over all stopping positions, *i.e.*, $k = 1, 2, ...$, we can therefore derive a tighter bound by taking the size-dependent bound into account. Specifically, the total gain of the ideal solution, say I, is equal to:

$$I = \sum \Pr(k|p) I_k \tag{18}$$

But due to Eq. (16), we have:

$$I_k \leq \frac{g_k}{1 - (1 - 1/k)^k} \tag{19}$$

Therefore, the desired bound is:

$$\frac{G}{I} \geq \frac{\sum \Pr(k|p) g_k}{\sum \left(\frac{\Pr(k|p) g_k}{(1-(1-1/k)^k)} \right)} \geq 1 - 1/e \tag{20}$$

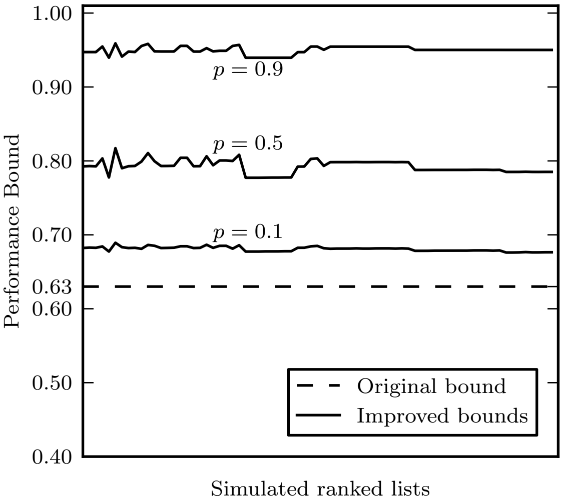

Figure 3: Comparison of the original $(1 - 1/e)$ and the improved bounds for various values of stopping probability p.

We compared this bound against the original bound of $(1-1/e)$ by running the greedy algorithm on 1000 synthetic ranked lists. Figure 3 shows the bounds obtained for various values of the stopping probability p (note that $\Pr(\cdot|p)$ is a geometric distribution with parameter p). Note that the improved bound is dependent on the greedy scores as well as the stopping probability, which is evident in Eq. (20). Also, the bound improves (get closer to 1) with increasing values of p, which is expected behavior because higher values of p correspond to higher likelihood of the user to stop at one of the top ranks, where the greedy algorithm guarantees better worst-case performance: In the extreme case of $p = 1$, *i.e.*, the user only reads the first document, the greedy strategy of choosing the document with most number of nuggets (breaking ties arbitrarily) is provably optimal.